

Ayoaron

0.0.4

Généré par Doxygen 1.7.4

Wed Nov 30 2011 14 :30 :11

Table des matières

1	Liste des choses à faire	1
2	Index des espaces de nommage	2
2.1	Liste des espaces de nommage	2
3	Index des classes	3
3.1	Hiérarchie des classes	3
4	Index des classes	5
4.1	Liste des classes	5
5	Index des fichiers	8
5.1	Liste des fichiers	8
6	Documentation des espaces de nommage	12
6.1	Référence de l'espace de nommage ayoaron	12
6.1.1	Description détaillée	12
6.2	Référence de l'espace de nommage ayoaron : :audio	12
6.2.1	Description détaillée	13
6.2.2	Documentation du type de l'énumération	13
6.3	Référence de l'espace de nommage ayoaron : :core	13
6.3.1	Description détaillée	14
6.4	Référence de l'espace de nommage ayoaron : :data	14
6.4.1	Documentation du type de l'énumération	14
6.4.2	Documentation des fonctions	15
6.5	Référence de l'espace de nommage ayoaron : :event	16
6.5.1	Description détaillée	17
6.6	Référence de l'espace de nommage ayoaron : :graphic	17
6.6.1	Description détaillée	17
6.7	Référence de l'espace de nommage ayoaron : :objets	17
6.7.1	Documentation des fonctions	18
6.8	Référence de l'espace de nommage ayoaron : :physic	19
6.8.1	Documentation du type de l'énumération	19
6.9	Référence de l'espace de nommage ayoaron : :reseau	19

6.9.1	Description détaillée	20
6.10	Référence de l'espace de nommage physique	20
6.10.1	Description détaillée	20
7	Documentation des classes	20
7.1	Référence de la classe ayoaron ::audio ::CSound	20
7.1.1	Documentation des constructeurs et destructeur	22
7.1.2	Documentation des énumérations membres	22
7.1.3	Documentation des données membres	27
7.2	Référence de la classe ayoaron ::audio ::CSoundEngine	27
7.2.1	Description détaillée	30
7.2.2	Documentation des énumérations membres	30
7.2.3	Documentation des constructeurs et destructeur	31
7.2.4	Documentation des fonctions membres	31
7.2.5	Documentation des données membres	45
7.3	Référence de la classe ayoaron ::audio ::CSoundOptions	48
7.3.1	Description détaillée	48
7.3.2	Documentation des constructeurs et destructeur	48
7.3.3	Documentation des données membres	48
7.4	Référence de la classe ayoaron ::core ::CAiPlayer	49
7.4.1	Description détaillée	51
7.4.2	Documentation des constructeurs et destructeur	51
7.4.3	Documentation des fonctions membres	51
7.5	Référence de la classe ayoaron ::core ::CAvailableAi	51
7.5.1	Description détaillée	52
7.5.2	Documentation des données membres	52
7.6	Référence de la classe ayoaron ::core ::CAvailableMap	52
7.6.1	Description détaillée	53
7.6.2	Documentation des données membres	53
7.7	Référence de la classe ayoaron ::core ::CConsole	54
7.7.1	Description détaillée	55
7.7.2	Documentation des constructeurs et destructeur	55
7.7.3	Documentation des fonctions membres	55
7.7.4	Documentation des données membres	57

7.8	Référence de la classe ayoaron ::core ::CEngine	58
7.8.1	Description détaillée	61
7.8.2	Documentation des énumérations membres	61
7.8.3	Documentation des constructeurs et destructeur	62
7.8.4	Documentation des fonctions membres	62
7.8.5	Documentation des données membres	72
7.9	Référence de la classe ayoaron ::core ::CGame	74
7.9.1	Description détaillée	75
7.9.2	Documentation des constructeurs et destructeur	75
7.9.3	Documentation des fonctions membres	76
7.9.4	Documentation des données membres	78
7.10	Référence de la classe ayoaron ::core ::CGameEngine	79
7.10.1	Description détaillée	82
7.10.2	Documentation des énumérations membres	82
7.10.3	Documentation des constructeurs et destructeur	83
7.10.4	Documentation des fonctions membres	83
7.10.5	Documentation des données membres	96
7.11	Référence de la classe ayoaron ::core ::CGameRound	99
7.11.1	Documentation des énumérations membres	101
7.11.2	Documentation des constructeurs et destructeur	101
7.11.3	Documentation des fonctions membres	102
7.11.4	Documentation des données membres	106
7.12	Référence de la classe ayoaron ::core ::CLogger	107
7.12.1	Description détaillée	109
7.12.2	Documentation des énumérations membres	109
7.12.3	Documentation des constructeurs et destructeur	109
7.12.4	Documentation des fonctions membres	109
7.12.5	Documentation des données membres	110
7.13	Référence de la classe ayoaron ::core ::CParameterAnalyser	110
7.13.1	Description détaillée	111
7.13.2	Documentation des constructeurs et destructeur	112
7.13.3	Documentation des données membres	112
7.14	Référence de la classe ayoaron ::core ::CPlayer	112
7.14.1	Description détaillée	114

7.14.2 Documentation des constructeurs et destructeur	114
7.14.3 Documentation des fonctions membres	114
7.14.4 Documentation des données membres	114
7.15 Référence de la classe ayoaron ::core ::CPlayerOptions	115
7.15.1 Description détaillée	115
7.15.2 Documentation des énumérations membres	115
7.15.3 Documentation des données membres	116
7.16 Référence de la classe ayoaron ::core ::CRoundOptions	117
7.16.1 Documentation des énumérations membres	117
7.16.2 Documentation des constructeurs et destructeur	118
7.16.3 Documentation des fonctions membres	118
7.16.4 Documentation des données membres	118
7.17 Référence de la classe ayoaron ::core ::CShared	119
7.17.1 Description détaillée	121
7.17.2 Documentation des constructeurs et destructeur	121
7.17.3 Documentation des fonctions membres	121
7.17.4 Documentation des données membres	123
7.18 Référence de la classe ayoaron ::core ::CSharedInstance	123
7.18.1 Description détaillée	125
7.18.2 Documentation des constructeurs et destructeur	125
7.18.3 Documentation des données membres	125
7.19 Référence de la classe ayoaron ::core ::CXMLExtractor	125
7.19.1 Description détaillée	126
7.19.2 Documentation des constructeurs et destructeur	126
7.19.3 Documentation des fonctions membres	126
7.19.4 Documentation des données membres	127
7.20 Référence de la classe ayoaron ::data ::CConfigFile	127
7.20.1 Description détaillée	128
7.20.2 Documentation des constructeurs et destructeur	128
7.20.3 Documentation des fonctions membres	128
7.20.4 Documentation des données membres	130
7.21 Référence de la classe ayoaron ::data ::CData	130
7.21.1 Description détaillée	132
7.21.2 Documentation des constructeurs et destructeur	132

7.21.3 Documentation des fonctions membres	133
7.21.4 Documentation des données membres	146
7.22 Référence de la classe ayoaron ::data ::CDataNode	146
7.22.1 Description détaillée	148
7.22.2 Documentation des constructeurs et destructeur	149
7.22.3 Documentation des fonctions membres	150
7.22.4 Documentation des données membres	172
7.23 Référence de la structure ayoaron ::data ::CDataStructure	173
7.23.1 Description détaillée	174
7.23.2 Documentation des données membres	174
7.24 Référence de la classe ayoaron ::data ::CFileManager	175
7.24.1 Description détaillée	177
7.24.2 Documentation des constructeurs et destructeur	177
7.24.3 Documentation des fonctions membres	177
7.24.4 Documentation des données membres	183
7.25 Référence de la structure ayoaron ::data ::SConfigFileData	184
7.25.1 Description détaillée	184
7.25.2 Documentation des données membres	184
7.26 Référence de la classe ayoaron ::event ::CEngineEvent	185
7.26.1 Description détaillée	186
7.26.2 Documentation des énumérations membres	186
7.26.3 Documentation des fonctions membres	187
7.26.4 Documentation des données membres	187
7.27 Référence de la classe ayoaron ::event ::CEventHandler	188
7.27.1 Description détaillée	189
7.27.2 Documentation des constructeurs et destructeur	189
7.27.3 Documentation des fonctions membres	190
7.27.4 Documentation des données membres	191
7.28 Référence de la classe ayoaron ::event ::CGameEventHandler	192
7.28.1 Description détaillée	193
7.28.2 Documentation des constructeurs et destructeur	193
7.28.3 Documentation des fonctions membres	194
7.28.4 Documentation des données membres	195
7.29 Référence de la classe ayoaron ::event ::CMenuEventHandler	196

7.29.1	Description détaillée	198
7.29.2	Documentation des constructeurs et destructeur	198
7.29.3	Documentation des fonctions membres	198
7.29.4	Documentation des données membres	199
7.30	Référence de la classe ayoaron ::event ::CSerializableVector3df	200
7.30.1	Description détaillée	201
7.30.2	Documentation des constructeurs et destructeur	201
7.30.3	Documentation des fonctions membres	201
7.31	Référence de la classe ayoaron ::graphic ::CGauge	201
7.31.1	Description détaillée	203
7.31.2	Documentation des constructeurs et destructeur	203
7.31.3	Documentation des fonctions membres	204
7.31.4	Documentation des données membres	206
7.32	Référence de la classe ayoaron ::graphic ::CGaugeMoteur	207
7.32.1	Description détaillée	209
7.32.2	Documentation des constructeurs et destructeur	209
7.32.3	Documentation des fonctions membres	209
7.32.4	Documentation des données membres	213
7.33	Référence de la classe ayoaron ::graphic ::CGaugeVitesse	214
7.33.1	Description détaillée	216
7.33.2	Documentation des constructeurs et destructeur	216
7.33.3	Documentation des fonctions membres	217
7.33.4	Documentation des données membres	223
7.34	Référence de la classe ayoaron ::graphic ::CGraphicsEngine	225
7.34.1	Description détaillée	228
7.34.2	Documentation des énumérations membres	228
7.34.3	Documentation des constructeurs et destructeur	229
7.34.4	Documentation des fonctions membres	230
7.34.5	Documentation des données membres	244
7.35	Référence de la classe ayoaron ::graphic ::CGraphicsOptions	248
7.35.1	Description détaillée	248
7.35.2	Documentation des constructeurs et destructeur	249
7.35.3	Documentation des données membres	249
7.36	Référence de la classe ayoaron ::graphic ::CGuiSystem	250

7.36.1	Description détaillée	252
7.36.2	Documentation des énumérations membres	252
7.36.3	Documentation des constructeurs et destructeur	252
7.36.4	Documentation des fonctions membres	253
7.36.5	Documentation des données membres	267
7.37	Référence de la classe ayoaron ::graphic ::CSceneGraph	267
7.37.1	Description détaillée	269
7.37.2	Documentation des constructeurs et destructeur	269
7.37.3	Documentation des fonctions membres	269
7.37.4	Documentation des données membres	273
7.38	Référence de la classe ayoaron ::graphic ::CVaisseauPanel	274
7.38.1	Description détaillée	275
7.38.2	Documentation des constructeurs et destructeur	275
7.38.3	Documentation des fonctions membres	275
7.38.4	Documentation des données membres	281
7.39	Référence de la structure ayoaron ::graphic ::datamot	282
7.39.1	Documentation des données membres	282
7.40	Référence de la classe ayoaron ::objets ::CAsteroide	282
7.40.1	Description détaillée	286
7.40.2	Documentation des constructeurs et destructeur	286
7.40.3	Documentation des fonctions membres	286
7.40.4	Documentation des données membres	314
7.41	Référence de la classe ayoaron ::objets ::CBoiteNoire	317
7.41.1	Description détaillée	319
7.41.2	Documentation des constructeurs et destructeur	319
7.41.3	Documentation des fonctions membres	319
7.41.4	Documentation des données membres	320
7.42	Référence de la classe ayoaron ::objets ::CModulePilotage	321
7.42.1	Description détaillée	322
7.42.2	Documentation des constructeurs et destructeur	322
7.42.3	Documentation des fonctions membres	322
7.42.4	Documentation des données membres	323
7.43	Référence de la classe ayoaron ::objets ::CMoteur	325
7.43.1	Description détaillée	329

7.43.2 Documentation des constructeurs et destructeur	329
7.43.3 Documentation des fonctions membres	329
7.43.4 Documentation des données membres	362
7.44 Référence de la classe ayoaron ::objets ::CPousMouv	365
7.44.1 Description détaillée	366
7.44.2 Documentation des constructeurs et destructeur	366
7.44.3 Documentation des fonctions membres	366
7.44.4 Documentation des données membres	368
7.45 Référence de la classe ayoaron ::objets ::CVaisseau	368
7.45.1 Description détaillée	372
7.45.2 Documentation des constructeurs et destructeur	372
7.45.3 Documentation des fonctions membres	373
7.45.4 Documentation des données membres	406
7.46 Référence de la classe ayoaron ::objets ::CVaisseauX1	409
7.46.1 Documentation des constructeurs et destructeur	414
7.46.2 Documentation des fonctions membres	414
7.46.3 Documentation des données membres	449
7.47 Référence de la classe ayoaron ::physic ::CEffort	453
7.47.1 Description détaillée	454
7.47.2 Documentation des constructeurs et destructeur	454
7.47.3 Documentation des fonctions membres	454
7.47.4 Documentation des données membres	456
7.48 Référence de la classe ayoaron ::physic ::CGalaxie	457
7.48.1 Description détaillée	458
7.48.2 Documentation des constructeurs et destructeur	458
7.48.3 Documentation des fonctions membres	458
7.48.4 Documentation des données membres	462
7.49 Référence de la classe ayoaron ::physic ::COBJet	463
7.49.1 Description détaillée	465
7.49.2 Documentation des constructeurs et destructeur	465
7.49.3 Documentation des fonctions membres	465
7.49.4 Documentation des données membres	475
7.50 Référence de la classe ayoaron ::physic ::CPhysicEngine	477
7.50.1 Description détaillée	480

7.50.2 Documentation des énumérations membres	480
7.50.3 Documentation des constructeurs et destructeur	480
7.50.4 Documentation des fonctions membres	480
7.50.5 Documentation des données membres	490
7.51 Référence de la classe ayoaron ::physic ::CSecteur	493
7.51.1 Description détaillée	494
7.51.2 Documentation des constructeurs et destructeur	494
7.51.3 Documentation des fonctions membres	495
7.51.4 Documentation des données membres	501
7.52 Référence de la classe ayoaron ::physic ::CSolideRigide	502
7.52.1 Description détaillée	505
7.52.2 Documentation des constructeurs et destructeur	505
7.52.3 Documentation des fonctions membres	506
7.52.4 Documentation des données membres	533
7.53 Référence de la classe ayoaron ::physic ::CWorld	535
7.53.1 Description détaillée	536
7.53.2 Documentation des constructeurs et destructeur	536
7.53.3 Documentation des fonctions membres	537
7.53.4 Documentation des données membres	539
7.54 Référence de la classe ayoaron ::reseau ::CInternetNetworkInterface	540
7.54.1 Description détaillée	541
7.54.2 Documentation des énumérations membres	541
7.54.3 Documentation des constructeurs et destructeur	542
7.54.4 Documentation des fonctions membres	542
7.54.5 Documentation des données membres	545
7.55 Référence de la classe ayoaron ::reseau ::CLANNetworkInterface	546
7.55.1 Description détaillée	547
7.55.2 Documentation des énumérations membres	547
7.55.3 Documentation des constructeurs et destructeur	548
7.55.4 Documentation des fonctions membres	548
7.55.5 Documentation des données membres	552
7.56 Référence de la classe ayoaron ::reseau ::CNetworkEngine	552
7.56.1 Description détaillée	556
7.56.2 Documentation des énumérations membres	556

7.56.3 Documentation des constructeurs et destructeur	556
7.56.4 Documentation des fonctions membres	557
7.56.5 Documentation des données membres	580
7.57 Référence de la classe ayoaron ::reseau ::CNetworkInterface	584
7.57.1 Description détaillée	585
7.57.2 Documentation des énumérations membres	586
7.57.3 Documentation des constructeurs et destructeur	586
7.57.4 Documentation des fonctions membres	586
7.57.5 Documentation des données membres	589
7.58 Référence de la classe ayoaron ::reseau ::CNetworkMachine	590
7.58.1 Description détaillée	591
7.58.2 Documentation des constructeurs et destructeur	591
7.58.3 Documentation des fonctions membres	592
7.58.4 Documentation des données membres	595
7.59 Référence de la classe ayoaron ::reseau ::CNetworkPlayer	597
7.59.1 Description détaillée	598
7.59.2 Documentation des constructeurs et destructeur	598
7.59.3 Documentation des fonctions membres	598
7.59.4 Documentation des données membres	599
7.60 Référence de la classe game_round	599
7.60.1 Description détaillée	599
7.61 Référence de la classe round_options	599
7.61.1 Description détaillée	600
8 Documentation des fichiers	600
8.1 Référence du fichier CAiPlayer.cpp	600
8.1.1 Description détaillée	601
8.2 Référence du fichier CAiPlayer.h	602
8.2.1 Description détaillée	603
8.3 Référence du fichier CAsteroide.cpp	603
8.3.1 Description détaillée	604
8.4 Référence du fichier CAsteroide.h	604
8.4.1 Description détaillée	605
8.5 Référence du fichier CAvailableAi.h	605

8.5.1	Description détaillée	606
8.6	Référence du fichier CAvailableMap.h	607
8.6.1	Description détaillée	608
8.7	Référence du fichier CBoiteNoire.cpp	608
8.7.1	Description détaillée	609
8.8	Référence du fichier CBoiteNoire.h	609
8.8.1	Description détaillée	611
8.9	Référence du fichier CConfigFile.cpp	611
8.9.1	Description détaillée	612
8.10	Référence du fichier CConfigFile.h	612
8.10.1	Description détaillée	613
8.11	Référence du fichier CConsole.cpp	613
8.11.1	Description détaillée	614
8.12	Référence du fichier CConsole.h	614
8.12.1	Description détaillée	615
8.13	Référence du fichier CData.cpp	615
8.13.1	Description détaillée	616
8.14	Référence du fichier CData.h	616
8.14.1	Description détaillée	618
8.15	Référence du fichier CDataNode.cpp	618
8.15.1	Description détaillée	619
8.16	Référence du fichier CDataNode.h	619
8.16.1	Description détaillée	621
8.17	Référence du fichier CEffort.cpp	621
8.17.1	Description détaillée	621
8.18	Référence du fichier CEffort.h	622
8.18.1	Description détaillée	623
8.19	Référence du fichier CEngine.cpp	623
8.19.1	Description détaillée	623
8.20	Référence du fichier CEngine.h	624
8.20.1	Description détaillée	625
8.21	Référence du fichier CEngineEvent.h	625
8.21.1	Description détaillée	627
8.22	Référence du fichier CEventHandler.h	627

8.22.1	Description détaillée	628
8.23	Référence du fichier CFileManager.cpp	628
8.23.1	Description détaillée	629
8.24	Référence du fichier CFileManager.h	629
8.24.1	Description détaillée	631
8.25	Référence du fichier CGalaxie.cpp	631
8.25.1	Description détaillée	631
8.26	Référence du fichier CGalaxie.h	632
8.26.1	Description détaillée	633
8.27	Référence du fichier CGame.cpp	633
8.27.1	Description détaillée	634
8.28	Référence du fichier CGame.h	634
8.28.1	Description détaillée	635
8.29	Référence du fichier CGameEngine.cpp	636
8.29.1	Description détaillée	636
8.30	Référence du fichier CGameEngine.h	637
8.30.1	Description détaillée	638
8.31	Référence du fichier CGameEventHandler.cpp	638
8.31.1	Description détaillée	639
8.32	Référence du fichier CGameEventHandler.h	639
8.32.1	Description détaillée	640
8.33	Référence du fichier CGameRound.cpp	640
8.33.1	Description détaillée	641
8.34	Référence du fichier CGameRound.h	642
8.34.1	Description détaillée	643
8.35	Référence du fichier CGauge.cpp	643
8.35.1	Description détaillée	644
8.36	Référence du fichier CGauge.h	644
8.36.1	Description détaillée	646
8.37	Référence du fichier CGaugeMoteur.cpp	646
8.37.1	Description détaillée	647
8.38	Référence du fichier CGaugeMoteur.h	647
8.38.1	Description détaillée	648
8.39	Référence du fichier CGaugeVitesse.cpp	648

8.39.1	Description détaillée	649
8.40	Référence du fichier CGaugeVitesse.h	650
8.40.1	Description détaillée	651
8.41	Référence du fichier CGraphicsEngine.cpp	651
8.41.1	Description détaillée	652
8.42	Référence du fichier CGraphicsEngine.h	652
8.42.1	Description détaillée	654
8.43	Référence du fichier CGraphicsOptions.h	654
8.43.1	Description détaillée	655
8.44	Référence du fichier CGuiSystem.cpp	655
8.44.1	Description détaillée	656
8.45	Référence du fichier CGuiSystem.h	656
8.45.1	Description détaillée	658
8.46	Référence du fichier CInternetNetworkInterface.cpp	658
8.46.1	Description détaillée	659
8.47	Référence du fichier CInternetNetworkInterface.h	659
8.47.1	Description détaillée	660
8.48	Référence du fichier CLANNetworkInterface.cpp	660
8.48.1	Description détaillée	661
8.49	Référence du fichier CLANNetworkInterface.h	662
8.49.1	Description détaillée	663
8.50	Référence du fichier CLogger.cpp	663
8.50.1	Description détaillée	664
8.51	Référence du fichier CLogger.h	664
8.51.1	Description détaillée	665
8.52	Référence du fichier CMenuEventHandler.cpp	666
8.52.1	Description détaillée	666
8.53	Référence du fichier CMenuEventHandler.h	666
8.53.1	Description détaillée	667
8.54	Référence du fichier CModulePilotage.cpp	668
8.54.1	Description détaillée	668
8.55	Référence du fichier CModulePilotage.h	668
8.55.1	Description détaillée	669
8.56	Référence du fichier CMoteur.cpp	670

8.56.1	Description détaillée	670
8.57	Référence du fichier CMoteur.h	670
8.57.1	Description détaillée	671
8.58	Référence du fichier CNetworkEngine.cpp	672
8.58.1	Description détaillée	672
8.59	Référence du fichier CNetworkEngine.h	673
8.59.1	Description détaillée	674
8.59.2	Documentation des macros	675
8.60	Référence du fichier CNetworkInterface.cpp	675
8.60.1	Description détaillée	675
8.61	Référence du fichier CNetworkInterface.h	676
8.61.1	Description détaillée	677
8.62	Référence du fichier CNetworkMachine.cpp	677
8.62.1	Description détaillée	678
8.63	Référence du fichier CNetworkMachine.h	678
8.63.1	Description détaillée	680
8.64	Référence du fichier CNetworkPlayer.cpp	680
8.64.1	Description détaillée	681
8.65	Référence du fichier CNetworkPlayer.h	682
8.65.1	Description détaillée	683
8.66	Référence du fichier CObjet.cpp	683
8.66.1	Description détaillée	684
8.67	Référence du fichier CObjet.h	684
8.67.1	Description détaillée	685
8.68	Référence du fichier CParameterAnalyser.cpp	685
8.68.1	Description détaillée	686
8.69	Référence du fichier CParameterAnalyser.h	686
8.69.1	Description détaillée	688
8.70	Référence du fichier CPhysicEngine.cpp	688
8.70.1	Description détaillée	688
8.71	Référence du fichier CPhysicEngine.h	689
8.71.1	Description détaillée	689
8.72	Référence du fichier CPlayer.cpp	690
8.72.1	Description détaillée	690

8.73 Référence du fichier CPlayer.h	691
8.73.1 Description détaillée	692
8.74 Référence du fichier CPlayerOptions.h	692
8.74.1 Description détaillée	693
8.75 Référence du fichier creaobjet.cpp	693
8.75.1 Description détaillée	694
8.76 Référence du fichier creaobjet.h	694
8.76.1 Description détaillée	695
8.77 Référence du fichier CRoundOptions.h	696
8.77.1 Description détaillée	697
8.78 Référence du fichier CSceneGraph.cpp	697
8.78.1 Description détaillée	698
8.79 Référence du fichier CSceneGraph.h	698
8.79.1 Description détaillée	699
8.80 Référence du fichier CSecteur.cpp	699
8.80.1 Description détaillée	700
8.81 Référence du fichier CSecteur.h	700
8.81.1 Description détaillée	701
8.82 Référence du fichier CShared.cpp	702
8.82.1 Description détaillée	702
8.83 Référence du fichier CShared.h	703
8.83.1 Description détaillée	704
8.84 Référence du fichier CSolideRigide.cpp	704
8.84.1 Description détaillée	705
8.85 Référence du fichier CSolideRigide.h	705
8.86 Référence du fichier CSound.cpp	706
8.86.1 Description détaillée	706
8.87 Référence du fichier CSound.h	707
8.87.1 Description détaillée	708
8.88 Référence du fichier CSoundEngine.cpp	708
8.88.1 Description détaillée	708
8.89 Référence du fichier CSoundEngine.h	709
8.89.1 Description détaillée	710
8.90 Référence du fichier CSoundOptions.h	710

8.90.1	Description détaillée	711
8.91	Référence du fichier CVaisseau.cpp	711
8.91.1	Description détaillée	712
8.92	Référence du fichier CVaisseau.h	712
8.92.1	Description détaillée	713
8.93	Référence du fichier CVaisseauPanel.cpp	713
8.93.1	Description détaillée	714
8.94	Référence du fichier CVaisseauPanel.h	714
8.94.1	Description détaillée	715
8.95	Référence du fichier CVaisseauX1.cpp	715
8.95.1	Description détaillée	716
8.96	Référence du fichier CVaisseauX1.h	716
8.96.1	Description détaillée	717
8.97	Référence du fichier CWorld.cpp	718
8.97.1	Description détaillée	718
8.98	Référence du fichier CWorld.h	718
8.98.1	Description détaillée	719
8.99	Référence du fichier CXMLExtractor.cpp	720
8.99.1	Description détaillée	720
8.100	Référence du fichier CXMLExtractor.h	720
8.100.1	Description détaillée	722
8.101	Référence du fichier main.cpp	722
8.101.1	Description détaillée	723
8.101.2	Documentation des fonctions	723

1 Liste des choses à faire

Classe ayoaron ::core ::CAvailableMap il faudra enrichir cette classe pour le traitement d'autres resources...

Membre ayoaron ::core ::CConsole ::CConsole(CGameEngine *) rendre ssl

Membre ayoaron ::core ::CConsole ::processCommand(std ::string) rajouter des commandes

Membre ayoaron ::core ::CConsole ::serverThreadTCPReceive(boost ::asio ::ip ::tcp ::socket *le) fonction à repérer

Membre ayoaron ::core ::CGameRound ::isFull() check du remplissage de la partie

Membre ayoaron ::core ::CGameRound ::loadMap(const std ::string &fileconf)
rendre dynamique la gestion du nom de la map
rendre dynamique le chargement des objets

Membre ayoaron ::data ::CFileManager ::readConfigFile(std ::string filename)
ajouter d'autres catégories

Membre ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(core ::CGame *, bool)
try to add the data directory to the filesystem or something like that

Membre ayoaron ::graphic ::CGuiSystem ::displayAide(int) gestion du ok
gestion du ok

Membre ayoaron ::graphic ::CGuiSystem ::displayIngameMenu() cabler les autres boutons

Membre ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal() cabler les autres boutons
ajouter le menu de chargement

Membre ayoaron ::graphic ::CGuiSystem ::displayOptions(int) rendre effective cette procedure
gestion des données d'options
gestion du ok
gestion du ok

Membre ayoaron ::physic ::CObjet ::bougeObjet(irr ::core ::vector3df translation=irr ::core ::vector3df(0, 0, 0))
la camera est fixe par rapport à l'objet, la rendre bougeante

Membre ayoaron ::physic ::CObjet ::getAbsoluteRotVelocity() include la rotation du parent

Membre ayoaron ::physic ::CObjet ::loadObjetData(data ::CDataNode *d) rendre disponible plus de paramètres de texture
faire le lien avec les options globales de traitement

Membre ayoaron ::physic ::CSolideRigide ::loadObjetData(data ::CDataNode *d=NULL)
rendre disponible plus de paramètres de texture
faire le lien avec les options globales de traitement

Membre ayoaron ::physic ::CWorld ::frame(double dt) gestion des collisions

Membre ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive(boost ::asio ::ip ::tcp ::socket *)
send password by ssl, so on another socket

2 Index des espaces de nommage

2.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description :

ayoaron (L'espace de nommage de base du projet ayoaron)	12
ayoaron : :audio (Espace de nommage contenant tous les objets destiné au moteur de son)	12
ayoaron : :core (Espace de nommage contenant tous les objets de base)	13
ayoaron : :data	14
ayoaron : :event (Espace de nommage contenant tous les objets destiné à la gestion des événements)	16
ayoaron : :graphic (Espace de nommage contenant tous les objets destiné au moteur graphique)	17
ayoaron : :objects	17
ayoaron : :physic	19
ayoaron : :reseau (Espace de nommage contenant tous les objets destiné au moteur réseau)	19
physique (Espace de nommage relatif à la physique du jeu)	20

3 Index des classes

3.1 Hiérarchie des classes

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

ayoaron : :audio : :CSound	20
ayoaron : :audio : :CSoundOptions	48
ayoaron : :core : :CAvailableAi	51
ayoaron : :core : :CAvailableMap	52
ayoaron : :core : :CConsole	54
ayoaron : :core : :CEngine	58
ayoaron : :audio : :CSoundEngine	27

ayoaron ::core ::CGameEngine	79
ayoaron ::graphic ::CGraphicsEngine	225
ayoaron ::physic ::CPhysicEngine	477
ayoaron ::reseau ::CNetworkEngine	552
ayoaron ::core ::CGame	74
ayoaron ::core ::CGameRound	99
ayoaron ::core ::CLogger	107
ayoaron ::core ::CParameterAnalyser	110
ayoaron ::core ::CPlayer	112
ayoaron ::core ::CAiPlayer	49
ayoaron ::reseau ::CNetworkPlayer	597
ayoaron ::core ::CPlayerOptions	115
ayoaron ::core ::CRoundOptions	117
ayoaron ::core ::CShared	119
ayoaron ::core ::CSharedInstance	123
ayoaron ::core ::CXMLExtractor	125
ayoaron ::data ::CConfigFile	127
ayoaron ::data ::CData	130
ayoaron ::data ::CDataNode	146
ayoaron ::data ::CDataStructure	173
ayoaron ::data ::CFileManager	175
ayoaron ::data ::SConfigFileData	184
ayoaron ::event ::CEngineEvent	185
ayoaron ::event ::CEventHandler	188
ayoaron ::event ::CGameEventHandler	192
ayoaron ::event ::CMenuEventHandler	196
ayoaron ::event ::CSerializableVector3df	200

ayoaron ::graphic ::CGauge	201
ayoaron ::graphic ::CGaugeMoteur	207
ayoaron ::graphic ::CGaugeVitesse	214
ayoaron ::graphic ::CGraphicsOptions	248
ayoaron ::graphic ::CGuiSystem	250
ayoaron ::graphic ::CSceneGraph	267
ayoaron ::graphic ::CVaisseauPanel	274
ayoaron ::graphic ::datamot	282
ayoaron ::objets ::CBoiteNoire	317
ayoaron ::objets ::CModulePilotage	321
ayoaron ::objets ::CPousMouv	365
ayoaron ::physic ::CEffort	453
ayoaron ::physic ::CGalaxie	457
ayoaron ::physic ::CObjet	463
ayoaron ::physic ::CSolideRigide	502
ayoaron ::objets ::CAsteroide	282
ayoaron ::objets ::CMoteur	325
ayoaron ::objets ::CVaisseau	368
ayoaron ::objets ::CVaisseauX1	409
ayoaron ::physic ::CWorld	535
ayoaron ::physic ::CSecteur	493
ayoaron ::reseau ::CNetworkInterface	584
ayoaron ::reseau ::CInternetNetworkInterface	540
ayoaron ::reseau ::CLANNetworkInterface	546
ayoaron ::reseau ::CNetworkMachine	590
game_round	599
round_options	599

4 Index des classes

4.1 Liste des classes

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

ayoaron ::audio ::CSound	20
ayoaron ::audio ::CSoundEngine (Le moteur de son)	27
ayoaron ::audio ::CSoundOptions (Classe des options sonore)	48
ayoaron ::core ::CAiPlayer (Les pnj)	49
ayoaron ::core ::CAvailableAi (Contient le nom du script)	51
ayoaron ::core ::CAvailableMap (La liste des maps)	52
ayoaron ::core ::CConsole (Classe contenant la CConsole à faire apparaître en jeu)	54
ayoaron ::core ::CEngine (Classe de base des moteurs)	58
ayoaron ::core ::CGame (Classe de base du jeu)	74
ayoaron ::core ::CGameEngine (Le moteur de jeu)	79
ayoaron ::core ::CGameRound	99
ayoaron ::core ::CLogger (Module de log)	107
ayoaron ::core ::CParameterAnalyser (Classe d'analyse des paramètres)	110
ayoaron ::core ::CPlayer (L'objet joueur)	112
ayoaron ::core ::CPlayerOptions (Rassemblement des options joueur)	115
ayoaron ::core ::CRoundOptions	117
ayoaron ::core ::CShared (Singleton)	119
ayoaron ::core ::CSharedInstance (Donnée stockée dans le singleton)	123
ayoaron ::core ::CXMLExtractor (Cree les interfaces XML)	125
ayoaron ::data ::CConfigFile (Traitement des fichiers de config)	127
ayoaron ::data ::CData (Classe de manipulation de données)	130
ayoaron ::data ::CDataNode (Un bloc de donnée noeud de l'arbre)	146
ayoaron ::data ::CDataStructure (Rassemblement des structures de données)	173

ayoaron : :data : :CFileManager (Classe de gestion des fichiers XML notamment)	175
ayoaron : :data : :SConfigFileData (Conteneur de donnée fichier config)	184
ayoaron : :event : :CEngineEvent (Classe générique pour manipuler les événement entre moteurs)	185
ayoaron : :event : :CEventHandler (Classe générique de gestionnaire d'événement)	188
ayoaron : :event : :CGameEventHandler (Gestionnaire d'événement du jeu)	192
ayoaron : :event : :CMenuEventHandler (Gestionnaire d'événement du menu)	196
ayoaron : :event : :CSerializableVector3df (Classe vecteur sérialisable)	200
ayoaron : :graphic : :CGauge (Classe d'encapsulation des compteurs)	201
ayoaron : :graphic : :CGaugeMoteur (Classe de dessin de l'écran moteur)	207
ayoaron : :graphic : :CGaugeVitesse (La classe de gestion du compteur de vitesse à six axes)	214
ayoaron : :graphic : :CGraphicsEngine (Le moteur graphique)	225
ayoaron : :graphic : :CGraphicsOptions (Résumé des options graphiques)	248
ayoaron : :graphic : :CGuiSystem (Class traitant entièrement l'interface utilisateur (avec CEGUI) chaque événement déclenche un handle_XX , appelant une fonction XX)	250
ayoaron : :graphic : :CSceneGraph (Le graph de scène)	267
ayoaron : :graphic : :CVaisseauPanel (Une classe englobante permettant le dessin complet du tableau de bord)	274
ayoaron : :graphic : :datamot	282
ayoaron : :objets : :CAsteroide (Une classe de description d'un astéroïde)	282
ayoaron : :objets : :CBoiteNoire (L'élément de boite-noire)	317
ayoaron : :objets : :CModulePilotage (Une classe de module de pilotage)	321
ayoaron : :objets : :CMoteur (Classe de description des moteurs)	325
ayoaron : :objets : :CPousMouv (Une classe de vecteur un peu particulière spécifique au pilotage des moteurs)	365

ayoaron ::objets ::CVaisseau (Classe de basse de description d'un vaisseau quelconque)	368
ayoaron ::objets ::CVaisseauX1	409
ayoaron ::physic ::CEffort (Classe permettant de définir un effort appliqué à un objet)	453
ayoaron ::physic ::CGalaxie (La classe décrivant la galaxie)	457
ayoaron ::physic ::CObjet (Classe de base de tous les objets)	463
ayoaron ::physic ::CPhysicEngine (Classe de description du moteur de physique)	477
ayoaron ::physic ::CSecteur (La classe décrivant un secteur de la galaxie)	493
ayoaron ::physic ::CSolideRigide (Une classe d'objet physique)	502
ayoaron ::physic ::CWorld (Classe de définition d'un monde d'objet)	535
ayoaron ::reseau ::CIinternetNetworkInterface (Interface réseau internet)	540
ayoaron ::reseau ::CLANNetworkInterface (Interface réseau LAN)	546
ayoaron ::reseau ::CNetworkEngine (Moteur réseau)	552
ayoaron ::reseau ::CNetworkInterface (Interface réseau)	584
ayoaron ::reseau ::CNetworkMachine (Une machine réseau)	590
ayoaron ::reseau ::CNetworkPlayer (L'objet joueur réseau)	597
game_round (Classe de gestion de la partie)	599
round_options (Rassemblement d'option de partie)	599

5 Index des fichiers

5.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

CAiPlayer.cpp (Implémentation de la classe CAiPlayer)	600
CAiPlayer.h (Déclaration de la classe CAiPlayer)	602
CAsteroide.cpp (Fichier de description d'un astéroïde)	603
CAsteroide.h (Fichier de description d'un astéroïde)	604

CAvailableAi.h (Déclaration de la classe CAvailableAi)	605
CAvailableMap.h (Déclaration de la classe CAvailableMap)	607
CBoiteNoire.cpp	608
CBoiteNoire.h (Un element essentiel de vaisseau : la boite noire)	609
CConfigFile.cpp (Implémentation de la classe CConfigFile)	611
CConfigFile.h (Déclaration de la classe CConfigFile)	612
CConsole.cpp (Implémentation de la classe CConsole)	613
CConsole.h (Déclaration de la classe CConsole)	614
CData.cpp	615
CData.h	616
CDataNode.cpp	618
CDataNode.h	619
CEffort.cpp	621
CEffort.h	622
CEngine.cpp (Implémentation de la classe CEngine)	623
CEngine.h (Déclaration de la classe CEngine)	624
CEngineEvent.h (Déclaration de la classe CEngineEvent et CSerializableVector3df)	625
CEventHandler.h (Déclaration de la classe CEventHandler)	627
CFileManager.cpp	628
CFileManager.h	629
CGalaxie.cpp (La classe décrivant la galaxie)	631
CGalaxie.h (La classe décrivant la galaxie)	632
CGame.cpp (Implémentation de la classe game)	633
CGame.h (Déclaration de la classe CGame)	634
CGameEngine.cpp (Implémentation de la classe CGameEngine)	636
CGameEngine.h (Déclaration de la classe CGameEngine)	637
CGameEventHandler.cpp (Déclaration de la classe CGameEventHandler)	638

CGameEventHandler.h (Déclaration de la classe CGameEventHandler)	639
CGameRound.cpp (Implémentation de la classe CGameRound)	640
CGameRound.h (Déclaration de la classe CGameRound)	642
CGauge.cpp	643
CGauge.h	644
CGaugeMoteur.cpp	646
CGaugeMoteur.h	647
CGaugeVitesse.cpp	648
CGaugeVitesse.h	650
CGraphicsEngine.cpp (Implémentation de la classe CGraphicsEngine)	651
CGraphicsEngine.h (Déclaration de la classe CGraphicsEngine)	652
CGraphicsOptions.h (Déclaration de la classe CGraphicsOptions)	654
CGuiSystem.cpp (Implémentation de la classe CGuiSystem)	655
CGuiSystem.h (Déclaration de la classe CGuiSystem)	656
CInternetNetworkInterface.cpp (Déclaration de la classe CInternetNetworkInterface)	658
CInternetNetworkInterface.h (Déclaration de la classe CInternetNetworkInterface)	659
CLANNetworkInterface.cpp (Déclaration de la classe CLANNetworkInterface)	660
CLANNetworkInterface.h (Déclaration de la classe CLANNetworkInterface)	662
CLogger.cpp (Implémentation de la classe CLogger)	663
CLogger.h (Déclaration de la classe CLogger)	664
CMenuEventHandler.cpp (Déclaration de la classe CMenuEventHandler)	666
CMenuEventHandler.h (Déclaration de la classe CMenuEventHandler)	666
CModulePilotage.cpp	668
CModulePilotage.h	668
CMoteur.cpp	670

CMoteur.h	670
CNetworkEngine.cpp (Déclaration de la classe <code>network_engine</code>)	672
CNetworkEngine.h (Déclaration de la classe <code>network_engine</code>)	673
CNetworkInterface.cpp (Implémentation de la classe <code>CNetworkInterface</code>)	675
CNetworkInterface.h (Déclaration de la classe <code>CNetworkInterface</code>)	676
CNetworkMachine.cpp (Declaration de la classe <code>CNetworkMachine</code>)	677
CNetworkMachine.h (Declaration de la classe <code>CNetworkMachine</code>)	678
CNetworkPlayer.cpp (Declaration de la classe <code>CNetworkPlayer</code>)	680
CNetworkPlayer.h (Declaration de la classe <code>CNetworkPlayer</code>)	682
COBJET.cpp	683
COBJET.h	684
CParameterAnalyser.cpp (Implémentation de la classe d'analyse de paramètres)	685
CParameterAnalyser.h (Declaration de la classe d'analyse de paramètres)	686
CPhysicEngine.cpp (Fichier de description du poteur de physique)	688
CPhysicEngine.h (Fichier de description du poteur de physique)	689
CPlayer.cpp (Implementation de la classe <code>CPlayer</code>)	690
CPlayer.h (Declaration de la classe <code>player</code>)	691
CPlayerOptions.h (Declaration de la classe <code>CPlayerOptions</code>)	692
creaobjet.cpp	693
creaobjet.h	694
CRoundOptions.h (Declaration de la classe <code>CRoundOptions</code>)	696
CSceneGraph.cpp (Declaration de la classe <code>CSceneGraph</code>)	697
CSceneGraph.h (Declaration de la classe <code>CSceneGraph</code>)	698
CSecteur.cpp (La classe décrivant un secteur de la galaxie)	699
CSecteur.h (La classe décrivant un secteur de la galaxie)	700
CShared.cpp (Implémentation de la classe <code>CShared</code> et <code>CSharedInstance</code>)	702
CShared.h (Declaration de la classe <code>CShared</code> et <code>CSharedInstance</code>)	703

CSolideRigide.cpp	704
CSolideRigide.h	705
CSound.cpp	706
CSound.h	707
CSoundEngine.cpp (Declaration de la classe CSoundEngine)	708
CSoundEngine.h (Declaration de la classe CSoundEngine)	709
CSoundOptions.h	710
CVaisseau.cpp	711
CVaisseau.h	712
CVaisseauPanel.cpp	713
CVaisseauPanel.h	714
CVaisseauX1.cpp	715
CVaisseauX1.h	716
CWorld.cpp	718
CWorld.h	718
CXMLExtractor.cpp (Déclaration de la classe CXMLExtractor)	720
CXMLExtractor.h (Déclaration de la classe CXMLExtractor)	720
main.cpp (Partie principale du programme)	722

6 Documentation des espaces de nommage

6.1 Référence de l'espace de nommage ayoaron

l'espace de nomage de base du projet ayoaron

Espaces de nommage

- namespace **audio**
espace de nomage contenant tous les objets destiné au moteur de son
- namespace **core**
espace de nomage contenant tous les objets de base
- namespace **data**
- namespace **event**

espace de nomage contenant tous les objets destiné à la gestion des événements

- namespace **graphic**

espace de nomage contenant tous les objets destiné au moteur graphique

- namespace **objets**

- namespace **physic**

- namespace **reseau**

espace de nomage contenant tous les objets destiné au moteur réseau

6.1.1 Description détaillée

l'espace de nomage de base du projet ayoaron

6.2 Référence de l'espace de nommage ayoaron : :audio

espace de nomage contenant tous les objets destiné au moteur de son

Classes

- class **CSound**
- class **CSoundEngine**
 - le moteur de son*
- class **CSoundOptions**
 - classe des options sonore*

Énumérations

- enum **SOUND_TYPE** { **SND_2D_INSTANT**, **SND_2D_MUSIC**, **SND_3D_CONTINUOUS**, **SND_3D_INSTANT** }

6.2.1 Description détaillée

espace de nomage contenant tous les objets destiné au moteur de son

6.2.2 Documentation du type de l'énumération

6.2.2.1 enum ayoaron : :audio : :SOUND_TYPE

Valeurs énumérées :

SND_2D_INSTANT
SND_2D_MUSIC
SND_3D_CONTINUOUS
SND_3D_INSTANT

6.3 Référence de l'espace de nommage ayoaron : :core

espace de nomage contenant tous les objets de base

Classes

- class **CAiPlayer**
les pnj
- class **CAvailableAi**
contient le nom du script
- class **CAvailableMap**
La liste des maps.
- class **CConsole**
*classe contenant la **CConsole** à faire apparaître en jeu*
- class **CEngine**
Classe de base des moteurs.
- class **CGame**
classe de base du jeu
- class **CGameEngine**
Le moteur de jeu.
- class **CGameRound**
- class **CLogger**
module de log
- class **CParameterAnalyser**
Classe d'analyse des paramètres.
- class **CPlayer**
l'objet joueur
- class **CPlayerOptions**
rassemblement des options joueur
- class **CRoundOptions**
- class **CShared**
singleton
- class **CSharedInstance**
donnée stockée dans le singleton
- class **CXMLExtractor**
cree les interfaces XML

6.3.1 Description détaillée

espace de nommage contenant tous les objets de base

6.4 Référence de l'espace de nommage ayoaron : :data

Classes

- class **CConfigFile**
Traitemet des fichiers de config.
- class **CData**
classe de manipulation de données
- class **CDataNode**
un bloc de donnée noeud de l'arbre
- struct **CDataStructure**
rassemblement des structures de données
- class **CFileManager**
classe de gestion des fichiers XML notamment
- struct **SConfigFileData**
conteneur de donnée fichier config

Énumérations

- enum **DATATYPE** {
 NOTYPE, UINT, SINT, FLOAT,
 VECTOR, MATRIX, COLOR, BOOL,
 STRING }

Fonctions

- **DATATYPE str2dtyp** (std ::string str)
- std ::wstring **strtowchar** (const std ::string)
 *fonction de conversion d'un std ::string en wchar_t**
- const std ::string **wchartostr** (const wchar_t *)
 fonction de conversion entre un wchar_t et un std ::string*

6.4.1 Documentation du type de l'énumération

6.4.1.1 enum ayoaron ::data ::DATATYPE

Valeurs énumérées :

NOTYPE NOTYPE.

UINT UINT.

SINT SINT.

FLOAT FLOAT.

VECTOR VECTOR.

MATRIX MATRIX.

COLOR COLOR.

BOOL BOOL.

STRING STRING.

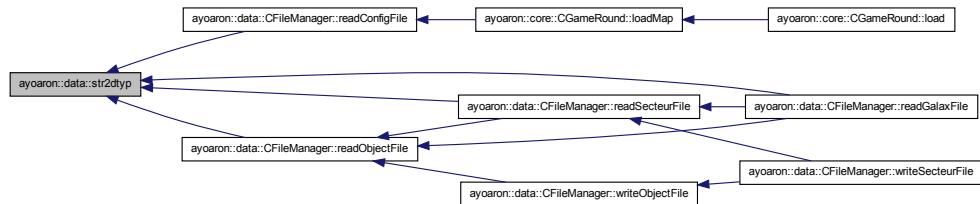
6.4.2 Documentation des fonctions

6.4.2.1 DATATYPE ayoaron ::data ::str2dtyp (std ::string str)

Références BOOL, COLOR, FLOAT, MATRIX, NOTYPE, SINT, STRING, UINT, et VECTOR.

Référencé par ayoaron ::data ::CFileManager ::readConfigFile(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), et ayoaron ::data ::CFileManager ::readSecteurFile().

Voici le graphe d'appel pour cette fonction :

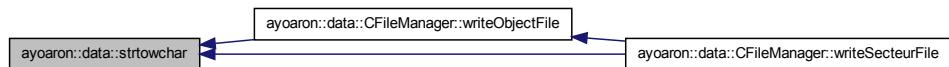


6.4.2.2 std ::wstring ayoaron ::data ::strtowchar (const std ::string str)

fonction de conversion d'un std ::string en wchar_t*

Référencé par ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :

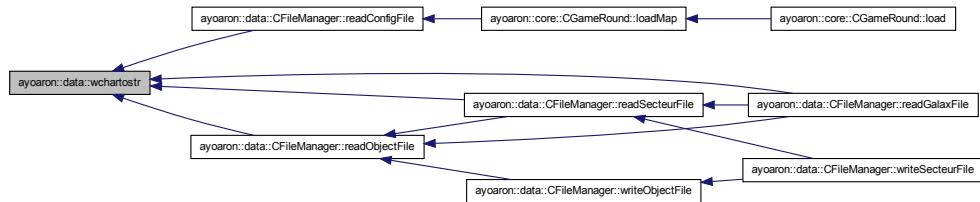


6.4.2.3 const std ::string ayoaron ::data ::wchartostr (const wchar_t * wc)

fonction de conversion entre un wchar_t* et un std ::string

Référencé par ayoaron ::data ::CFileManager ::readConfigFile(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), et ayoaron ::data ::CFileManager ::readSecteurFile().

Voici le graphe d'appel pour cette fonction :



6.5 Référence de l'espace de nommage ayoaron : :event

espace de nomage contenant tous les objets destiné à la gestion des événements

Classes

- class **CEngineEvent**
classe générique pour manipuler les événement entre moteurs
- class **CEventHandler**
classe générique de gestionnaire d'évenement
- class **CGameEventHandler**
gestionnaire d'évenement du jeu
- class **CMenuEventHandler**
gestionnaire d'évenement du menu
- class **CSerializableVector3df**
classe vecteur sérialisable

6.5.1 Description détaillée

espace de nomage contenant tous les objets destiné à la gestion des événements

6.6 Référence de l'espace de nommage ayoaron : :graphic

espace de nomage contenant tous les objets destiné au moteur graphique

Classes

- class **CGauge**
classe d'encapsulation des compteurs
- class **CGaugeMoteur**
classe de dessin de l'écran moteur
- class **CGaugeVitesse**
la classe de gestion du compteur de vitesse à six axes
- class **CGraphicsEngine**
le moteur graphique
- class **CGraphicsOptions**
résumé des options graphiques
- class **CGuiSystem**
classe traitant entièrement l'interface utilisateur (avec CEGUI) chaque événement déclenche un handle_XX , appelant une fonction XX
- class **CSceneGraph**
le graph de scène
- class **CVaisseauPanel**
une classe englobante permettant le dessin complet du tableau de bord
- struct **datamot**

6.6.1 Description détaillée

espace de nomage contenant tous les objets destiné au moteur graphique

6.7 Référence de l'espace de nommage ayoaron ::objets

Classes

- class **CAsteroid**
Une classe de description d'un astéroïde.
- class **CBoiteNoire**
l'élément de boite-noire
- class **CModulePilotage**
une classe de module de pilotage
- class **CMoteur**
classe de description des moteurs
- class **CPousMouv**
une classe de vecteur un peu particulière spécifique au pilotage des moteurs
- class **CVaisseau**
Classe de base de description d'un vaisseau quelconque.
- class **CVaisseauX1**

Fonctions

- **physic ::CSolideRigide * creaobjet (data ::CDataNode *donn, physic ::CWorld *world, physic ::CSolideRigide *parent)**
- **irr ::f32 mround (irr ::f32 a, irr ::u32 off)**
- **irr ::f32 vmin (irr ::f32 v, irr ::f32 eps, irr ::f32 VM)**

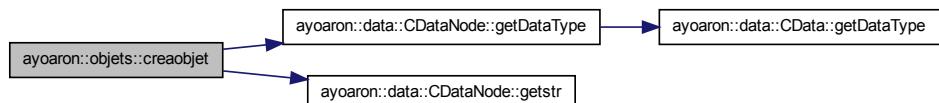
6.7.1 Documentation des fonctions

6.7.1.1 **physic ::CSolideRigide * ayoaron ::objets ::creaobjet (data ::CDataNode * donn, physic ::CWorld * world, physic ::CSolideRigide * parent)**

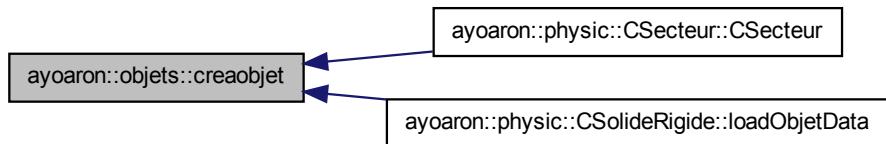
Références ayoaron ::data ::CDataNode ::getDataType(), ayoaron ::data ::CDataNode ::getstr(), et ayoaron ::data ::STRING.

Référencé par ayoaron ::physic ::CSecteur ::CSecteur(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



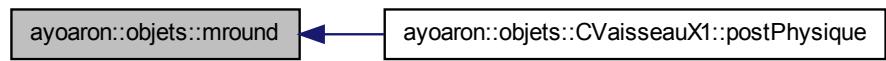
Voici le graphe d'appel pour cette fonction :



6.7.1.2 irr ::f32 ayoaron ::objets ::mround (irr ::f32 a, irr ::u32 off)

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



6.7.1.3 irr ::f32 ayoaron ::objets ::vmin (irr ::f32 v, irr ::f32 eps, irr ::f32 VM)

6.8 Référence de l'espace de nommage ayoaron ::physic

Classes

- class **CEffort**
classe permettant de définir un effort appliqué à un objet
- class **CGalaxie**
la classe décrivant la galaxie
- class **COBJET**
Classe de base de tous les objets.
- class **CPhysicEngine**
classe de description du moteur de physique
- class **CSecteur**
la classe décrivant un secteur de la galaxie
- class **CSolideRigide**
une classe d'objet physique
- class **CWorld**
Classe de définition d'un monde d'objet.

Énumérations

- enum REPERE { LOCAL, PARENT, GENERAL }

6.8.1 Documentation du type de l'énumération

6.8.1.1 enum ayoaron ::physic ::REPERE

Valeurs énumérées :

LOCAL

PARENT

GENERAL

6.9 Référence de l'espace de nommage ayoaron ::reseau

espace de nomage contenant tous les objets destiné au moteur réseau

Classes

- class CInternetNetworkInterface
 - interface réseau internet*
- class CLANNetworkInterface
 - interface réseau LAN*
- class CNetworkEngine
 - moteur réseau*
- class CNetworkInterface
 - interface réseau*
- class CNetworkMachine
 - une machine réseau*
- class CNetworkPlayer
 - l'objet joueur réseau*

6.9.1 Description détaillée

espace de nomage contenant tous les objets destiné au moteur réseau

6.10 Référence de l'espace de nommage physique

espace de nommage relatif à la physique du jeu

6.10.1 Description détaillée

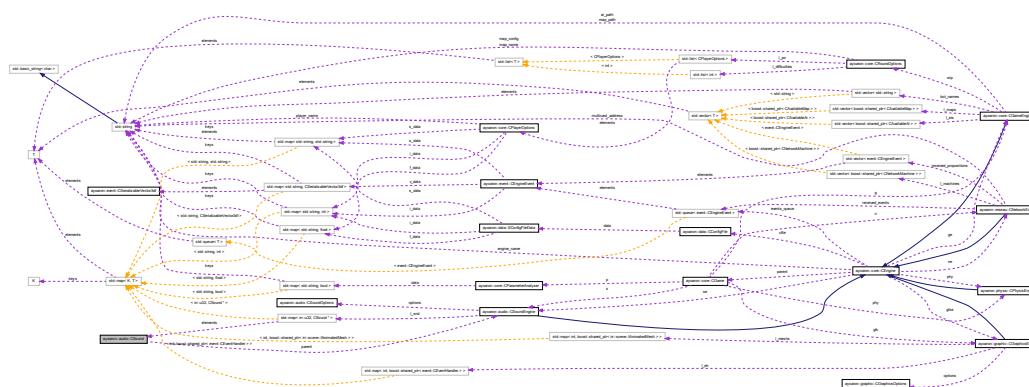
espace de nommage relatif à la physique du jeu

7 Documentation des classes

7.1 Référence de la classe ayoaron::audio::CSound

```
#include <CSound.h>
```

Graphe de collaboration de ayoaron ::audio ::CSound ::



Fonctions membres publiques

- **CSound (CSoundEngine *p)**
Constructeur.
 - **void destrSound ()**
détruit le son (mais pas la classe)
 - **irr : f64 getPitch ()**
définit la vitesse de lecture (seulement si sans effets spéciaux)
 - **irr : core : vector3df getPos ()**
renvoie la position
 - **SOUND_TYPE getType ()**
renvoie le type de son
 - **irr : core : vector3df getVit ()**
renvoie la vitesse
 - **irr : f64 getVolume ()**
renvoie le volume
 - **bool isDoomed ()**
renvoie true si le son doit être détruit
 - **bool isPlaying ()**
renvoie true si le son est en cours de lecture
 - **bool isRepeat ()**
dit si le son est en mode repeat
 - **void loadSound (std : string file)**
charge un son depuis un fichier
 - **void playSound ()**
lance la lecture d'un son
 - **void setPitch (irr : f64 p)**
définit la vitesse de lecture (seulement si sans effets spéciaux)
 - **void setPos (irr : core : vector3df pos=irr : core : vector3df(0, 0, 0))**
renvoie la position de la source sonore
 - **void setType (SOUND_TYPE t)**

- void **setVit** (irr ::core ::vector3df **vit**=irr ::core ::vector3df(0, 0, 0))

renvoie la vitesse de la source sonore
- void **setVolume** (irr ::f64 **vol**)

defini le volume
- void **stopSound** ()

stop le son en cours s'il est joué
- virtual **~CSound** ()

Destructeur.

Attributs privés

- **CSoundEngine * parent**

pointeur vers le moteur parent
- irr ::core ::vector3df **pos**

la position
- irrklang ::ISound * **snd**

pointeur vers le son joué
- irrklang ::ISoundSource * **sndsource**

pointeur vers le la soubndsource associée
- **SOUND_TYPE sndtype**

le type de son
- irr ::core ::vector3df **vit**

la vitesse

7.1.1 Documentation des constructeurs et destructeur

7.1.1.1 ayoaron ::audio ::CSound ::CSound (**CSoundEngine * p**)

Constructeur.

Références parent, snd, ayoaron ::audio ::SND_2D_INSTANT, sndsource, et sndtype.

7.1.1.2 ayoaron ::audio ::CSound ::~CSound () [virtual]

Destructeur.

Références snd, et sndsource.

7.1.2 Documentation des fonctions membres

7.1.2.1 void ayoaron ::audio ::CSound ::destrSound ()

detruit le son (mais pas la classe)

Références snd.

7.1.2.2 irr ::f64 ayoaron ::audio ::CSound ::getPitch ()

definit la vitesse de lecture (seulement si sans effets speciaux)

Références snd.

7.1.2.3 irr ::core ::vector3df ayoaron ::audio ::CSound ::getPos ()

renvoie la position

Références pos, et snd.

7.1.2.4 SOUND_TYPE ayoaron ::audio ::CSound ::getType () [inline]

renvoie le type de son

Références sndtype.

7.1.2.5 irr ::core ::vector3df ayoaron ::audio ::CSound ::getVit ()

renvoie la vitesse

Références snd, et vit.

7.1.2.6 irr ::f64 ayoaron ::audio ::CSound ::getVolume ()

renvoie le volume

Références snd.

7.1.2.7 bool ayoaron ::audio ::CSound ::isDoomed ()

renvoie true si le son dois être détruit

Références snd.

7.1.2.8 bool ayoaron ::audio ::CSound ::isPlaying ()

renvoie true si le son est en cours de lecture

Références snd.

Référencé par ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.1.2.9 bool ayoaron ::audio ::CSound ::isRepeat () [inline]

dit si le son est en mode repeat

Références snd.

7.1.2.10 void ayoaron ::audio ::CSound ::loadSound (std ::string *file*)

charge un son depuis un fichier

Paramètres

in	<i>file</i>	le nom du fichier à charger
----	-------------	-----------------------------

Références ayoaron ::audio ::CSoundEngine ::getDevice(), parent, snd, ayoaron ::audio ::SND_2D_INSTANT, ayoaron ::audio ::SND_2D_MUSIC, ayoaron ::audio ::SND_3D_CONTINUOUS, ayoaron ::audio ::SND_3D_INSTANT, sndsource, et sndtype.

Voici le graphe d'appel pour cette fonction :



7.1.2.11 void ayoaron ::audio ::CSound ::playSound ()

lance la lecture d'un son

Références ayoaron ::audio ::CSoundEngine ::getDevice(), parent, pos, snd, ayoaron ::audio ::SND_2D_INSTANT, ayoaron ::audio ::SND_2D_MUSIC, ayoaron ::audio ::SND_3D_CONTINUOUS, ayoaron ::audio ::SND_3D_INSTANT, sndsource, sndtype, et vit.

Référencé par ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.1.2.12 void ayoaron ::audio ::CSound ::setPitch (irr ::f64 *p*)

definit la vitesse de lecture (seulement si sans effets speciaux)

Références snd.

Référencé par ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.1.2.13 void ayoaron ::audio ::CSound ::setPos (irr ::core ::vector3df *pos* = irr ::core ::vector3df(0, 0, 0))

renvoie la position de la source sonore

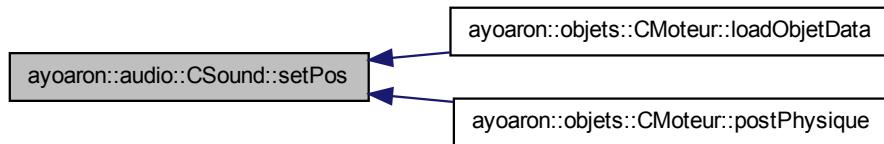
Paramètres

in	<i>pos</i>	le vecteur position
----	------------	---------------------

Références pos, et snd.

Référencé par ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.1.2.14 void ayoaron ::audio ::CSound ::setType (SOUND_TYPE t)

definit le type de son

Références snd, ayoaron ::audio ::SND_2D_INSTANT, ayoaron ::audio ::SND_2D_MUSIC, ayoaron ::audio ::SND_3D_CONTINUOUS, ayoaron ::audio ::SND_3D_INSTANT, et sndtype.

7.1.2.15 void ayoaron ::audio ::CSound ::setVit (irr ::core ::vector3df vit = irr ::core ::vector3df(0, 0, 0))

renvoie la vitesse de la source sonore

Paramètres

in	vit	le vecteur vitesse
----	-----	--------------------

Références snd, et vit.

7.1.2.16 void ayoaron ::audio ::CSound ::setVolume (irr ::f64 vol)

defini le volume

Références snd.

Référencé par ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.1.2.17 void ayoaron ::audio ::CSound ::stopSound ()

stop le son en cours s'il est joué

Références snd, ayoaron ::audio ::SND_2D_INSTANT, ayoaron ::audio ::SND_2D_MUSIC, ayoaron ::audio ::SND_3D_CONTINUOUS, ayoaron ::audio ::SND_3D_INSTANT, et sndtype.

Référencé par ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.1.3 Documentation des données membres

7.1.3.1 CSoundEngine* ayoaron ::audio ::CSound ::parent [private]

pointeur vers le moteur parent

Référencé par CSound(), loadSound(), et playSound().

7.1.3.2 irr ::core ::vector3df ayoaron ::audio ::CSound ::pos [private]

la position

Référencé par getPos(), playSound(), et setPos().

7.1.3.3 irrklang ::ISound* ayoaron ::audio ::CSound ::snd [private]

pointeur vers le son joué

Référencé par CSound(), destrSound(), getPitch(), getPos(), getVit(), getVolume(), isDoomed(), isPlaying(), isRepeat(), loadSound(), playSound(), setPitch(), setPos(), setType(), setVit(), setVolume(), stopSound(), et ~CSound().

7.1.3.4 irrklang ::ISoundSource* ayoaron ::audio ::CSound ::sndsource [private]

pointeur vers la soudndsource associée

Référencé par CSound(), loadSound(), playSound(), et ~CSound().

7.1.3.5 SOUND_TYPE ayoaron ::audio ::CSound ::sndtype [private]

le type de son

Référencé par CSound(), getType(), loadSound(), playSound(), setType(), et stopSound().

7.1.3.6 irr ::core ::vector3df ayoaron ::audio ::CSound ::vit [private]

la vitesse

Référencé par getVit(), playSound(), et setVit().

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

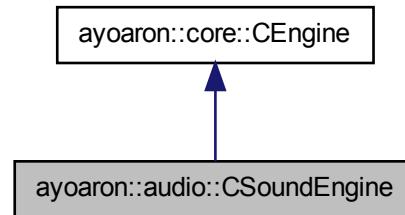
- [CSound.h](#)
- [CSound.cpp](#)

7.2 Référence de la classe ayoaron ::audio ::CSoundEngine

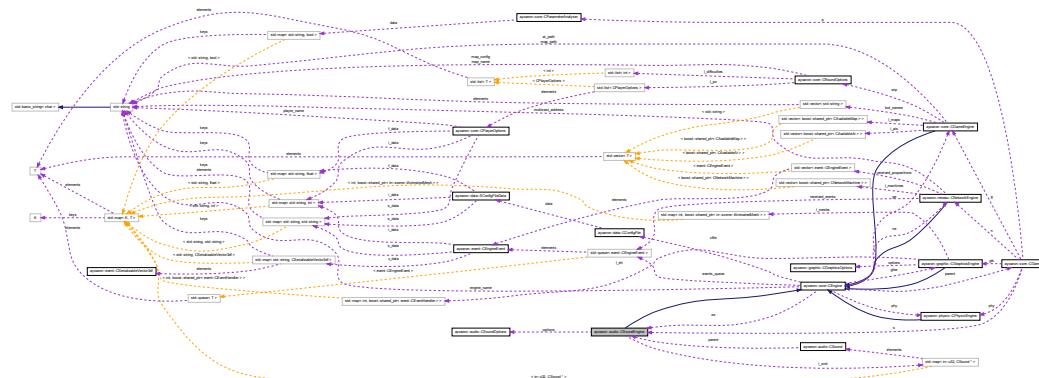
le moteur de son

```
#include <CSoundEngine.h>
```

Graphe d'héritage de ayoaron ::audio ::CSoundEngine :



Graphe de collaboration de ayoaron ::audio ::CSoundEngine :



Types publics

- enum {
 STATE NOTHING, STATE QUITTING, STATE_WAITING_PLAYERS, STATE_PLAYING_INTRO,
STATE_LOADING_ROUND, STATE_PLAYING }

Constantes d'état.

Fonctions membres publiques

- **CSound * addSound (SOUND_TYPE type=SND_2D_INSTANT, std ::string file="", bool start=true, irr ::core ::vector3df pos=irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df vit=irr ::core ::vector3df(0, 0, 0))**
ajoute un son au moteur
- void **attachGameEngine (CGameEngine *e)**
lien avec le moteur de jeu
- void **attachGraphicsEngine (graphic ::CGraphicsEngine *e)**
lien avec le moteur de rendu graphique
- void **attachNetworkEngine (reseau ::CNetworkEngine *e)**
lien avec le moteur de réseau
- void **attachPhysicEngine (physic ::CPhysicEngine *e)**
lien avec le moteur de physique
- void **attachSoundEngine (audio ::CSoundEngine *e)**
lien avec le moteur de son
- **CSoundEngine (core ::CGame *, bool)**
constructeur
- void **dropAllSound ()**
supprime tous les sons
- void **dropSound (irr ::f64 id)**
detruit le son
- void **frame (double dt)**
Methode Frame.
- irrklang ::ISoundEngine * **getDevice ()**
renvoie le device
- CGameEngine * **getGameEngine ()**
lien vers le moteur de jeu
- **graphic ::CGraphicsEngine * getGraphicsEngine ()**
lien vers le moteur de rendu graphique
- **reseau ::CNetworkEngine * getNetworkEngine ()**
lien vers le moteur de réseau
- CGame * **getParent ()**
lien vers la classe game parente
- **CSound * getSound (irr ::f64 id)**
renvoie le son selon son ID
- **audio ::CSoundEngine * getSoundEngine ()**
lien vers le moteur de son
- void **processEvent (event ::CEngineEvent &)**
trate les évenements
- void **processQueue ()**
vidange de l'event_queue
- void **pushEvent (event ::CEngineEvent &e)**
ajout d'un évenement
- void **sendMessageToGame (event ::CEngineEvent &e)**
envoie un message au moteur de jeu
- void **sendMessageToGraphics (event ::CEngineEvent &e)**
envoie un message au moteur de rendu graphique
- void **sendMessageToNetwork (event ::CEngineEvent &e)**

- `void sendMessageToNetwork (event : :CEngineEvent &e)`
envoie un message au moteur de réseau
- `void sendMessageToPhysics (event : :CEngineEvent &e)`
envoie un message au moteur physique
- `void sendMessageToSound (event : :CEngineEvent &e)`
envoie un message au moteur de son
- `void setDopplerProperties (irr : :f64 fac=1.0, irr : :f64 dist=1.0)`
la vitesse doppler
- `void setListenerPosition (irr : :core ::vector3df pos, irr : :core ::vector3df look=irr : :core ::vector3df(0, 0, 1), irr : :core ::vector3df vel=irr : :core ::vector3df(0, 0, 0), irr : :core ::vector3df up=irr : :core ::vector3df(0, 1, 0))`
definit la position de l'écoutant
- `virtual ~CSoundEngine ()`
destructeur

Attributs publics

- `int state`
l'état actuel
- `boost ::mutex state_mutex`
Mutex de protection de l'état.

Fonctions membres protégées

- `void getConfigData ()`
lit le fichier de config

Attributs protégés

- `data : :CConfigFile cfile`
Fichier de configuration.
- `irrklang : :ISoundEngine * device`
le device de son
- `std ::string engine_name`
nom de ce moteur
- `std ::queue< event : :CEngineEvent > events_queue`
liste des événement à traiter
- `CGameEngine * ge`
pointeur vers le moteur de jeu
- `graphic : :CGraphicsEngine * gfxe`
pointeur vers le moteur de rendu graphique
- `std ::map< irr : :u32, CSound * > l_snd`
liste des sons gérés par le moteur
- `reseau : :CNetworkEngine * ne`
pointeur vers le moteur de réseau
- `CSoundOptions options`
options sonores
- `CGame * parent`
pointeur vers le parent
- `bool passive_mode`
true si mode passif activé
- `physic : :CPhysicsEngine * phy`
pointeur vers le moteur de physique
- `boost ::mutex queue_mutex`
Mutex de protection de la liste d'événement event_queue.
- `audio : :CSoundEngine * se`

- pointeur vers le moteur de son*
– irr ::scene ::ISceneManager * **smgr**
manager de scène

7.2.1 Description détaillée

le moteur de son

7.2.2 Documentation des énumérations membres

7.2.2.1 anonymous enum [inherited]

Constantes d'état.

Valeurs énumérées :

STATE NOTHING
STATE QUITTING
STATE_WAITING_PLAYERS
STATE_PLAYING_INTRO
STATE_LOADING_ROUND
STATE_PLAYING

7.2.3 Documentation des constructeurs et destructeur

7.2.3.1 ayoaron ::audio ::CSoundEngine ::CSoundEngine (core ::CGame * g, bool passif)

constructeur

Paramètres

in	<i>g</i>	le jeu parent
in	<i>passif</i>	activation du mode passif

Références device, ayoaron ::audio ::CSoundOptions ::distfac, ayoaron ::audio ::CSoundOptions ::dolpfac, getConfigData(), ayoaron ::audio ::CSoundOptions ::maxdist, ayoaron ::audio ::CSoundOptions ::mindist, options, ayoaron ::core ::CEngine ::passive_mode, ayoaron ::audio ::CSoundOptions ::rolloff, et ayoaron ::audio ::CSoundOptions ::volume.

Voici le graphe d'appel pour cette fonction :



7.2.3.2 ayoaron ::audio ::CSoundEngine ::~CSoundEngine() [virtual]

destructeur

Références device.

7.2.4 Documentation des fonctions membres

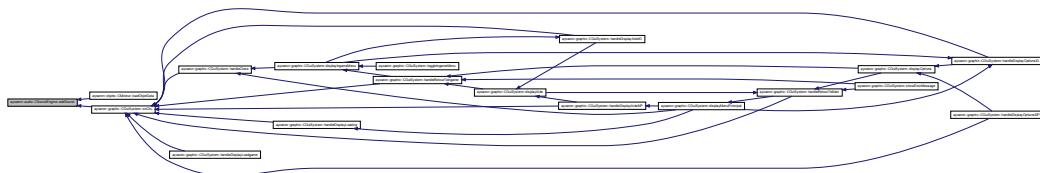
```
7.2.4.1 CSound * ayoaron :audio :CSoundEngine :addSound( SOUND_TYPE type =  
SND_2D_INSTANT, std::string file = " ", bool start = true, irr::core::vector3df  
pos = irr::core::vector3df(0, 0, 0), irr::core::vector3df vit =  
irr::core::vector3df(0, 0, 0) )
```

ajoute un son au moteur

Références I_snd.

Référencé par ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::graphic ::CGui-System ::onClic().

Voici le graphe d'appel pour cette fonction :



7.2.4.2 void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e)
[inline, inherited]

lien avec le moteur de jeu

Paramètres

in e pointeur vers le moteur de jeu à lier

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.2.4.3 void ayoaron ::core ::CEngine ::attachGraphicsEngine (
graphic : CGraphicsEngine * e) [inline, inherited]

lien avec le moteur de rendu graphique

Paramètres

in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.2.4.4 void ayoaron ::core ::CEngine ::attachNetworkEngine (
reseau : CNetworkEngine * e) [inline, inherited]

lien avec le moteur de réseau

Paramètres

in	e	pointeur vers le moteur de réseau à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.2.4.5 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * e) [inline, inherited]

lien avec le moteur de physique

Paramètres

in	e	pointeur vers le moteur de physique à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références ayoaron ::core ::CEngine ::phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.2.4.6 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline, inherited]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.2.4.7 void ayoaron ::audio ::CSoundEngine ::dropAllSound ()

supprime tous les sons

Références dropSound(), et l_snd.

Voici le graphe d'appel pour cette fonction :



7.2.4.8 void ayoaron ::audio ::CSoundEngine ::dropSound (irr ::f64 id)

detruit le son

Références l_snd.

Référencé par dropAllSound().

Voici le graphe d'appel pour cette fonction :



7.2.4.9 void ayoaron ::audio ::CSoundEngine ::frame (double dt) [virtual]

Methode Frame.

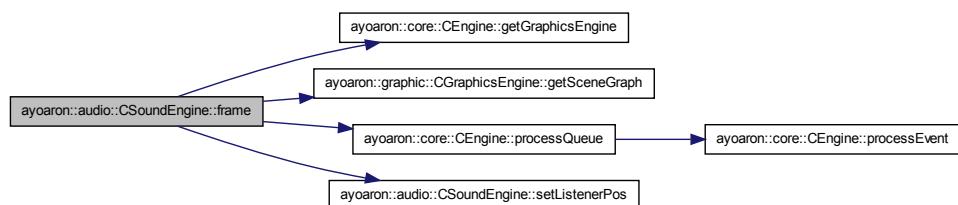
Cette méthode est utiliser pour mettre à jour le moteur de son

Implémente [ayoaron ::core ::CEngine](#).

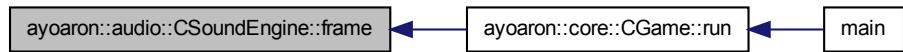
Références [ayoaron ::core ::CEngine ::getGraphicsEngine\(\)](#), [ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph\(\)](#), [l_snd](#), [ayoaron ::core ::CEngine ::processQueue\(\)](#), [setListenerPos\(\)](#), et [smgr](#).

Référencé par [ayoaron ::core ::CGame ::run\(\)](#).

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.2.4.10 void ayoaron ::audio ::CSoundEngine ::getConfigData () [protected, virtual]

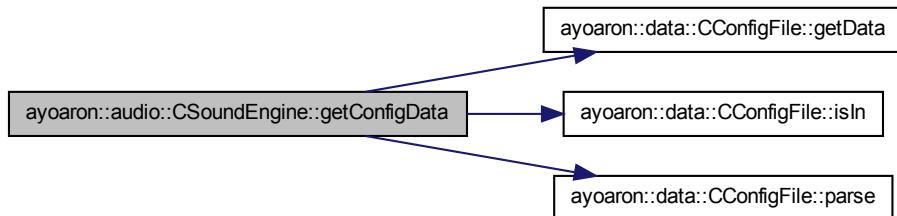
lit le fichier de config

Implémente [ayoaron ::core ::CEngine](#).

Références [ayoaron ::core ::CEngine ::file](#), [ayoaron ::audio ::CSoundOptions ::distfac](#), [ayoaron ::audio ::CSoundOptions ::doplfac](#), [ayoaron ::data ::SConfigFileData ::f_data](#), [ayoaron ::data ::CConfigFile ::getData\(\)](#), [ayoaron ::data ::CConfigFile :::isIn\(\)](#), [ayoaron ::audio ::CSoundOptions ::maxdist](#), [ayoaron ::audio ::CSoundOptions ::mindist](#), [options](#), [ayoaron ::data ::CConfigFile ::parse\(\)](#), [ayoaron ::audio ::CSoundOptions ::rolloff](#), et [ayoaron ::audio ::CSoundOptions ::volume](#).

Référencé par [CSoundEngine\(\)](#).

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



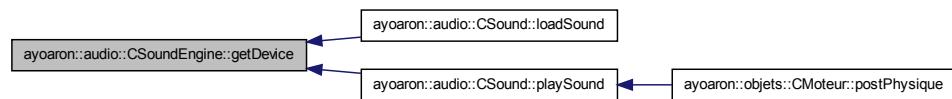
7.2.4.11 `irrklang ::ISoundEngine* ayoaron ::audio ::CSoundEngine ::getDevice ()`
 [inline]

renvoie le device

Références device.

Référencé par ayoaron ::audio ::CSound ::loadSound(), et ayoaron ::audio ::CSound ::playSound().

Voici le graphe d'appel pour cette fonction :



7.2.4.12 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine () [inline, inherited]

lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect(), ayoaron ::graphic ::CGuiSystem ::handleProcessCommand(), ayoaron ::graphic ::CGuiSystem ::loadGame(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::event ::CGameEventHandler ::onEvent(), et ayoaron ::graphic ::CGraphicsEngine ::processEvent().

Voici le graphe d'appel pour cette fonction :



7.2.4.13 graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine () [inline, inherited]

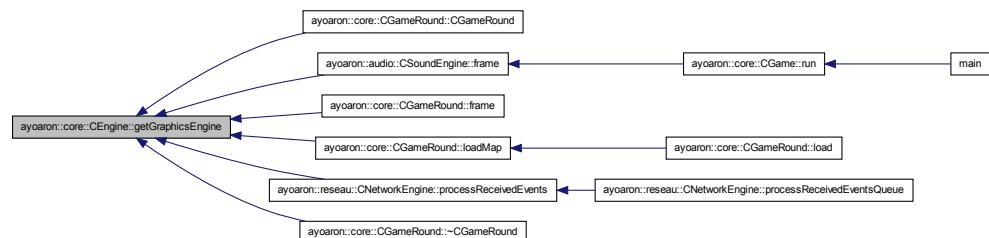
lien vers le moteur de rendu graphique

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), frame(), ayoaron ::core ::CGameRound ::frame(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.2.4.14 reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::getNetworkEngine () [inline, inherited]

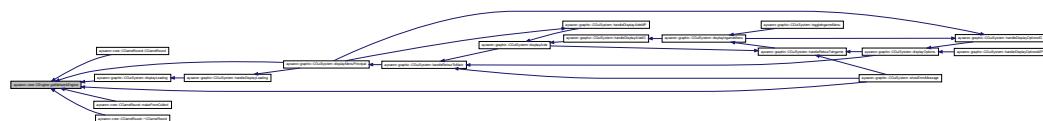
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.2.4.15 CGame* ayoaron ::core ::CEngine ::getParent () [inline, inherited]

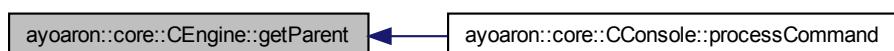
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.2.4.16 CSound * ayoaron ::audio ::CSoundEngine ::getSound (irr ::f64 id)

renvoie le son selon son ID

Références l_snd.

7.2.4.17 audio ::CSoundEngine* ayoaron ::core ::CEngine ::getSoundEngine () [inline, inherited]

lien vers le moteur de son

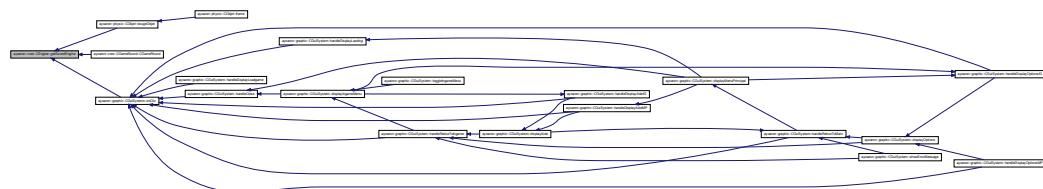
Cette méthode est utiliser pour obtenir un pointeur vers le moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::core ::CGame-

Round ::CGameRound(), et ayoaron ::graphic ::CGuiSystem ::onClic().

Voici le graphe d'appel pour cette fonction :



7.2.4.18 void ayoaron ::audio ::CSoundEngine ::processEvent (event ::CEngineEvent &) [virtual]

traite les événements

Implémente [ayoaron ::core ::CEngine](#).

7.2.4.19 void ayoaron ::core ::CEngine ::processQueue () [inherited]

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

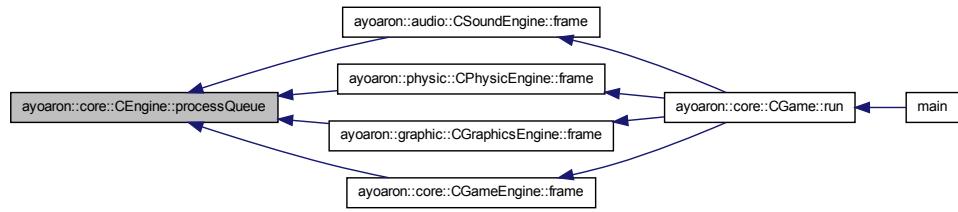
Références ayoaron ::core ::CEngine ::events_queue, ayoaron ::core ::CEngine ::processEvent(), et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par frame(), ayoaron ::physic ::CPhysicEngine ::frame(), ayoaron ::graphic ::CGraphicsEngine ::frame(), et ayoaron ::core ::CGameEngine ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.2.4.20 void ayoaron ::core ::CEngine ::pushEvent (event : :CEngineEvent & e)
 [inherited]

ajout d'un événement

Paramètres

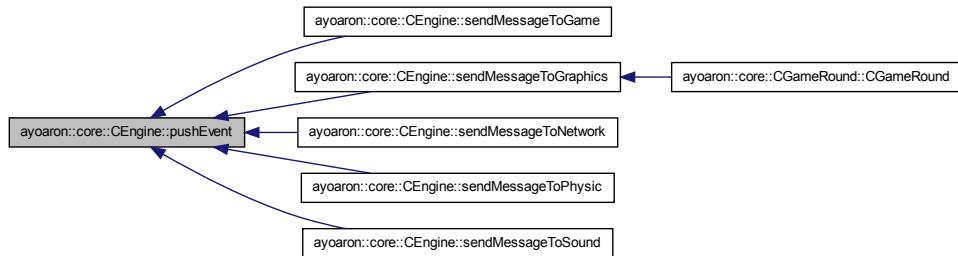
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références ayoaron ::core ::CEngine ::events_queue, et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par ayoaron ::core ::CEngine ::sendMessageToGame(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::core ::CEngine ::sendMessageToPhysic(), et ayoaron ::core ::CEngine ::sendMessageToSound().

Voici le graphe d'appel pour cette fonction :



7.2.4.21 void ayoaron ::core ::CEngine ::sendMessageToGame (event : :CEngineEvent & e) [inherited]

envoie un message au moteur de jeu

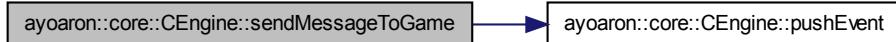
Paramètres

in	e	evenement à envoyer vers le moteur de jeu
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ayoaron ::core ::CEngine ::ge, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.2.4.22 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event : :CEngineEvent & e) [inherited]

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique

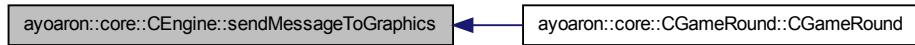
Références ayoaron ::core ::CEngine ::gixe, et ayoaron ::core ::CEngine ::pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.2.4.23 void ayoaron ::core ::CEngine ::sendMessageToNetwork (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ayoaron ::core ::CEngine ::ne, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.2.4.24 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e) [inherited]

envoie un message au moteur physique

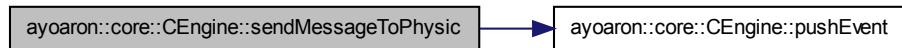
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de physique

Références ayoaron ::core ::CEngine ::phy, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.2.4.25 void ayoaron ::core ::CEngine ::sendMessageToSound (event : :CEngineEvent & e) [inherited]

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de son

Références ayoaron ::core ::CEngine ::pushEvent(), et ayoaron ::core ::CEngine ::se.

Voici le graphe d'appel pour cette fonction :



7.2.4.26 void ayoaron ::audio ::CSoundEngine ::setDopplerProperties (irr ::f64 fac = 1.0, irr ::f64 dist = 1.0)

la vitesse doppler

Références device.

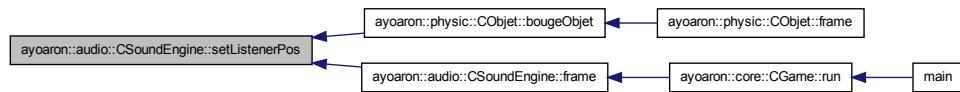
7.2.4.27 void ayoaron ::audio ::CSoundEngine ::setListenerPos (irr ::core ::vector3df pos, irr ::core ::vector3df look = irr ::core ::vector3df(0, 0, 1), irr ::core ::vector3df vel = irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df up = irr ::core ::vector3df(0, 1, 0))

definit la position de l'écoutant

Références device.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), et frame().

Voici le graphe d'appel pour cette fonction :



7.2.5 Documentation des données membres

7.2.5.1 data : :CConfigFile ayoaron ::core ::CEngine ::cfile [protected, inherited]

Fichier de configuration.

Référencé par `getConfigData()`, `ayoaron ::reseau ::CNetworkEngine ::getConfigData()`, et `ayoaron ::graphic ::CGraphicsEngine ::getConfigData()`.

7.2.5.2 irrklang ::ISoundEngine* ayoaron ::audio ::CSoundEngine ::device [protected]

le device de son

Référencé par `CSoundEngine()`, `getDevice()`, `setDopplerProperties()`, `setListenerPos()`, et `~CSoundEngine()`.

7.2.5.3 std ::string ayoaron ::core ::CEngine ::engine_name [protected, inherited]

nom de ce moteur

Référencé par `ayoaron ::core ::CGameEngine ::CGameEngine()`, `ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine()`, et `ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine()`.

7.2.5.4 std ::queue<event ::CEngineEvent> ayoaron ::core ::CEngine ::events_queue [protected, inherited]

liste des événement à traiter

Référencé par `ayoaron ::core ::CEngine ::processQueue()`, et `ayoaron ::core ::CEngine ::pushEvent()`.

7.2.5.5 CGameEngine* ayoaron ::core ::CEngine ::ge [protected, inherited]

pointeur vers le moteur de jeu

Référencé par `ayoaron ::core ::CEngine ::attachGameEngine()`, `ayoaron ::core ::CEngine ::CEngine()`, `ayoaron ::core ::CEngine ::getGameEngine()`, `ayoaron ::core ::CEngine ::setGameEngine()`, et `ayoaron ::core ::CEngine ::updateGameEngine()`.

gine ::sendMessageToGame(), et ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive().

7.2.5.6 `graphic : :CGraphicsEngine* ayoaron : :core : :CEngine : :gffe`
[protected, inherited]

pointeur vers le moteur de rendu graphique

Référencé par ayoaron ::core ::CEngine ::attachGraphicsEngine(), ayoaron ::core ::CEngine :: :CEngine(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), et ayoaron ::core ::CGameEngine ::togglePause().

7.2.5.7 `std ::map<irr ::u32,CSound*> ayoaron : :audio : :CSoundEngine : :l_snd`
[protected]

liste des sons gérés par le moteur

Référencé par addSound(), dropAllSound(), dropSound(), frame(), et getSound().

7.2.5.8 `reseau : :CNetworkEngine* ayoaron : :core : :CEngine : :ne`
[protected, inherited]

pointeur vers le moteur de réseau

Référencé par ayoaron ::core ::CEngine ::attachNetworkEngine(), ayoaron ::core ::CEngine :: :CEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::reseau ::CNetworkEngine ::sendEventTCP(), ayoaron ::reseau ::CNetworkEngine ::sendEventUDP(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive(), et ayoaron ::reseau ::CNetworkEngine ::udpAsyncRead().

7.2.5.9 `CSoundOptions ayoaron : :audio : :CSoundEngine : :options`
[protected]

options sonores

Référencé par CSoundEngine(), et getConfigData().

7.2.5.10 `CGame* ayoaron : :core : :CEngine : :parent` [protected,
inherited]

pointeur vers le parent

Référencé par ayoaron ::graphic ::CGraphicsEngine ::frame(), ayoaron ::core ::CEngine ::getParent(), et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

7.2.5.11 `bool ayoaron : :core : :CEngine : :passive_mode` [protected,
inherited]

true si mode passif activé

Référencé par ayoaron ::core ::CEngine :: :CEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et CSoundEngine().

7.2.5.12 **physic : :CPhysicEngine*** ayoaron ::core ::CEngine ::phy
[protected, inherited]

pointeur vers le moteur de physique

Référencé par ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::CEngine(), et ayoaron ::core ::CEngine ::sendMessageToPhysic().

7.2.5.13 **boost ::mutex ayoaron ::core ::CEngine ::queue_mutex** [protected, inherited]

Mutex de protection de la liste d'événement event_queue.

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.2.5.14 **audio : :CSoundEngine*** ayoaron ::core ::CEngine ::se
[protected, inherited]

pointeur vers le moteur de son

Référencé par ayoaron ::core ::CEngine ::attachSoundEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getSoundEngine(), et ayoaron ::core ::CEngine ::sendMessageToSound().

7.2.5.15 **irr ::scene ::ISceneManager*** ayoaron ::audio ::CSoundEngine ::smgr
[protected]

manager de scène

Référencé par frame().

7.2.5.16 **int ayoaron ::core ::CEngine ::state** [inherited]

l'état actuel

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::physic ::CSecteur ::frame(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::reseau ::CNetworkEngine ::multicastReceive(), ayoaron ::graphic ::CGraphicsEngine ::setState(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.2.5.17 **boost ::mutex ayoaron ::core ::CEngine ::state_mutex** [inherited]

Mutex de protection de l'état.

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

kEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

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

- [CSoundEngine.h](#)
- [CSoundEngine.cpp](#)

7.3 Référence de la classe ayoaron ::audio ::CSoundOptions

classe des options sonore

```
#include <CSoundOptions.h>
```

Fonctions membres publiques

- [CSoundOptions \(\)](#)
constructeur
- [~CSoundOptions \(\)](#)
destructeur

Attributs publics

- irr ::f64 [distfac](#)
doppler distance factor
- irr ::f64 [dolifac](#)
doppler factor
- irr ::f64 [maxdist](#)
max distance
- irr ::f64 [mindist](#)
min distance
- irr ::f64 [rolloff](#)
le rolloffactor
- irr ::f64 [volume](#)
le volume

7.3.1 Description détaillée

classe des options sonore

7.3.2 Documentation des constructeurs et destructeur

7.3.2.1 ayoaron ::audio ::CSoundOptions ::CSoundOptions() [inline]

constructeur

7.3.2.2 ayoaron ::audio ::CSoundOptions ::~CSoundOptions() [inline]

destructeur

7.3.3 Documentation des données membres

7.3.3.1 irr ::f64 ayoaron ::audio ::CSoundOptions ::distfac

doppler distance factor

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

7.3.3.2 irr ::f64 ayoaron ::audio ::CSoundOptions ::dolffac

doppler factor

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

7.3.3.3 irr ::f64 ayoaron ::audio ::CSoundOptions ::maxdist

max distance

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

7.3.3.4 irr ::f64 ayoaron ::audio ::CSoundOptions ::mindist

min distance

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

7.3.3.5 irr ::f64 ayoaron ::audio ::CSoundOptions ::rolloff

le rolloffactor

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

7.3.3.6 irr ::f64 ayoaron ::audio ::CSoundOptions ::volume

le volume

Référencé par ayoaron ::audio ::CSoundEngine ::CSoundEngine(), et ayoaron ::audio ::CSoundEngine ::getConfigData().

La documentation de cette classe a été générée à partir du fichier suivant :

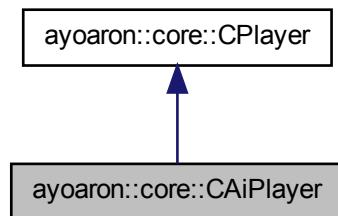
– [CSoundOptions.h](#)

7.4 Référence de la classe ayoaron ::core ::CAiPlayer

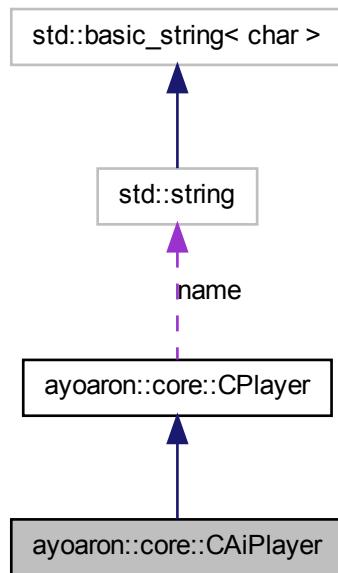
les pnj

```
#include <CAiPlayer.h>
```

Graphe d'héritage de ayoaron ::core ::CAiPlayer :



Graphe de collaboration de ayoaron ::core ::CAiPlayer :



Fonctions membres publiques

- [CAiPlayer \(\)](#)

- Constructeur.*
- [CAiPlayer \(CPlayerOptions *\)](#)
constructeur avec options
 - [std ::string & getName \(\)](#)
renvoie le nom du joueur

7.4.1 Description détaillée

les pnj

7.4.2 Documentation des constructeurs et destructeur

7.4.2.1 ayoaron ::core ::CAiPlayer ::CAiPlayer () [inline]

Constructeur.

7.4.2.2 ayoaron ::core ::CAiPlayer ::CAiPlayer (CPlayerOptions * po)

constructeur avec options

Paramètres

in	po	les option du joueur IA
----	----	-------------------------

change le niveau de l'AI

Références ayoaron ::core ::CRoundOptions ::AI_EASY, ayoaron ::core ::CRoundOptions ::AI_HARD, ayoaron ::core ::CRoundOptions ::AI_MEDIUM, et ayoaron ::core ::CPlayerOptions ::level.

7.4.3 Documentation des fonctions membres

7.4.3.1 std ::string& ayoaron ::core ::CPlayer ::getName () [inline, inherited]

renvoie le nom du joueur

Références ayoaron ::core ::CPlayer ::name.

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

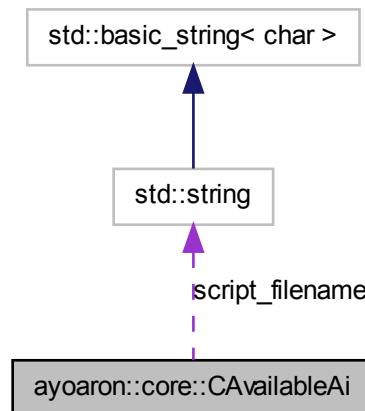
- [CAiPlayer.h](#)
- [CAiPlayer.cpp](#)

7.5 Référence de la classe ayoaron ::core ::CAvailableAi

contient le nom du script

```
#include <CAvailableAi.h>
```

Graphe de collaboration de ayoaron ::core ::CAvailableAi :



Attributs publics

- std ::string [script_filename](#)
nom du script d'AI

7.5.1 Description détaillée

contient le nom du script

à terme cette classe contiendra le traitement de l'AI...

7.5.2 Documentation des données membres

7.5.2.1 std ::string ayoaron ::core ::CAvailableAi ::script_filename

nom du script d'AI

La documentation de cette classe a été générée à partir du fichier suivant :

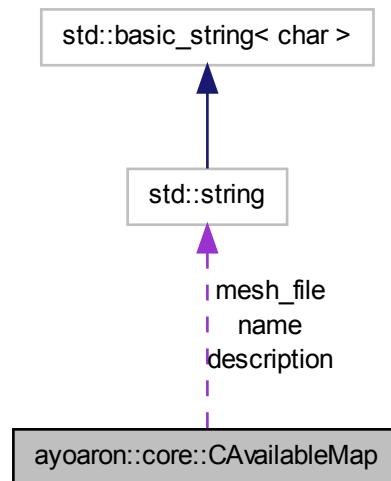
- [CAvailableAi.h](#)

7.6 Référence de la classe ayoaron ::core ::CAvailableMap

La liste des maps.

```
#include <CAvailableMap.h>
```

Graphe de collaboration de ayoaron ::core ::CAvailableMap :



Attributs publics

- `std ::string description`
Description de la map.
- `std ::string mesh_file`
Nom du fichier contenant le maillage.
- `std ::string name`
Nom de la map.

7.6.1 Description détaillée

La liste des maps.

À faire

il faudra enrichir cette classe pour le traitement d'autres resources...

7.6.2 Documentation des données membres

7.6.2.1 `std ::string ayoaron ::core ::CAvailableMap ::description`

Description de la map.

7.6.2.2 std ::string ayoaron ::core ::CAvailableMap ::mesh_file

Nom du fichier contenant le maillage.

7.6.2.3 std ::string ayoaron ::core ::CAvailableMap ::name

Nom de la map.

La documentation de cette classe a été générée à partir du fichier suivant :

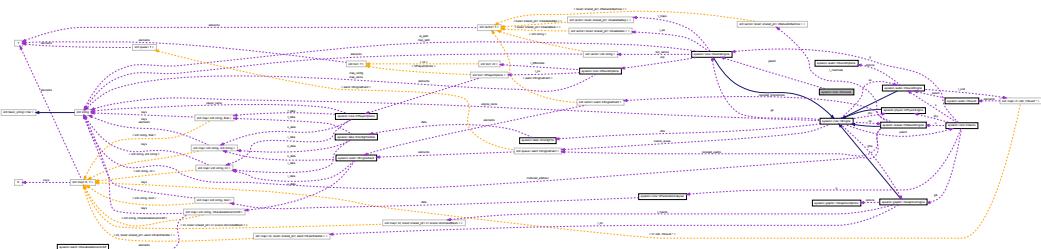
- [CAvailableMap.h](#)

7.7 Référence de la classe ayoaron ::core ::CConsole

classe contenant la [CConsole](#) à faire apparaître en jeu

```
#include <CConsole.h>
```

Graphe de collaboration de ayoaron ::core ::CConsole :



Fonctions membres publiques

- [CConsole \(CGameEngine *\)](#)
constructeur
- [void handleAcceptTCP \(const boost ::system ::error_code &e, boost ::asio ::ip ::tcp ::socket *sock\)](#)
démarre un environnement d'écoute TCP
- [void processCommand \(std ::string\)](#)
exécution de la commande
- [void processCommand \(CEGUI ::String &\)](#)
exécution de la commande
- [void serverThreadTCPReceive \(boost ::asio ::ip ::tcp ::socket *le\)](#)
reçoit une connexion TCP
- [~CConsole \(\)](#)
destructeur

Attributs privés

- [boost ::asio ::io_service io](#)
le service d'entrée/sortie
- [CGameEngine * parent](#)
pointeur vers le parent
- [boost ::asio ::ip ::tcp ::socket * s](#)

- *le socket TCP*
- `boost ::asio ::ip ::tcp ::acceptor * tcp_acceptor`
l'écouteur TCP

7.7.1 Description détaillée

classe contenant la `CConsole` à faire apparaître en jeu

7.7.2 Documentation des constructeurs et destructeur

7.7.2.1 ayoaron ::core ::CConsole ::CConsole (`CGameEngine * g`)

constructeur

Paramètres

<code>in</code>	<code>g</code>	pointeur vers la classe game_engine parente
-----------------	----------------	---

À faire

rendre ssl

Références `handleAcceptTCP()`, `io`, `s`, et `tcp_acceptor`.

Voici le graphe d'appel pour cette fonction :



7.7.2.2 ayoaron ::core ::CConsole ::~CConsole ()

destructeur

7.7.3 Documentation des fonctions membres

7.7.3.1 `void ayoaron ::core ::CConsole ::handleAcceptTCP (const boost ::system ::error_code & e, boost ::asio ::ip ::tcp ::socket * sock)`

démarre un environnement d'écoute TCP

Paramètres

<code>out</code>	<code>e</code>	le code d'erreur généré
<code>in</code>	<code>sock</code>	socket le socket TCP

Références `io`, `s`, `serverThreadTCPReceive()`, et `tcp_acceptor`.

Référencé par `CConsole()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.7.3.2 void ayoaron ::core ::CConsole ::processCommand (std ::string c)

exécution de la commande

Paramètres

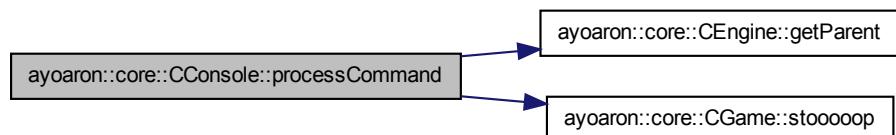
in	c	commande à traiter (format std)
----	---	---------------------------------

À faire

rajouter des commandes

Références ayoaron ::core ::CEngine ::getParent(), parent, et ayoaron ::core ::CGame ::stooooop().

Voici le graphe d'appel pour cette fonction :



7.7.3.3 void ayoaron ::core ::CConsole ::processCommand (CEGUI ::String & s)

exécution de la commande

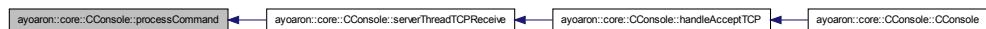
Paramètres

in	s	commande à traiter (format CEGUI)
----	---	-----------------------------------

Références s.

Référencé par serverThreadTCPReceive().

Voici le graphe d'appel pour cette fonction :



7.7.3.4 void ayoaron ::core ::CConsole ::serverThreadTCPReceive (boost ::asio ::ip ::tcp ::socket * *le*)

recoie une connexion TCP

Paramètres

in	<i>le</i>	socket TCP à lire
----	-----------	-------------------

À faire

fonction à reparer

Références processCommand().

Référencé par handleAcceptTCP().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.7.4 Documentation des données membres

7.7.4.1 `boost ::asio ::io_service ayoaron ::core ::CConsole ::io` [private]

le service d'entrée/sortie

Référencé par `CConsole()`, et `handleAcceptTCP()`.

7.7.4.2 `CGameEngine* ayoaron ::core ::CConsole ::parent` [private]

pointeur vers le parent

Référencé par `processCommand()`.

7.7.4.3 `boost ::asio ::ip ::tcp ::socket* ayoaron ::core ::CConsole ::s` [private]

le socket TCP

Référencé par `CConsole()`, `handleAcceptTCP()`, et `processCommand()`.

7.7.4.4 `boost ::asio ::ip ::tcp ::acceptor* ayoaron ::core ::CConsole ::tcp_acceptor` [private]

l'écouteur TCP

Référencé par `CConsole()`, et `handleAcceptTCP()`.

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

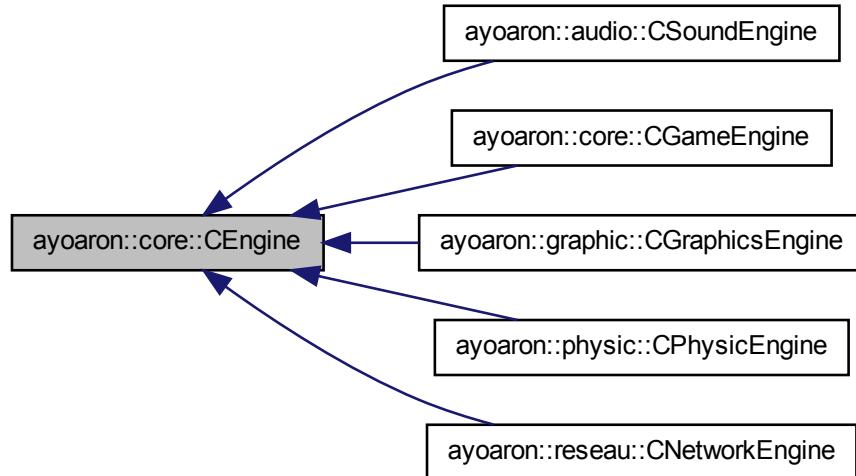
- [CConsole.h](#)
- [CConsole.cpp](#)

7.8 Référence de la classe ayoaron ::core ::CEngine

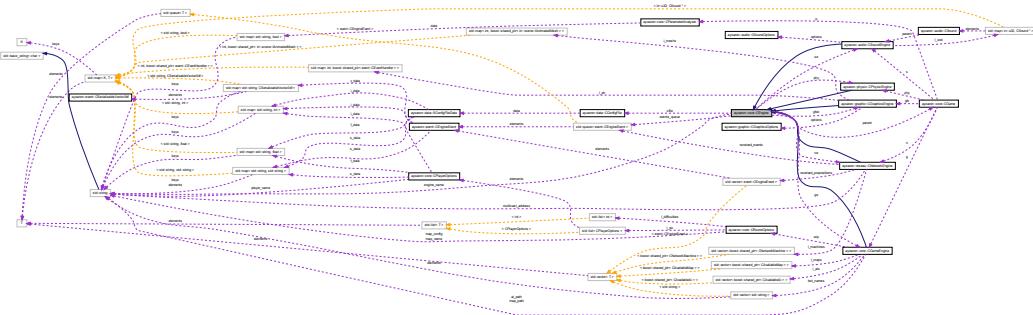
Classe de base des moteurs.

```
#include <CEngine.h>
```

Graphe d'héritage de ayoaron ::core ::CEngine :



Graphe de collaboration de ayoaron ::core ::CEngine :



Types publics

- enum {
 STATE_NOTHING, STATE_QUITTING, STATE_WAITING_PLAYERS, STATE_PLAYING_INTRO,
STATE_LOADING_ROUND, STATE_PLAYING }

Constantes d'état.

Fonctions membres publiques

- void **attachGameEngine** (CGameEngine *e)
lien avec le moteur de jeu
- void **attachGraphicsEngine** (graphic ::CGraphicsEngine *e)
lien avec le moteur de rendu graphique
- void **attachNetworkEngine** (reseau ::CNetworkEngine *e)
lien avec le moteur de réseau
- void **attachPhysicEngine** (physic ::CPhysicEngine *e)
lien avec le moteur de physique
- void **attachSoundEngine** (audio ::CSoundEngine *e)
lien avec le moteur de son
- **CEngine** (CGame *g)
Constructeur.
- virtual void **frame** (double dt)=0
Méthode Frame.
- **CGameEngine** * **getGameEngine** ()
lien vers le moteur de jeu
- **graphic** ::CGraphicsEngine * **getGraphicsEngine** ()
lien vers le moteur de rendu graphique
- **reseau** ::CNetworkEngine * **getNetworkEngine** ()
lien vers le moteur de réseau
- **CGame** * **getParent** ()
lien vers la classe game parente
- **audio** ::CSoundEngine * **getSoundEngine** ()
lien vers le moteur de son
- void **processQueue** ()
vidange de l'event_queue
- void **pushEvent** (event ::CEngineEvent &e)
ajout d'un événement
- void **sendMessageToGame** (event ::CEngineEvent &e)
envoie un message au moteur de jeu
- void **sendMessageToGraphics** (event ::CEngineEvent &e)
envoie un message au moteur de rendu graphique
- void **sendMessageToNetwork** (event ::CEngineEvent &e)
envoie un message au moteur de réseau
- void **sendMessageToPhysic** (event ::CEngineEvent &e)
envoie un message au moteur physique
- void **sendMessageToSound** (event ::CEngineEvent &e)
envoie un message au moteur de son
- virtual ~**CEngine** ()
Destructeur.

Attributs publics

- int **state**
l'état actuel
- boost ::mutex **state_mutex**
Mutex de protection de l'état.

Fonctions membres protégées

- virtual void **getConfigData** ()=0
récupération de la configuration actuelle
- virtual void **processEvent** (event ::CEngineEvent &e)=0
traitement d'un événement

Attributs protégés

- **data** : `CConfigFile cfile`
Fichier de configuration.
- **std** : `string engine_name`
nom de ce moteur
- **std** : `queue<event : CEngineEvent> events_queue`
liste des événement à traiter
- **CGameEngine** * `ge`
pointeur vers le moteur de jeu
- **graphic** : `CGraphicsEngine * gixe
pointeur vers le moteur de rendu graphique`
- **reseau** : `CNetworkEngine * ne
pointeur vers le moteur de réseau`
- **CGame** * `parent`
pointeur vers le parent
- **bool** `passive_mode`
true si mode passif activé
- **physic** : `CPHysicEngine * phy
pointeur vers le moteur de physique`
- **boost** : `mutex queue_mutex`
Mutex de protection de la liste d'événement event_queue.
- **audio** : `CSoundEngine * se
pointeur vers le moteur de son`

7.8.1 Description détaillée

Classe de base des moteurs.

Classe abstraite, non instanciable Les moteurs sont utilisé dans l'objet game

Chaque moteur est connecté aux autres moteurs (jeu, graphique, réseau) Deux possibilité pour envoyer des message à un autre moteur :

- appel direct à travers un pointeur de la fonction du moteur à contacter
- ajout d'un message au `event_queue` du moteur à contacter la `event_queue` est vidée à chaque frame

7.8.2 Documentation des énumérations membres

7.8.2.1 anonymous enum

Constantes d'état.

Valeurs énumérées :

```
STATE NOTHING
STATE QUITTING
STATE WAITING PLAYERS
STATE PLAYING INTRO
STATE LOADING ROUND
STATE PLAYING
```

7.8.3 Documentation des constructeurs et destructeur

7.8.3.1 ayoaron ::core ::CEngine ::CEngine (CGame * g)

Constructeur.

Paramètres

in	g	pointeur vers la classe game parente
----	---	--------------------------------------

Références ge, gixe, ne, passive_mode, phy, et se.

7.8.3.2 ayoaron ::core ::CEngine ::~CEngine () [virtual]

Destructeur.

7.8.4 Documentation des fonctions membres

7.8.4.1 void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e) [inline]

lien avec le moteur de jeu

Paramètres

in	e	pointeur vers le moteur de jeu à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.8.4.2 void ayoaron ::core ::CEngine ::attachGraphicsEngine (CGraphicsEngine * e) [inline]

lien avec le moteur de rendu graphique

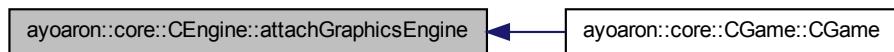
Paramètres

in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique Références gixe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.8.4.3 void ayoaron ::core ::CEngine ::attachNetworkEngine (reseau ::CNetworkEngine * e) [inline]

lien avec le moteur de réseau

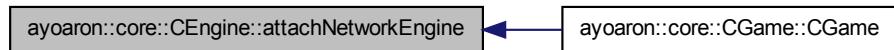
Paramètres

in	e	pointeur vers le moteur de réseau à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau Références ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.8.4.4 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * e) [inline]

lien avec le moteur de physique

Paramètres

in	e	pointeur vers le moteur de physique à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.8.4.5 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.8.4.6 virtual void ayoaron ::core ::CEngine ::frame (double dt) [pure virtual]

Methode Frame.

Cette méthode est utiliser pour mettre à jour le moteur

Implémenté dans [ayoon ::audio ::CSoundEngine](#), [ayoon ::core ::CGameEngine](#), [ayoon ::graphic ::CGraphicsEngine](#), [ayoon ::physic ::CPhysicsEngine](#), et [ayoon ::reseau ::CNetworkEngine](#).

7.8.4.7 virtual void ayoaron ::core ::CEngine ::getConfigData() [protected, pure virtual]

récupération de la configuration actuelle

Implémenté dans [ayoaron ::audio ::CSoundEngine](#), [ayoaron ::core ::CGameEngine](#), [ayoaron ::graphic ::CGraphicsEngine](#), [ayoaron ::physic ::CPhysicEngine](#), et [ayoaron ::reseau ::CNetworkEngine](#).

7.8.4.8 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine() [inline]

lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références ge.

Référencé par [ayoaron ::reseau ::CNetworkMachine ::connect\(\)](#), [ayoaron ::graphic ::CGuiSystem ::handleProcessCommand\(\)](#), [ayoaron ::graphic ::CGuiSystem ::loadGame\(\)](#), [ayoaron ::core ::CGameRound ::makeFromCollect\(\)](#), [ayoaron ::event ::CGameEventHandler ::onEvent\(\)](#), et [ayoaron ::graphic ::CGraphicsEngine ::processEvent\(\)](#).

Voici le graphe d'appel pour cette fonction :

**7.8.4.9 graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine() [inline]**

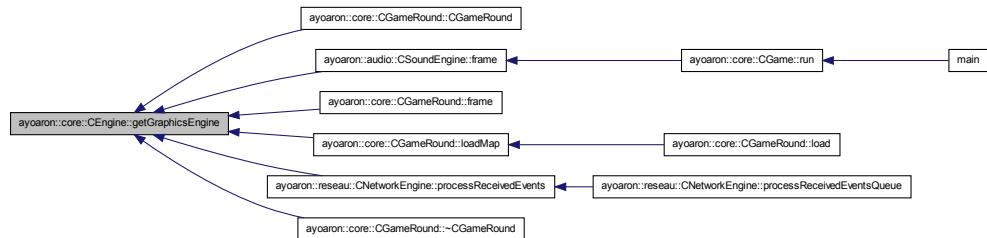
lien vers le moteur de rendu graphique

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de rendu graphique

Références gixe.

Référencé par [ayoaron ::core ::CGameRound ::CGameRound\(\)](#), [ayoaron ::audio ::CSoundEngine ::frame\(\)](#), [ayoaron ::core ::CGameRound ::frame\(\)](#), [ayoaron ::core ::CGameRound ::loadMap\(\)](#), [ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents\(\)](#), et [ayoaron ::core ::CGameRound ::~CGameRound\(\)](#).

Voici le graphe d'appel pour cette fonction :



7.8.4.10 reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::getNetworkEngine () [inline]

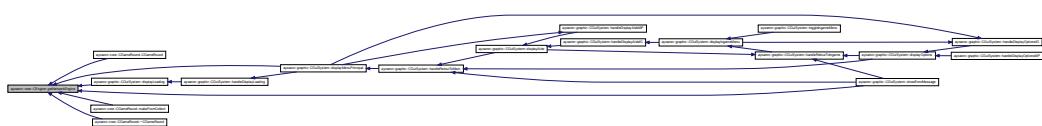
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.8.4.11 CGame* ayoaron ::core ::CEngine ::getParent () [inline]

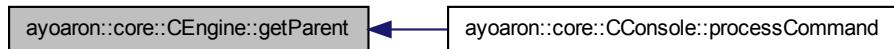
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.8.4.12 `audio : :CSoundEngine* ayoaron ::core ::CEngine ::getSoundEngine()` [inline]

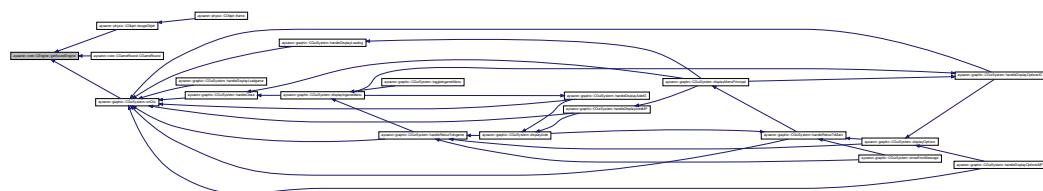
lien vers le moteur de son

Cette méthode est utilisée pour obtenir un pointeur vers le moteur de son

Références se.

Référencé par `ayoaron ::physic ::COBJET ::bougeObjet()`, `ayoaron ::core ::CGameround ::CGameRound()`, et `ayoaron ::graphic ::CGuiSystem ::onClic()`.

Voici le graphe d'appel pour cette fonction :



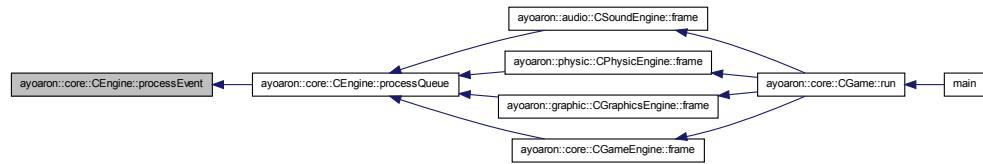
7.8.4.13 `virtual void ayoaron ::core ::CEngine ::processEvent(event : :CEngineEvent &)` [protected, pure virtual]

traitement d'un événement

Implémenté dans `ayoaron ::audio ::CSoundEngine`, `ayoaron ::core ::CGameEngine`, `ayoaron ::graphic ::CGraphicsEngine`, `ayoaron ::physic ::CPhysicEngine`, et `ayoaron ::reseau ::CNetworkEngine`.

Référencé par `processQueue()`.

Voici le graphe d'appel pour cette fonction :



7.8.4.14 void ayoaron ::core ::CEngine ::processQueue ()

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

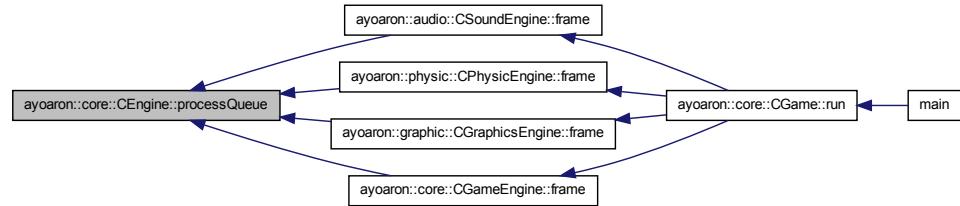
Références events_queue, processEvent(), et queue_mutex.

Référencé par ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::physic ::CPhysicEngine ::frame(), ayoaron ::graphic ::CGraphicsEngine ::frame(), et ayoaron ::core ::CGameEngine ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.8.4.15 void ayoaron ::core ::CEngine ::pushEvent (event ::CEngineEvent & e)

ajout d'un événement

Paramètres

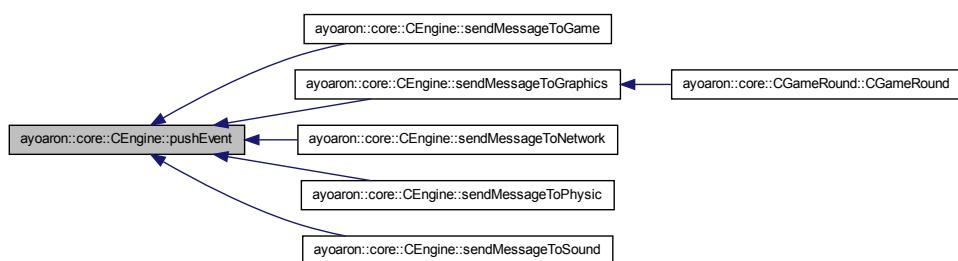
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références events_queue, et queue_mutex.

Référencé par sendMessageToGame(), sendMessageToGraphics(), sendMessageToNetwork(), sendMessageToPhysic(), et sendMessageToSound().

Voici le graphe d'appel pour cette fonction :



7.8.4.16 void ayoaron ::core ::CEngine ::sendMessageToGame (event ::CEngineEvent & e)

envoie un message au moteur de jeu

Paramètres

in	e	événement à envoyer vers le moteur de jeu
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ge, et pushEvent().

Voici le graphe d'appel pour cette fonction :



7.8.4.17 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event ::CEngineEvent & e)

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

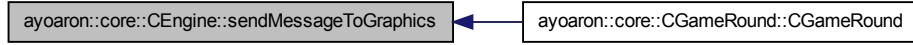
Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique Références gixe, et pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.8.4.18 void ayoaron ::core ::CEngine ::sendMessageToNetwork (event ::CEngineEvent & e)

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ne, et pushEvent().

Voici le graphe d'appel pour cette fonction :



7.8.4.19 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e)

envoie un message au moteur physique

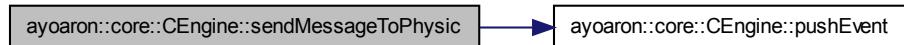
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un évenement vers moteur de physique

Références phy, et pushEvent().

Voici le graphe d'appel pour cette fonction :



7.8.4.20 void ayoaron ::core ::CEngine ::sendMessageToSound (event ::CEngineEvent & e)

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un évenement vers moteur de son

Références pushEvent(), et se.

Voici le graphe d'appel pour cette fonction :



7.8.5 Documentation des données membres

7.8.5.1 `data : :CConfigFile ayoaron ::core ::CEngine ::cfile` [protected]

Fichier de configuration.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::reseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.8.5.2 `std ::string ayoaron ::core ::CEngine ::engine_name` [protected]

nom de ce moteur

Référencé par ayoaron ::core ::CGameEngine ::CGameEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine().

7.8.5.3 `std ::queue<event ::CEngineEvent> ayoaron ::core ::CEngine ::events_queue` [protected]

liste des événement à traiter

Référencé par processQueue(), et pushEvent().

7.8.5.4 `CGameEngine* ayoaron ::core ::CEngine ::ge` [protected]

pointeur vers le moteur de jeu

Référencé par attachGameEngine(), CEngine(), getGameEngine(), sendMessageToGame(), et ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive().

7.8.5.5 `graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::gixe` [protected]

pointeur vers le moteur de rendu graphique

Référencé par attachGraphicsEngine(), CEngine(), ayoaron ::core ::CGameEngine ::frame(), getGraphicsEngine(), sendMessageToGraphics(), et ayoaron ::core ::CGameEngine ::togglePause().

**7.8.5.6 reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::ne
[protected]**

pointeur vers le moteur de réseau

Référencé par attachNetworkEngine(), CEngine(), getNetworkEngine(), ayoaron ::reseau ::CNetworkEngine ::sendEventTCP(), ayoaron ::reseau ::CNetworkEngine ::sendEventUDP(), sendMessageToNetwork(), ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive(), et ayoaron ::reseau ::CNetworkEngine ::udpAsyncRead().

7.8.5.7 CGame* ayoaron ::core ::CEngine ::parent [protected]

pointeur vers le parent

Référencé par ayoaron ::graphic ::CGraphicsEngine ::frame(), getParent(), et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

7.8.5.8 bool ayoaron ::core ::CEngine ::passive_mode [protected]

true si mode passif activé

Référencé par CEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::audio ::CSoundEngine ::CSoundEngine().

**7.8.5.9 physic ::CPhysicEngine* ayoaron ::core ::CEngine ::phy
[protected]**

pointeur vers le moteur de physique

Référencé par attachPhysicEngine(), CEngine(), et sendMessageToPhysic().

7.8.5.10 boost ::mutex ayoaron ::core ::CEngine ::queue_mutex [protected]

Mutex de protection de la liste d'événement event_queue.

Référencé par processQueue(), et pushEvent().

**7.8.5.11 audio ::CSoundEngine* ayoaron ::core ::CEngine ::se
[protected]**

pointeur vers le moteur de son

Référencé par attachSoundEngine(), CEngine(), getSoundEngine(), et sendMessageToSound().

7.8.5.12 int ayoaron ::core ::CEngine ::state

l'état actuel

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::physic ::CSecteur ::frame(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::reseau ::CNetworkEngine ::mul-

ticastReceive(), ayoaron ::graphic ::CGraphicsEngine ::setState(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.8.5.13 boost ::mutex ayoaron ::core ::CEngine ::state_mutex

Mutex de protection de l'état.

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

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

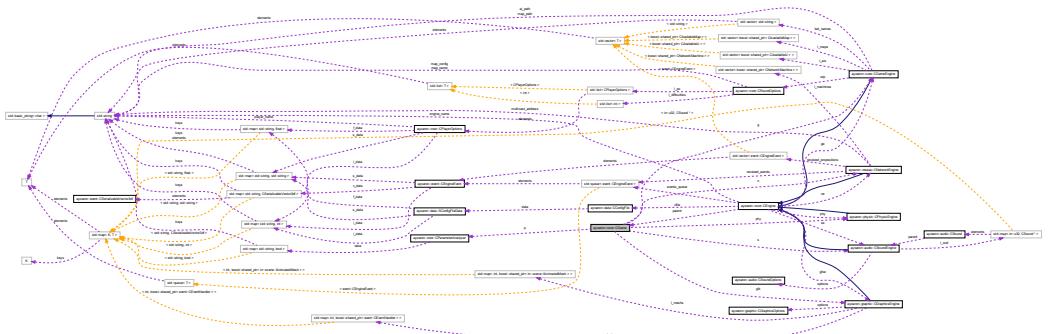
- [CEngine.h](#)
- [CEngine.cpp](#)

7.9 Référence de la classe ayoaron ::core ::CGame

classe de base du jeu

```
#include <CGame.h>
```

Graphe de collaboration de ayoaron ::core ::CGame :



Fonctions membres publiques

- [CGame \(CParameterAnalyser\)](#)
Constructeur.
- double [getDt \(\)](#)
cette méthode permet de récupérer le deltatemps
- int [init \(\)](#)
Méthode d'initialisation du moteur de jeu.
- void [run \(\)](#)
Méthode de boucle principale.
- void [stooooop \(\)](#)
Méthode d'arrêt du moteur de jeu.
- [~CGame \(\)](#)
Destructeur.

Attributs privés

- double **DeltaTemps**
le temps entre deux frame
- **CGameEngine * g**
le moteur de jeu
- **graphic : CGraphicsEngine * gfx**
le moteur graphique
- **reseau : CNetworkEngine * n**
le moteur de réseau
- **CParameterAnalyser p**
liste des paramètres d'entrée
- **physic : CPhysicEngine * phy**
le moteur de physique
- **audio : CSoundEngine * s**
le moteur de son
- bool **still_running**
true si le jeu tourne
- boost : :mutex **still_running_mutex**
Mutex permettant de savoir si tout fonctionne.
- irr : :u32 **then**
Constantes de temps.

7.9.1 Description détaillée

classe de base du jeu

la classe de base supportant toute l'architecture du jeu

7.9.2 Documentation des constructeurs et destructeur

7.9.2.1 ayoaron ::core ::CGame (CParameterAnalyser pa)

Constructeur.

Paramètres

in	pa	Les paramètres de lancement
----	----	-----------------------------

Crée le jeu en initialisant les moteurs

Paramètres

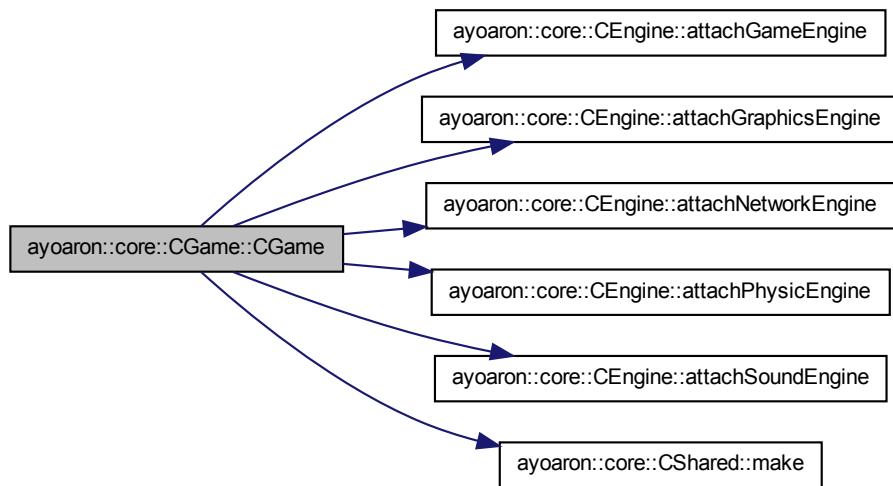
in	pa	Les paramètres de lancement
----	----	-----------------------------

initialise le moteur réseau initialise le moteur graphique initialise le moteur de jeu initialise le moteur de son

attache les moteurs entre eux

Références ayoaron ::core ::CEngine ::attachGameEngine(), ayoaron ::core ::CEngine ::attachGraphicsEngine(), ayoaron ::core ::CEngine ::attachNetworkEngine(), ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::attachSoundEngine(), ayoaron ::core ::CParameterAnalyser ::data, g, gfx, ayoaron ::core ::CShared ::make(), n, p, phy, s, et still_running.

Voici le graphe d'appel pour cette fonction :



7.9.2.2 ayoaron ::core ::CGame ::~CGame ()

Destructeur.

Destructeur de la classe game

Détruit les moteurs

Références g, gfx, n, phy, s, et ayoaron ::core ::CShared ::unmake().

Voici le graphe d'appel pour cette fonction :



7.9.3 Documentation des fonctions membres

7.9.3.1 double ayoaron ::core ::CGame ::getDt() [inline]

cette méthode permet de récupérer le deltatemps

Références DeltaTemps.

7.9.3.2 int ayoaron ::core ::CGame ::init()

Méthode d'initialisation du moteur de jeu.

Renvoie

code d'erreur =0 si pas d'erreur

7.9.3.3 void ayoaron ::core ::CGame ::run()

Méthode de boucle principale.

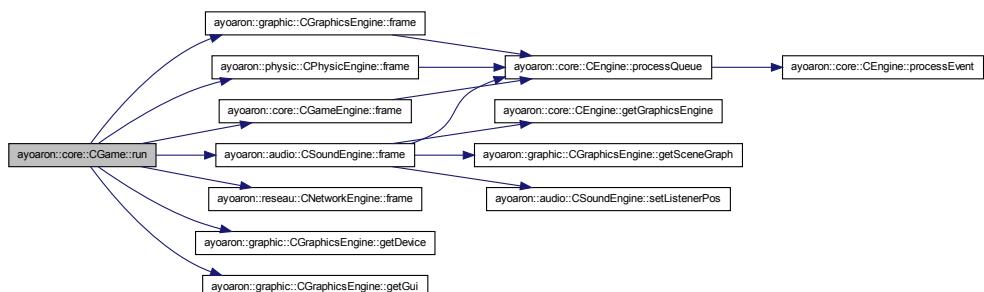
Cette méthode contient la boucle principale du code

effectue les lancement de base contient la boucle principale appellant chaque moteur

Références ayoaron ::core ::CParameterAnalyser ::data, DeltaTemps, ayoaron ::graphic ::CGraphicsEngine ::frame(), ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::physic ::CPhysicEngine ::frame(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::reseau ::CNetworkEngine ::frame(), g, ayoaron ::graphic ::CGraphicsEngine ::getDevice(), ayoaron ::graphic ::CGraphicsEngine ::getGui(), gfx, n, p, phy, s, still_running, still_running_mutex, et then.

Référencé par main().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.9.3.4 void ayoaron ::core ::CGame ::stooooop ()

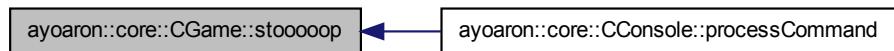
Méthode d'arrête du moteur de jeu.

Arrêt de tous les threads

Références still_running, et still_running_mutex.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.9.4 Documentation des données membres

7.9.4.1 double ayoaron ::core ::CGame ::DeltaTemps [private]

le temps entre deux frame

Référencé par getDt(), et run().

7.9.4.2 CGameEngine* ayoaron ::core ::CGame ::g [private]

le moteur de jeu

Référencé par CGame(), run(), et ~CGame().

7.9.4.3 graphic : CGraphicsEngine* ayoaron ::core ::CGame ::gfx [private]

le moteur graphique

Référencé par CGame(), run(), et ~CGame().

7.9.4.4 reseau : :CNetworkEngine* ayoaron ::core ::CGame ::n [private]

le moteur de réseau

Référencé par CGame(), run(), et ~CGame().

7.9.4.5 CParameterAnalyser ayoaron ::core ::CGame ::p [private]

liste des paramètres d'entrée

Référencé par CGame(), et run().

7.9.4.6 physic : :CPhysicEngine* ayoaron ::core ::CGame ::phy [private]

le moteur de physique

Référencé par CGame(), run(), et ~CGame().

7.9.4.7 audio : :CSoundEngine* ayoaron ::core ::CGame ::s [private]

le moteur de son

Référencé par CGame(), run(), et ~CGame().

7.9.4.8 bool ayoaron ::core ::CGame ::still_running [private]

true si le jeu tourne

Référencé par CGame(), run(), et stooooop().

7.9.4.9 boost ::mutex ayoaron ::core ::CGame ::still_running_mutex [private]

Mutex permettant de savoir si tout fonctionne.

Référencé par run(), et stooooop().

7.9.4.10 irr ::u32 ayoaron ::core ::CGame ::then [private]

Constantes de temps.

Référencé par run().

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

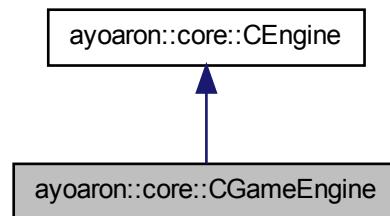
- [CGame.h](#)
- [CGame.cpp](#)

7.10 Référence de la classe ayoaron ::core ::CGameEngine

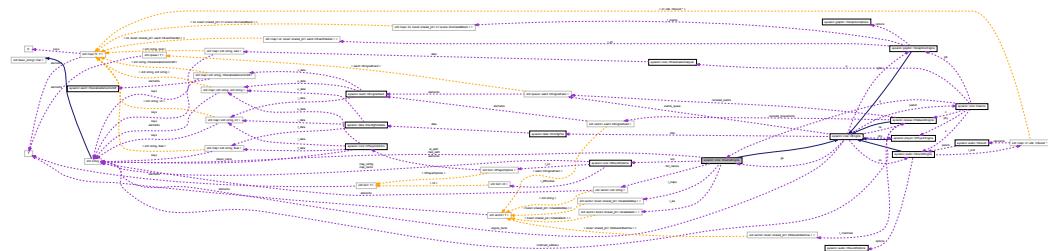
Le moteur de jeu.

```
#include <CGameEngine.h>
```

Graphe d'héritage de ayoaron ::core ::CGameEngine :



Graphe de collaboration de ayoaron ::core ::CGameEngine :



Types publics

- enum {
 STATE_NOTHING, STATE_QUITTING, STATE_WAITING_PLAYERS, STATE_PLAYING_-INTRO,
STATE_LOADING_ROUND, STATE_PLAYING }
- Constantes d'état.*

Fonctions membres publiques

- void **attachGameEngine** (CGameEngine *e)
lien avec le moteur de jeu
- void **attachGraphicsEngine** (graphic ::CGraphicsEngine *e)
lien avec le moteur de rendu graphique
- void **attachNetworkEngine** (reseau ::CNetworkEngine *e)
lien avec le moteur de réseau
- void **attachPhysicEngine** (physic ::CPhysicEngine *e)
lien avec le moteur de physique
- void **attachSoundEngine** (audio ::CSoundEngine *e)
lien avec le moteur de son

- `CGameEngine (CGame *)`
Constructeur.
- `void frame (double DeltaTemps)`
Méthode Frame.
- `boost ::shared_ptr< CConsole > getConsole ()`
accès à la console
- `CGameEngine * getGameEngine ()`
lien vers le moteur de jeu
- `graphic : CGraphicsEngine * getGraphicsEngine ()`
lien vers le moteur de rendu graphique
- `std ::string getMapPath ()`
renvoie le chemin vers la map
- `reseau : CNetworkEngine * getNetworkEngine ()`
lien vers le moteur de réseau
- `CGame * getParent ()`
lien vers la classe game parente
- `audio : CSoundEngine * getSoundEngine ()`
lien vers le moteur de son
- `bool isPaused ()`
dit si on en en pause
- `std ::string makeAiName ()`
générateur de nom pour l'AI
- `virtual void processEvent (event : CEngineEvent &)`
traitement d'un événement
- `void processQueue ()`
vidange de l'event_queue
- `void pushEvent (event : CEngineEvent &e)`
ajout d'un événement
- `void sendMessageToGame (event : CEngineEvent &e)`
envoie un message au moteur de jeu
- `void sendMessageToGraphics (event : CEngineEvent &e)`
envoie un message au moteur de rendu graphique
- `void sendMessageToNetwork (event : CEngineEvent &e)`
envoie un message au moteur de réseau
- `void sendMessageToPhysic (event : CEngineEvent &e)`
envoie un message au moteur physique
- `void sendMessageToSound (event : CEngineEvent &e)`
envoie un message au moteur de son
- `bool togglePause ()`
mets et sort le jeu du mode pause
- `virtual ~CGameEngine ()`
Destructeur.

Attributs publics

- `boost ::shared_ptr< CGameRound > gr`
pointeur vers le niveau actuel
- `int state`
l'état actuel
- `boost ::mutex state_mutex`
Mutex de protection de l'état.
- `CRoundOptions wip`

Fonctions membres protégées

- `void collectAvailableAis ()`
récupération de la liste des script AI disponibles

- void **collectAvailableMaps** ()

récupération de la liste des cartes disponibles
- void **getConfigData** ()

récupération de la configuration actuelle

Attributs protégés

- std ::string **ai_path**

chemin vers le répertoire des script AI
- std ::vector< std ::string > **bot_names**

liste des bot AI
- **data** ::CConfigFile **cfile**

Fichier de configuration.
- boost ::shared_ptr< CConsole > **cons**

pointeur vers la console
- std ::string **engine_name**

nom de ce moteur
- std ::queue< event ::CEngineEvent > **events_queue**

liste des événement à traiter
- long int **framect**
- CGameEngine * **ge**

pointeur vers le moteur de jeu
- CGraphicsEngine * **gixe**

pointeur vers le moteur de rendu graphique
- std ::vector< boost ::shared_ptr< CAvailableAi > > **l_ais**

liste des script AI disponibles
- std ::vector< boost ::shared_ptr< CAvailableMap > > **l_maps**

liste des cartes disponibles
- std ::string **map_path**

chemin vers le répertoire des maps
- CNetworkEngine * **ne**

pointeur vers le moteur de réseau
- CGame * **parent**

pointeur vers le parent
- bool **passive_mode**

true si mode passif activé
- bool **pause**

définit si le jeu est en pause
- CPhysicEngine * **phy**

pointeur vers le moteur de physique
- boost ::mutex **queue_mutex**

Mutex de protection de la liste d'événement event_queue.
- CSoundEngine * **se**

pointeur vers le moteur de son

7.10.1 Description détaillée

Le moteur de jeu.

7.10.2 Documentation des énumérations membres

7.10.2.1 anonymous enum [inherited]

Constantes d'état.

Valeurs énumérées :

STATE NOTHING
STATE QUITTING
STATE_WAITING_PLAYERS
STATE_PLAYING_INTRO
STATE_LOADING_ROUND
STATE_PLAYING

7.10.3 Documentation des constructeurs et destructeur

7.10.3.1 ayoaron ::core ::CGameEngine (CGame * g)

Constructeur.

Paramètres

in	g	pointeur vers la classe game parente
----	---	--------------------------------------

Références ai_path, cons, ayoaron ::core ::CEngine ::engine_name, framect, map_path, et pause.

7.10.3.2 ayoaron ::core ::CGameEngine ::~CGameEngine () [virtual]

Destructeur.

7.10.4 Documentation des fonctions membres

7.10.4.1 void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e) [inline, inherited]

lien avec le moteur de jeu

Paramètres

in	e	pointeur vers le moteur de jeu à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.10.4.2 void ayoaron ::core ::CEngine ::attachGraphicsEngine (graphic : :CGraphicsEngine * e) [inline, inherited]

lien avec le moteur de rendu graphique

Paramètres

in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.10.4.3 void ayoaron ::core ::CEngine ::attachNetworkEngine (reseau ::CNetworkEngine * e) [inline, inherited]

lien avec le moteur de réseau

Paramètres

in	e	pointeur vers le moteur de réseau à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.10.4.4 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * e) [inline, inherited]

lien avec le moteur de physique

Paramètres

in	e	pointeur vers le moteur de physique à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références ayoaron ::core ::CEngine ::phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.10.4.5 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline, inherited]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.10.4.6 void ayoaron ::core ::CGameEngine ::collectAvailableAis() [protected]

récupération de la liste des script AI disponibles

Recherche de tous les script AI disponibles Le rassemblement n'est fait qu'au lancement du jeu

Voir également

[CGameEngine ::collect_available_maps](#)

Références ai_path, et l_ais.

7.10.4.7 void ayoaron ::core ::CGameEngine ::collectAvailableMaps() [protected]

récupération de la liste des cartes disponibles

Recherche toute les carte disponibles Le rassemblement n'est fait qu'au lancement du jeu

Références l_maps, et map_path.

7.10.4.8 void ayoaron ::core ::CGameEngine ::frame(double DeltaTemps) [virtual]

Methode Frame.

Cette méthode est utiliser pour mettre à jour le moteur de jeu

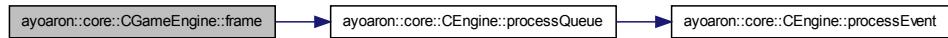
Appelé à chaque frame

Implémenté [ayoaron ::core ::CEngine](#).

Références framect, ayoaron ::core ::CEngine ::gixe, gr, ayoaron ::graphic ::CGraphicsEngine ::ON_GAME, pause, ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::state.

Référencé par ayoaron ::core ::CGame ::run().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.10.4.9 void ayoaron ::core ::CGameEngine ::getConfigData () [protected, virtual]

récupération de la configuration actuelle

Implémente [ayoaron ::core ::CEngine](#).

Références `bot_names`.

7.10.4.10 boost ::shared_ptr<CCConsole> ayoaron ::core ::CGameEngine ::getConsole () [inline]

acces à la console

Renvoie

pointeur vers la console

Références `cons`.

7.10.4.11 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine () [inline, inherited]

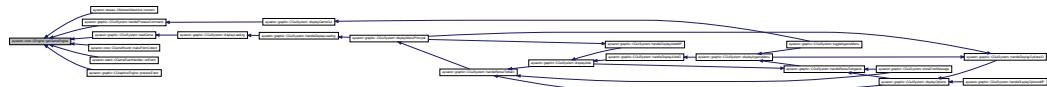
lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références `ayoaron ::core ::CEngine ::ge`.

Référencé par `ayoaron ::reseau ::CNetworkMachine ::connect()`, `ayoaron ::graphic ::CGuiSystem ::handleProcessCommand()`, `ayoaron ::graphic ::CGuiSystem ::loadGame()`, `ayoaron ::core ::CGameRound ::makeFromCollect()`, `ayoaron ::event ::CGameEventHandler ::onEvent()`, et `ayoaron ::graphic ::CGraphicsEngine ::processEvent()`.

Voici le graphe d'appel pour cette fonction :



7.10.4.12 `graphic : :CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine () [inline, inherited]`

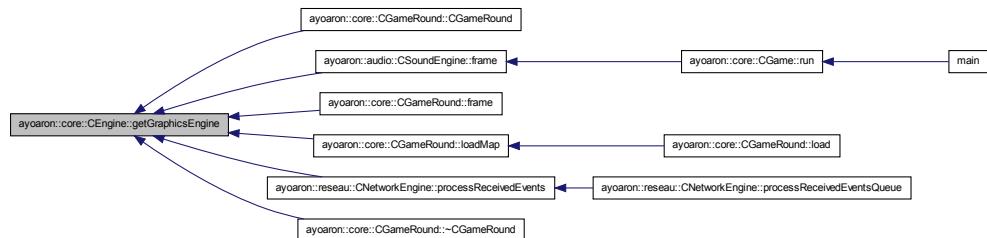
lien vers le moteur de rendu graphique

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::core ::CGameRound ::frame(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.10.4.13 `std ::string ayoaron ::core ::CGameEngine ::getMapPath () [inline]`

renvoie le chemin vers la map

Références map_path.

Référencé par ayoaron ::core ::CGameRound ::makeFromCollect().

Voici le graphe d'appel pour cette fonction :



7.10.4.14 reseau : :CNetworkEngine* ayoaron ::core ::CEngine ::getNetworkEngine()
[inline, inherited]

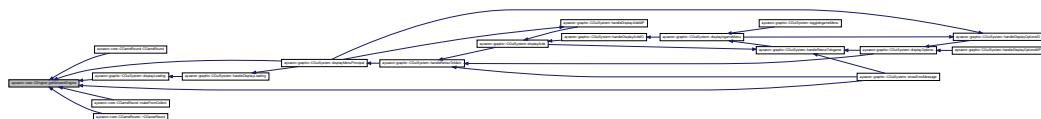
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.10.4.15 CGame* ayoaron ::core ::CEngine ::getParent() [inline, inherited]

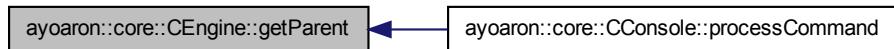
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.10.4.16 audio ::CSoundEngine* ayoaron ::core ::CEngine ::getSoundEngine() [inline, inherited]

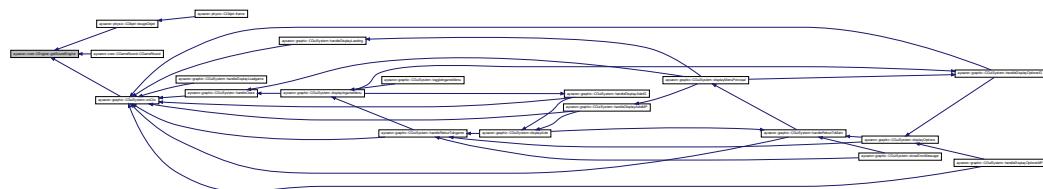
lien vers le moteur de son

Cette méthode est utilisée pour obtenir un pointeur vers le moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::core ::CGameRound ::CGameRound(), et ayoaron ::graphic ::CGuiSystem ::onClic().

Voici le graphe d'appel pour cette fonction :



7.10.4.17 bool ayoaron ::core ::CGameEngine ::isPaused() [inline]

dit si on est en pause

Références pause.

7.10.4.18 std ::string ayoaron ::core ::CGameEngine ::makeAiName()

générateur de nom pour l'AI

Renvoie

un nom pour le joueur AI

revoie un nom au hasard

Références bot_names.

Référencé par ayoaron ::core ::CGameRound ::makeFromCollect().

Voici le graphe d'appel pour cette fonction :



7.10.4.19 `void ayoaron ::core ::CGameEngine ::processEvent (event ::CEngineEvent & e) [virtual]`

traitement d'un événement

Fonction principale, traite un événement reçu

Implémente [ayoaron ::core ::CEngine](#).

7.10.4.20 `void ayoaron ::core ::CEngine ::processQueue () [inherited]`

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

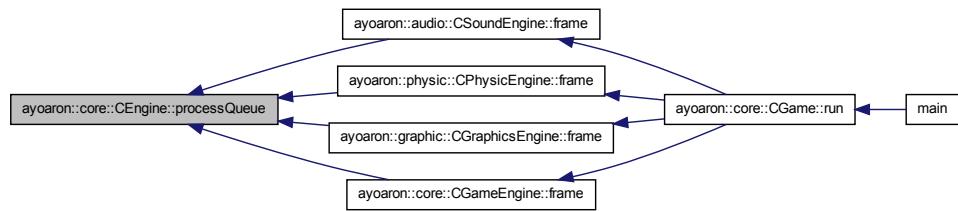
Références `ayoaron ::core ::CEngine ::events_queue`, `ayoaron ::core ::CEngine ::processEvent()`, et `ayoaron ::core ::CEngine ::queue_mutex`.

Référencé par `ayoaron ::audio ::CSoundEngine ::frame()`, `ayoaron ::physic ::CPhysicsEngine ::frame()`, `ayoaron ::graphic ::CGraphicsEngine ::frame()`, et `frame()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.10.4.21 void ayoaron ::core ::CEngine ::pushEvent (event ::CEngineEvent & e)
 [inherited]

ajout d'un événement

Paramètres

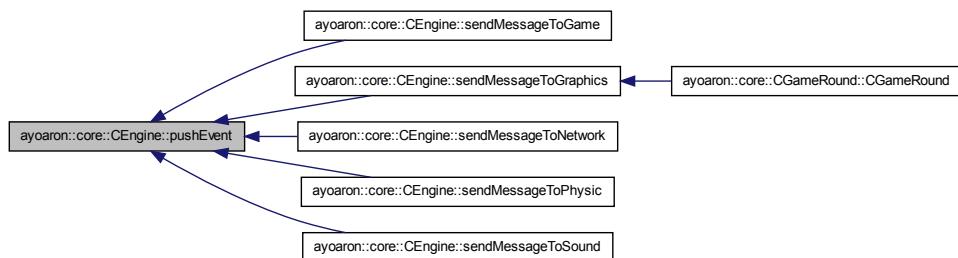
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références ayoaron ::core ::CEngine ::events_queue, et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par ayoaron ::core ::CEngine ::sendMessageToGame(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::core ::CEngine ::sendMessageToPhysic(), et ayoaron ::core ::CEngine ::sendMessageToSound().

Voici le graphe d'appel pour cette fonction :



7.10.4.22 void ayoaron ::core ::CEngine ::sendMessageToGame (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de jeu

Paramètres

in	e	evenement à envoyer vers le moteur de jeu
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ayoaron ::core ::CEngine ::ge, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.10.4.23 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe, et ayoaron ::core ::CEngine ::pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.10.4.24 void ayoaron ::core ::CEngine ::sendMessageToNetwork (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ayoaron ::core ::CEngine ::ne, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.10.4.25 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e) [inherited]

envoie un message au moteur physique

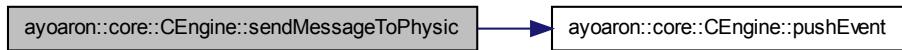
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de physique

Références ayoaron ::core ::CEngine ::phy, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.10.4.26 void ayoaron ::core ::CEngine ::sendMessageToSound (event : :CEngineEvent & e) [inherited]

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de son

Références ayoaron ::core ::CEngine ::pushEvent(), et ayoaron ::core ::CEngine ::se.

Voici le graphe d'appel pour cette fonction :

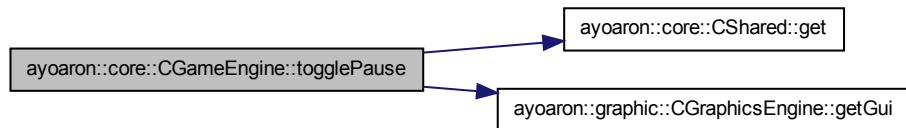


7.10.4.27 bool ayoaron ::core ::CGameEngine ::togglePause ()

mets et sort le jeu du mode pause

Références ayoaron ::core ::CShared ::get(), ayoaron ::graphic ::CGraphicsEngine ::getGui(), ayoaron ::core ::CEngine ::gixe, et pause.

Voici le graphe d'appel pour cette fonction :



7.10.5 Documentation des données membres

7.10.5.1 `std ::string ayoaron ::core ::CGameEngine ::ai_path` [protected]

chemin vers le répertoire des script AI

Référencé par CGameEngine(), et collectAvailableAis().

7.10.5.2 `std ::vector<std ::string> ayoaron ::core ::CGameEngine ::bot_names` [protected]

liste des bot AI

Référencé par getConfigData(), et makeAiName().

7.10.5.3 `data ::CConfigFile ayoaron ::core ::CEngine ::cfile` [protected, inherited]

Fichier de configuration.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::reseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.10.5.4 `boost ::shared_ptr<CCConsole> ayoaron ::core ::CGameEngine ::cons` [protected]

pointeur vers la console

Référencé par CGameEngine(), et getConsole().

7.10.5.5 `std ::string ayoaron ::core ::CEngine ::engine_name` [protected, inherited]

nom de ce moteur

Référencé par CGameEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine().

7.10.5.6 `std::queue<event ::CEngineEvent> ayoaron ::core ::CGameEngine ::events_ - queue` [protected, inherited]

liste des événement à traiter

Référencé par ayoaron ::core ::CGameEngine ::processQueue(), et ayoaron ::core ::CGameEngine ::pushEvent().

7.10.5.7 `long int ayoaron ::core ::CGameEngine ::framect` [protected]

Référencé par CGameEngine(), et frame().

7.10.5.8 `CGameEngine* ayoaron ::core ::CGameEngine ::ge` [protected, inherited]

pointeur vers le moteur de jeu

Référencé par ayoaron ::core ::CGameEngine ::attachGameEngine(), ayoaron ::core ::CGameEngine ::CGameEngine(), ayoaron ::core ::CGameEngine ::getGameEngine(), ayoaron ::core ::CGameEngine ::sendMessageToGame(), et ayoaron ::core ::CNetworkEngine ::serverThreadTCPReceive().

7.10.5.9 `graphic ::CGraphicsEngine* ayoaron ::core ::CGameEngine ::gixe` [protected, inherited]

pointeur vers le moteur de rendu graphique

Référencé par ayoaron ::core ::CGameEngine ::attachGraphicsEngine(), ayoaron ::core ::CGameEngine ::CGameEngine(), frame(), ayoaron ::core ::CGameEngine ::getGraphicsEngine(), ayoaron ::core ::CGameEngine ::sendMessageToGraphics(), et togglePause().

7.10.5.10 `boost ::shared_ptr<CGameRound> ayoaron ::core ::CGameEngine ::gr`

pointeur vers le niveau actuel

Référencé par frame().

7.10.5.11 `std ::vector<boost ::shared_ptr<CAvailableAi> > ayoaron ::core ::CGameEngine ::l_ais` [protected]

liste des script AI disponibles

Référencé par collectAvailableAis().

7.10.5.12 `std ::vector<boost ::shared_ptr<CAvailableMap> > ayoaron ::core ::CGameEngine ::l_maps` [protected]

liste des cartes disponibles

Référencé par collectAvailableMaps().

7.10.5.13 `std ::string ayoaron ::core ::CGameEngine ::map_path` [protected]

chemin vers le répertoire des maps

Référencé par CGameEngine(), collectAvailableMaps(), et getMapPath().

7.10.5.14 reseau : :CNetworkEngine* ayoaron : :core : :CEngine : :ne [protected, inherited]

pointeur vers le moteur de réseau

Référencé par ayoaron ::core ::CEngine ::attachNetworkEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::reseau ::CNetworkEngine ::sendEventTCP(), ayoaron ::reseau ::CNetworkEngine ::sendEventUDP(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive(), et ayoaron ::reseau ::CNetworkEngine ::udpAsyncRead().

7.10.5.15 CGame* ayoaron : :core : :CEngine : :parent [protected, inherited]

pointeur vers le parent

Référencé par ayoaron ::graphic ::CGraphicsEngine ::frame(), ayoaron ::core ::CEngine ::getParent(), et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

7.10.5.16 bool ayoaron : :core : :CEngine : :passive_mode [protected, inherited]

true si mode passif activé

Référencé par ayoaron ::core ::CEngine ::CEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::audio ::CSoundEngine ::CSoundEngine().

7.10.5.17 bool ayoaron : :core : :CGameEngine : :pause [protected]

définit si le jeu est en pause

Référencé par CGameEngine(), frame(), isPaused(), et togglePause().

7.10.5.18 physic : :CPhysicEngine* ayoaron : :core : :CEngine : :phy [protected, inherited]

pointeur vers le moteur de physique

Référencé par ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::CEngine(), et ayoaron ::core ::CEngine ::sendMessageToPhysic().

7.10.5.19 boost ::mutex ayoaron : :core : :CEngine : :queue_mutex [protected, inherited]

Mutex de protection de la liste d'événement event_queue.

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.10.5.20 **audio** : **CSoundEngine*** ayoaron ::core ::CEngine ::se
[protected, inherited]

pointeur vers le moteur de son

Référencé par ayoaron ::core ::CEngine ::attachSoundEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getSoundEngine(), et ayoaron ::core ::CEngine ::sendMessageToSound().

7.10.5.21 **int ayoaron ::core ::CEngine ::state** [inherited]

l'état actuel

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::physic ::CSecteur ::frame(), frame(), ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::reseau ::CNetworkEngine ::multicastReceive(), ayoaron ::graphic ::CGraphicsEngine ::setState(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.10.5.22 **boost ::mutex ayoaron ::core ::CEngine ::state_mutex** [inherited]

Mutex de protection de l'état.

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.10.5.23 **CRoundOptions ayoaron ::core ::CGameEngine ::wip**

Référencé par ayoaron ::core ::CGameRound ::makeFromCollect().

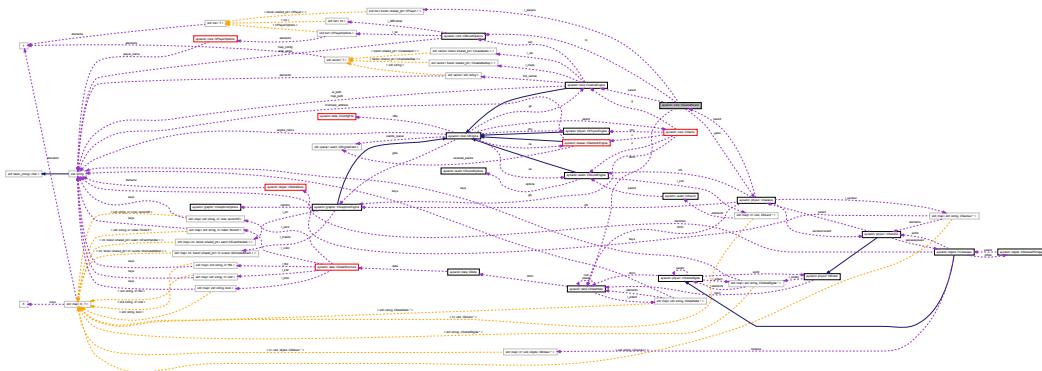
La documentation de cette classe a été générée à partir des fichiers suivants :

- [CGameEngine.h](#)
- [CGameEngine.cpp](#)

7.11 Référence de la classe ayoaron ::core ::CGameRound

```
#include <CGameRound.h>
```

Graphe de collaboration de ayoaron ::core ::CGameRound :



Types publics

- enum { `ON NOTHING, ON_CREATION, ON_PLAYING, ON_FINISH` }
Liste des états de game_round.

Fonctions membres publiques

- `CGameRound (CGameEngine *)`
constructeur
- void `frame (double dt)`
fonction de mise à jour
- bool `isFull ()`
la partie est-elle pleine
- void `load ()`
chargement du niveau de jeu
- void `makeFromCollect ()`
- `~CGameRound ()`
destructeur

Attributs publics

- int `state`
état de la partie.

Fonctions membres privées

- void `loadMap (const std ::string &fileconf)`
chargement de map
- void `loadPlayers (std ::list< CPlayerOptions > &)`
chargement des joueurs

Attributs privés

- `data ::CDataNode * donn`

- structure de donnée*
- **physic** : `:CGalaxie * galax`
pointeur vers la galaxie
 - `std ::list< boost ::shared_ptr< CPlayer > > l_players`
liste des joueurs
 - **CGameEngine * parent**
pointeur vers le moteur de jeu parent
 - **CRoundOptions ro**
options de partie

7.11.1 Documentation des énumérations membres

7.11.1.1 anonymous enum

Liste des états de `game_round`.

Valeurs énumérées :

ON_NOTHING
ON_CREATION
ON_PLAYING
ON_FINISH

7.11.2 Documentation des constructeurs et destructeur

7.11.2.1 ayoaron ::core ::CGameRound (`CGameEngine * ge`)

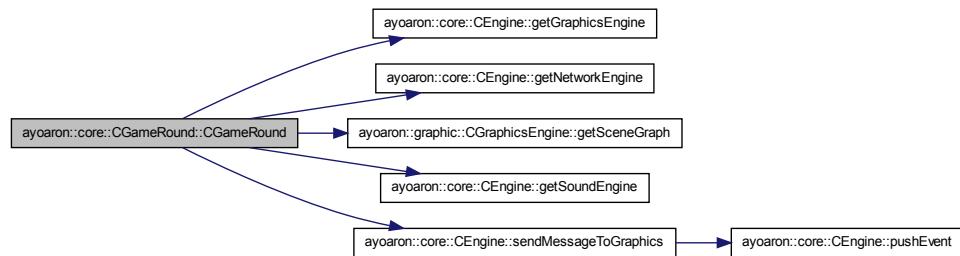
constructeur

Paramètres

<code>ge</code>	le moteur de jeu parent
-----------------	-------------------------

Références donn, galax, ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph(), ayoaron ::core ::CEngine ::getSoundEngine(), ayoaron ::event ::CEngineEvent ::LOAD, ayoaron ::reseau ::CNetworkEngine ::ni, `ON_CREATION`, parent, ayoaron ::core ::CEngine ::sendMessageToGraphics(), state, et ayoaron ::event ::CEngineEvent ::type.

Voici le graphe d'appel pour cette fonction :

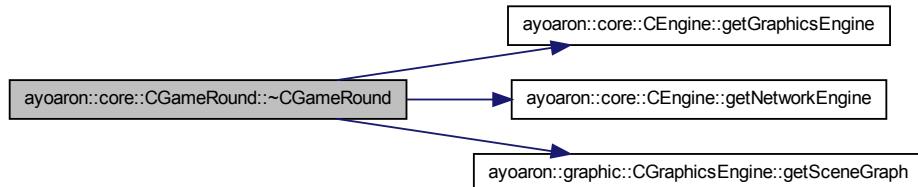


7.11.2.2 ayoaron ::core ::CGameRound ::~CGameRound()

destructeur

Références galax, ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph(), ayoaron ::reseau ::CNetworkEngine ::ni, et parent.

Voici le graphe d'appel pour cette fonction :



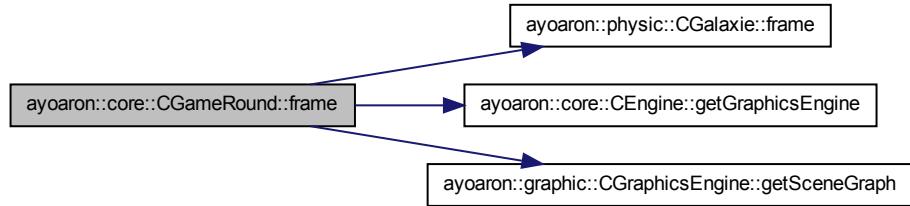
7.11.3 Documentation des fonctions membres

7.11.3.1 void ayoaron ::core ::CGameRound ::frame (double *dt*)

fonction de mise à jour

Références ayoaron ::physic ::CGalaxie ::frame(), galax, ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph(), et parent.

Voici le graphe d'appel pour cette fonction :



7.11.3.2 bool ayoaron ::core ::CGameRound ::isFull ()

la partie est-elle pleine

Renvoie

true si il n'y a plus de place disponible dans la partie

retourne si le jeu est plein ou non dépend de la mpa et des options

A faire

check du remplissage de la partie

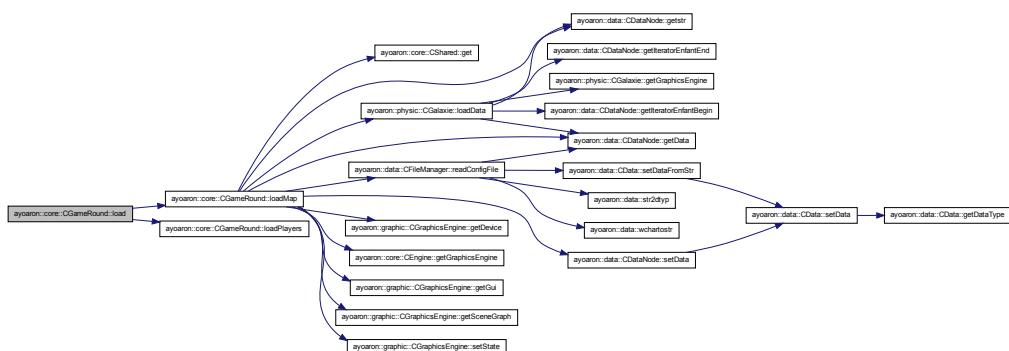
7.11.3.3 void ayoaron ::core ::CGameRound ::load ()

chargement du niveau de jeu

charge tous les éléments de la manche

Références ayoaron ::core ::CRoundOptions ::l_po, loadMap(), loadPlayers(), et ro.

Voici le graphe d'appel pour cette fonction :



7.11.3.4 void ayoaron ::core ::CGameRound ::loadMap (const std ::string & *fileconf*)
 [private]

chargement de map

Paramètres

in	<i>fileconf</i>	ma map à charger
----	-----------------	------------------

charge la map définie dans le round_option

chargement d'une map, on donnera une map par secteur

À faire

rendre dynamique la gestion du nom de la map

lecture de la base de données dans les fichiers galax

lecture de la base de données

charge la map

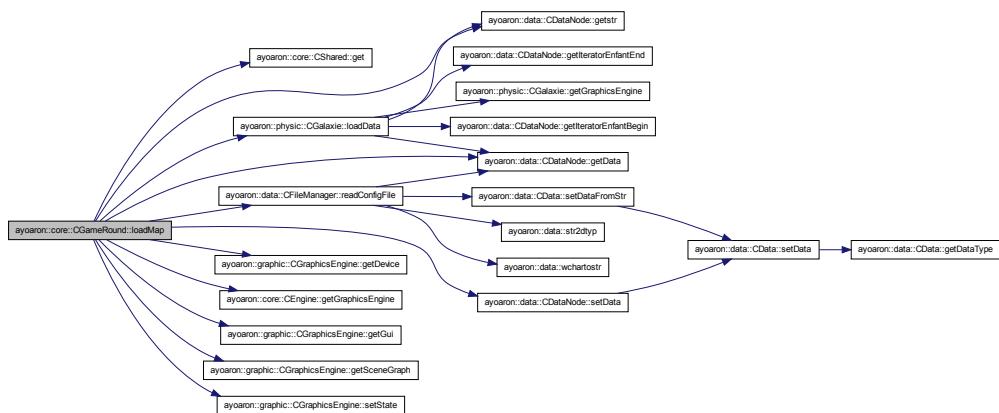
À faire

rendre dynamique le chargement des objets

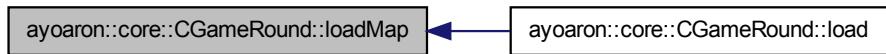
Références donn, galax, ayoaron ::core ::CShared ::get(), ayoaron ::data ::CDataNode ::getData(), ayoaron ::graphic ::CGraphicsEngine ::getDevice(), ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::graphic ::CGraphicsEngine ::getGui(), ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph(), ayoaron ::data ::CDataNode ::getstr(), ayoaron ::physic ::CGalaxie ::loadData(), ayoaron ::graphic ::CGraphicsEngine ::ON_GAME, parent, ayoaron ::data ::CFileManager ::readConfigFile(), ayoaron ::data ::CDataNode ::setData(), et ayoaron ::graphic ::CGraphicsEngine ::setState().

Référencé par load().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.11.3.5 `void ayoaron ::core ::CGameRound ::loadPlayers (std ::list< CPlayerOptions > & l) [private]`

chargement des joueurs

Paramètres

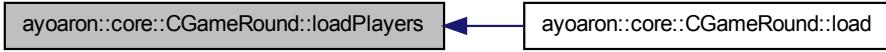
/ liste des joueurs à charger

Crée la liste des joueur prennant part à cette manche

Références ayoaron ::core ::CPlayerOptions ::IA_PLAYER, l_players, ayoaron ::core ::CPlayerOptions ::NETWORK_PLAYER, et ayoaron ::core ::CPlayerOptions ::SELF.

Référencé par load().

Voici le graphe d'appel pour cette fonction :



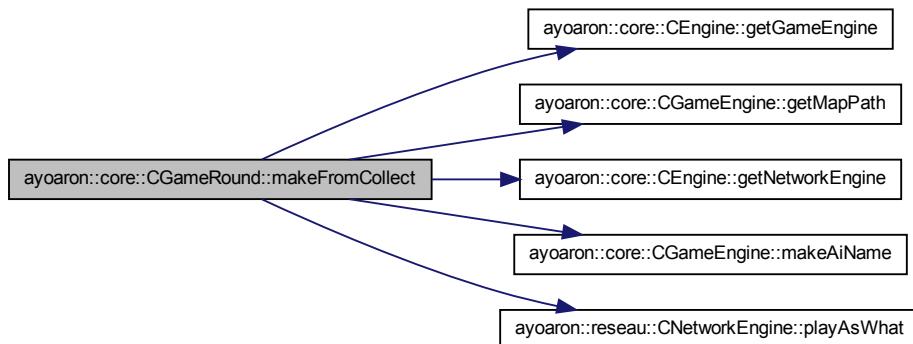
7.11.3.6 `void ayoaron ::core ::CGameRound ::makeFromCollect ()`

essaye de trouver tout seul toutes les options nécessaires à la création de la manche

Références ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::core ::CGameEngine ::getMapPath(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::core ::CPlayerOptions ::IA_PLAYER, ayoaron ::reseau ::CNetworkEngine ::INTERNET_CLIENT, ayoaron ::reseau ::CNetworkEngine ::INTERNET_SERVER, ayoaron ::core ::CRoundOptions ::l_difficulties, ayoaron ::reseau ::CNetworkEngine ::l_machines, ayoaron ::core ::CRoundOptions ::l_po, ayoaron ::reseau ::CNetworkEngine ::LAN_CLIENT, ayoaron ::reseau ::CNetworkEngine ::LAN_SERVER, ayoaron ::core ::CPlayerOptions ::level, ayoaron ::core ::CGameEngine ::makeAiName(), ayoaron ::core ::CRoundOptions ::map_name, ayoaron ::core ::CPlayerOptions ::NETWORK_PLAYER, ayoar-

```
ron ::reseau ::CNetworkEngine ::ni, ayoaron ::reseau ::CNetworkEngine ::NONE,
parent, ayoaron ::core ::CRoundOptions ::play_as, ayoaron ::reseau ::CNetworkEn-
gine ::playAsWhat(), ayoaron ::core ::CPlayerOptions ::player_name, ayoaron ::core ::CPlaye-
rOptions ::player_type, ayoaron ::core ::CPlayerOptions ::SELF, et ayoaron ::core ::CGa-
meEngine ::wip.
```

Voici le graphe d'appel pour cette fonction :



7.11.4 Documentation des données membres

7.11.4.1 data : CDataNode* ayoaron ::core ::CGameRound ::donn
[private]

structure de donnée

Référencé par CGameRound(), et loadMap().

7.11.4.2 physic : CGalaxie* ayoaron ::core ::CGameRound ::galax
[private]

pointeur vers la galaxie

Référencé par CGameRound(), frame(), loadMap(), et ~CGameRound().

7.11.4.3 std ::list<boost ::shared_ptr<CPlayer>>
ayoaron ::core ::CGameRound ::l_players [private]

liste des joueurs

Référencé par loadPlayers().

7.11.4.4 CGameEngine* ayoaron ::core ::CGameRound ::parent [private]

pointeur vers le moteur de jeu parent

Référencé par CGameRound(), frame(), loadMap(), makeFromCollect(), et ~CGameRound().

7.11.4.5 CRoundOptions ayoaron ::core ::CGameRound ::ro [private]

options de partie

Référencé par load().

7.11.4.6 int ayoaron ::core ::CGameRound ::state

état de la partie.

Référencé par CGameRound().

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

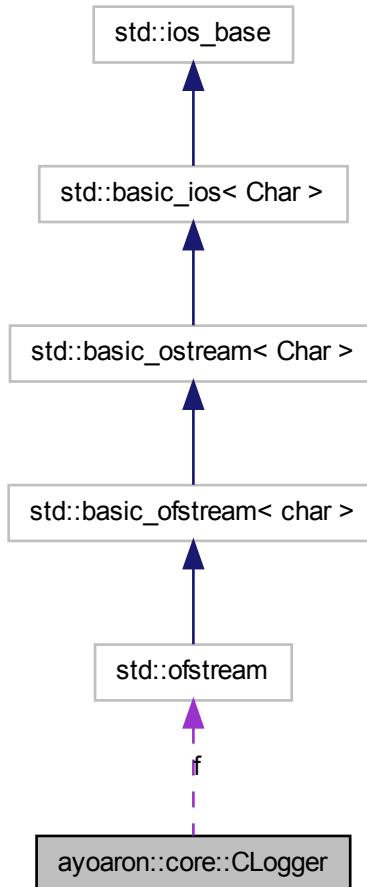
- [CGameRound.h](#)
- [CGameRound.cpp](#)

7.12 Référence de la classe ayoaron ::core ::CLogger

module de log

```
#include <CLogger.h>
```

Graphe de collaboration de ayoaron ::core ::CLogger :



Types publics

- enum { `ONLY_ERRORS`, `WARNING_OR_ERRORS`, `ALL` }

Fonctions membres publiques

- `CLogger` (int)
- `std::ofstream & log ()`
fonction de log
- `void logEvent (const CEGUI ::String &, CEGUI ::LoggingLevel)`
- `void setLogFilename (const CEGUI ::String &, bool)`

- virtual `~CLogger ()`

Attributs privés

- std ::ofstream * `f`
- int `type`

7.12.1 Description détaillée

module de log

7.12.2 Documentation des énumérations membres

7.12.2.1 anonymous enum

Valeurs énumérées :

ONLY_ERRORS

WARNING_OR_ERRORS

ALL

7.12.3 Documentation des constructeurs et destructeur

7.12.3.1 ayoaron ::core ::CLogger (int `t`)

constructeur

Paramètres

in	<code>t</code>	precision du logger
----	----------------	---------------------

Références `f`, et type.

7.12.3.2 ayoaron ::core ::CLogger ::~CLogger () [virtual]

destructeur

Références `f`.

7.12.4 Documentation des fonctions membres

7.12.4.1 std ::ofstream& ayoaron ::core ::CLogger ::log () [inline]

fonction de log

Références `f`.

7.12.4.2 void ayoaron ::core ::CLogger ::logEvent (const CEGUI ::String & , CEGUI ::LoggingLevel) [inline]

7.12.4.3 void ayoaron ::core ::CLogger ::setLogFilename (const CEGUI ::String & , bool)
[inline]

7.12.5 Documentation des données membres

7.12.5.1 std ::ofstream* ayoaron ::core ::CLogger ::f [private]

Référencé par CLogger(), log(), et ~CLogger().

7.12.5.2 int ayoaron ::core ::CLogger ::type [private]

Référencé par CLogger().

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

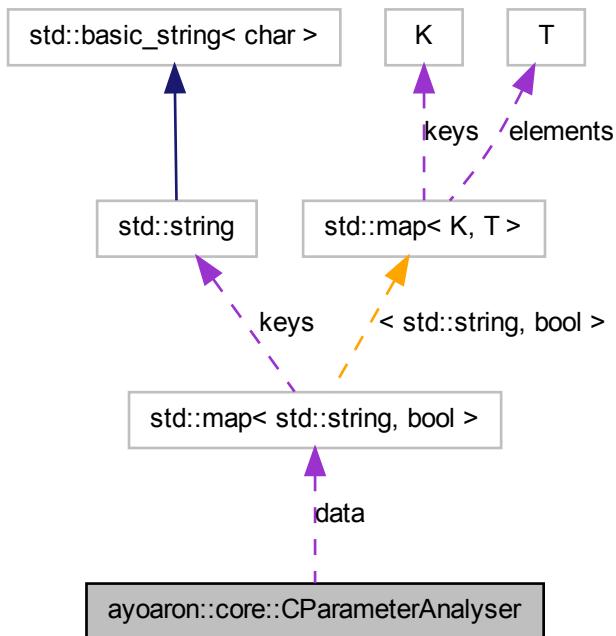
- [CLogger.h](#)
- [CLogger.cpp](#)

7.13 Référence de la classe ayoaron ::core ::CParameterAnalyser

Classe d'analyse des paramètres.

```
#include <CParameterAnalyser.h>
```

Graphe de collaboration de ayoaron ::core ::CParameterAnalyser :



Fonctions membres publiques

- [CParameterAnalyser \(\)](#)
Constructeur par défaut.
- [CParameterAnalyser \(int, char **\)](#)
Constructeur.

Attributs publics

- `std ::map< std ::string, bool > data`
Liste des paramètres.

7.13.1 Description détaillée

Classe d'analyse des paramètres.

Cette classe transforme les paramètres d'entrée sous forme argc/argv en une forme de tableau de donnée std ::string

7.13.2 Documentation des constructeurs et destructeur

7.13.2.1 ayoaron ::core ::CParameterAnalyser ::CParameterAnalyser ()

Constructeur par defaut.

Constructeur par defaut de la classe [CParameterAnalyser](#)

7.13.2.2 ayoaron ::core ::CParameterAnalyser ::CParameterAnalyser (int argc, char ** argv)

Constructeur.

Paramètres

in	argc	Nombre de mot dans argv
in	argv	Pointeur sur la liste de mots

Constructeur de la classe [CParameterAnalyser](#) remplit le tableau de données

Références data.

7.13.3 Documentation des données membres

7.13.3.1 std ::map<std ::string, bool> ayoaron ::core ::CParameterAnalyser ::data

Liste des paramètres.

Référencé par ayoaron ::core ::CGame ::CGame(), CParameterAnalyser(), et ayoaron ::core ::CGame ::run().

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

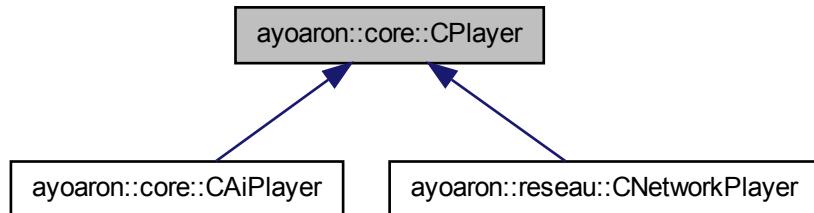
- [CParameterAnalyser.h](#)
- [CParameterAnalyser.cpp](#)

7.14 Référence de la classe ayoaron ::core ::CPlayer

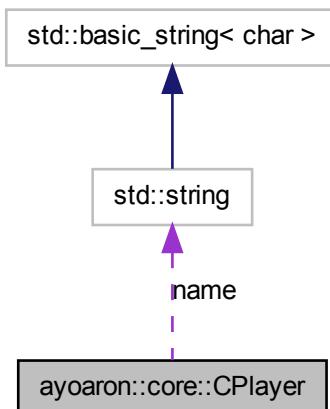
l'objet joueur

```
#include <CPlayer.h>
```

Graphe d'héritage de ayoaron ::core ::CPlayer :



Graphe de collaboration de ayoaron ::core ::CPlayer :



Fonctions membres publiques

- `CPlayer ()`
constructeur
- `CPlayer (CPlayerOptions *)`
constructeur
- `std::string & getName ()`
renvoie le nom du joueur
- `~CPlayer ()`

destructeur

Attributs privés

- std ::string **name**
nom du joueur

7.14.1 Description détaillée

l'objet joueur

7.14.2 Documentation des constructeurs et destructeur

7.14.2.1 ayoaron ::core ::CPlayer ::CPlayer ()

constructeur

7.14.2.2 ayoaron ::core ::CPlayer ::CPlayer (CPlayerOptions * *po*)

constructeur

Paramètres

in	<i>po</i>	les option que l'on attache au joueur
----	-----------	---------------------------------------

Références name, et ayoaron ::core ::CPlayerOptions ::player_name.

7.14.2.3 ayoaron ::core ::CPlayer ::~CPlayer ()

destructeur

7.14.3 Documentation des fonctions membres

7.14.3.1 std ::string& ayoaron ::core ::CPlayer ::getName () [inline]

renvoie le nom du joueur

Références name.

7.14.4 Documentation des données membres

7.14.4.1 std ::string ayoaron ::core ::CPlayer ::name [private]

nom du joueur

Référencé par CPlayer(), et getName().

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

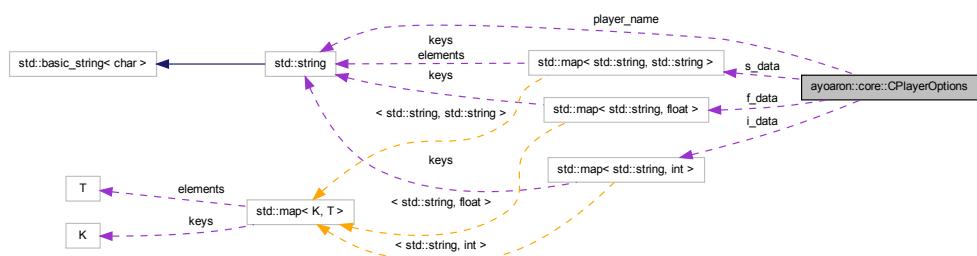
- [CPlayer.h](#)
- [CPlayer.cpp](#)

7.15 Référence de la classe ayoaron ::core ::CPlayerOptions

rassemblement des options joueur

```
#include <CPlayerOptions.h>
```

Graphe de collaboration de ayoaron ::core ::CPlayerOptions :



Types publics

- enum { **SELF**, **NETWORK_PLAYER**, **IA_PLAYER** }
type de joueur

Attributs publics

- std ::map< std ::string, float > **f_data**
autre donnée de type float
- std ::map< std ::string, int > **i_data**
autre donnée de type int
- int **level**
- std ::string **player_name**
nom du joueur
- int **player_type**
type de joueur
- std ::map< std ::string, std ::string > **s_data**
autre donnée de type string
- int **team_nb**
numéro d'équipe

7.15.1 Description détaillée

rassemblement des options joueur

Classe contenant toutes les options requises pour créer et jouer un joueur

7.15.2 Documentation des énumérations membres

7.15.2.1 anonymous enum

type de joueur

Valeurs énumérées :

SELF

NETWORK_PLAYER

IA_PLAYER

7.15.3 Documentation des données membres

7.15.3.1 `std ::map<std ::string, float> ayoaron ::core ::CPlayerOptions ::f_data`

autre donnée de type float

7.15.3.2 `std ::map<std ::string, int> ayoaron ::core ::CPlayerOptions ::i_data`

autre donnée de type int

7.15.3.3 `int ayoaron ::core ::CPlayerOptions ::level`

Référencé par `ayoaron ::core ::CAiPlayer ::CAiPlayer()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.15.3.4 `std ::string ayoaron ::core ::CPlayerOptions ::player_name`

nom du joueur

Référencé par `ayoaron ::core ::CPlayer ::CPlayer()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.15.3.5 `int ayoaron ::core ::CPlayerOptions ::player_type`

type de joueur

Référencé par `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.15.3.6 `std ::map<std ::string, std ::string> ayoaron ::core ::CPlayerOptions ::s_data`

autre donnée de type string

7.15.3.7 `int ayoaron ::core ::CPlayerOptions ::team_nb`

numéro d'équipe

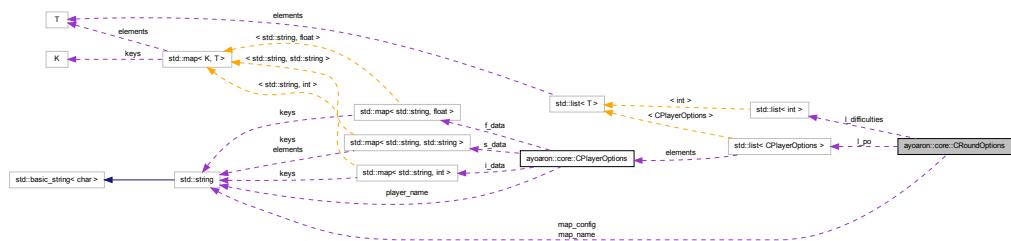
La documentation de cette classe a été générée à partir du fichier suivant :

- [CPlayerOptions.h](#)

7.16 Référence de la classe ayoaron ::core ::CRoundOptions

```
#include <CRoundOptions.h>
```

Graphe de collaboration de ayoaron ::core ::CRoundOptions :



Types publics

- enum { `AI_EASY`, `AI_MEDIUM`, `AI_HARD` }
liste des difficultées

Fonctions membres publiques

- void `clear ()`
réinitialise les options
- `CRoundOptions ()`
constructeur

Attributs publics

- `std::list< int > l_difficulties`
niveau de difficulté
- `std::list< CPlayerOptions > l_po`
liste des joueurs
- `std::string map_config`
fichier objet
- `std::string map_name`
nom de la carte
- `int play_as`
type de partie

7.16.1 Documentation des énumérations membres

7.16.1.1 anonymous enum

liste des difficultées

Valeurs énumérées :

`AI_EASY`

AI_MEDIUM

AI_HARD

7.16.2 Documentation des constructeurs et destructeur

7.16.2.1 **ayoaron ::core ::CRoundOptions ::CRoundOptions ()** [inline]
constructeur

7.16.3 Documentation des fonctions membres

7.16.3.1 **void ayoaron ::core ::CRoundOptions ::clear ()** [inline]
réinitialise les options
Références `l_difficulties`, `l_po`, `map_name`, et `play_as`.

7.16.4 Documentation des données membres

7.16.4.1 **std ::list<int> ayoaron ::core ::CRoundOptions ::l_difficulties**
nniveau de difficulté
Référencé par `clear()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.16.4.2 **std ::list<CPlayerOptions> ayoaron ::core ::CRoundOptions ::l_po**
liste des joueurs

Référencé par `clear()`, `ayoaron ::core ::CGameRound ::load()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.16.4.3 **std ::string ayoaron ::core ::CRoundOptions ::map_config**
fichier objet

7.16.4.4 **std ::string ayoaron ::core ::CRoundOptions ::map_name**
nom de la carte
Référencé par `clear()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.

7.16.4.5 **int ayoaron ::core ::CRoundOptions ::play_as**
type de partie

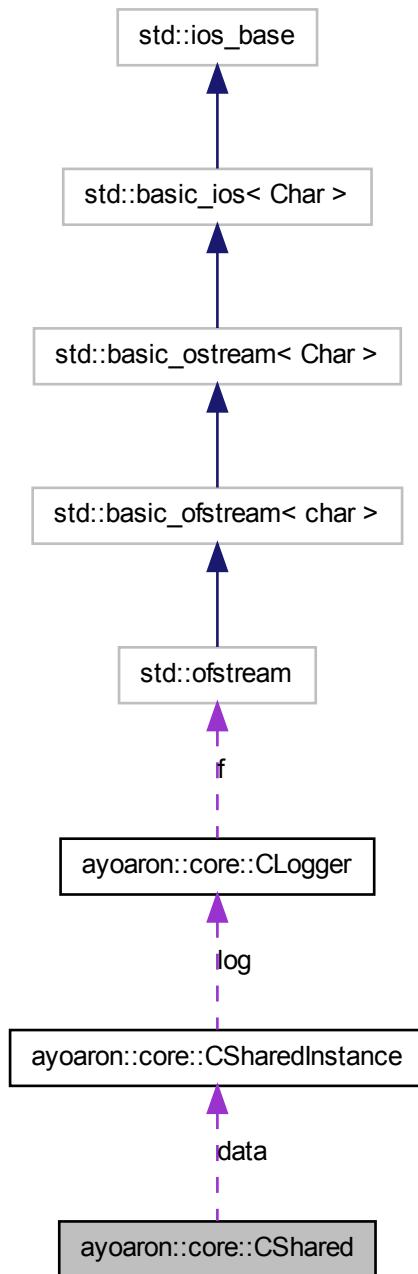
Référencé par `clear()`, et `ayoaron ::core ::CGameRound ::makeFromCollect()`.
La documentation de cette classe a été générée à partir du fichier suivant :
– [CRoundOptions.h](#)

7.17 Référence de la classe ayoaron ::core ::CShared

singleton

```
#include <CShared.h>
```

Graphe de collaboration de ayoaron ::core ::CShared :



Fonctions membres publiques statiques

- static **CSharedInstance** * **get** ()
recupère les donnée
- static void **make** ()
constructeur du bloc de donnée
- static void **unmake** ()
destructeur du bloc de donnée

Fonctions membres protégées

- **CShared** ()

Attributs privés statiques

- static **CSharedInstance** * **data**
données

7.17.1 Description détaillée

singleton

make et unmake sont les constructeur/destructeur appeler make avant to appel à get

7.17.2 Documentation des constructeurs et destructeur**7.17.2.1 ayoaron ::core ::CShared ::CShared() [protected]**

constructeur

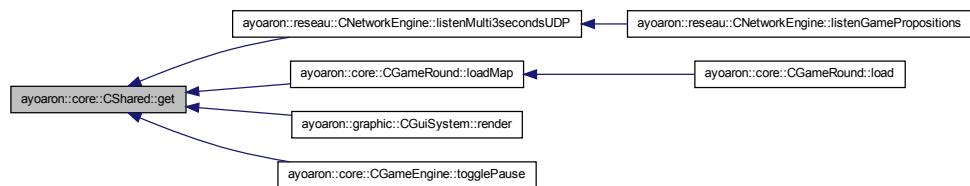
7.17.3 Documentation des fonctions membres**7.17.3.1 static CSharedInstance* ayoaron ::core ::CShared ::get() [inline, static]**

recupère les donnée

Références data.

Référencé par ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::graphic ::CGuiSystem ::render(), et ayoaron ::core ::CGameEngine ::togglePause().

Voici le graphe d'appel pour cette fonction :



7.17.3.2 static void ayoaron ::core ::CShared ::make () [inline, static]

constructeur du bloc de donnée

Références data.

Référencé par `ayoaron ::core ::CGame ::CGame()`.

Voici le graphe d'appel pour cette fonction :



7.17.3.3 static void ayoaron ::core ::CShared ::unmake () [inline, static]

destructeur du bloc de donnée

Références data.

Référencé par `ayoaron ::core ::CGame ::~CGame()`.

Voici le graphe d'appel pour cette fonction :



7.17.4 Documentation des données membres

7.17.4.1 **CSharedInstance * ayoaron ::core ::CShared ::data** [static, private]

données

Référencé par get(), make(), et unmake().

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

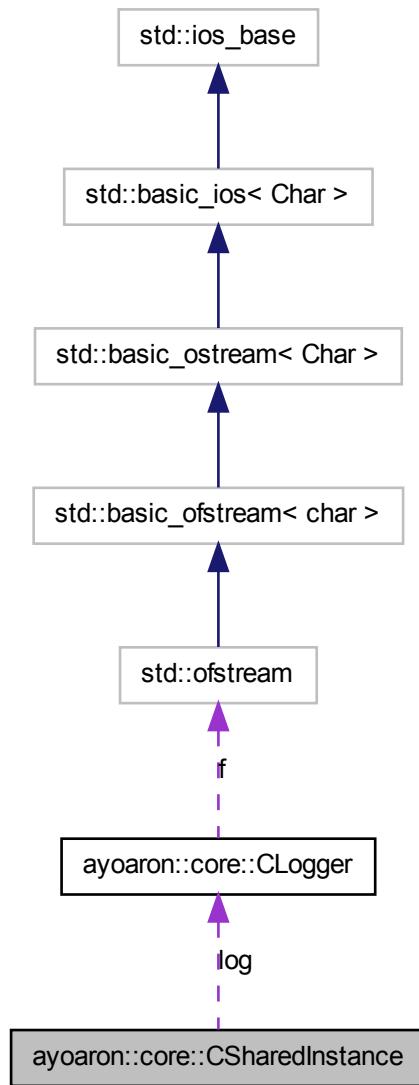
- [CShared.h](#)
- [CShared.cpp](#)

7.18 Référence de la classe ayoaron ::core ::CSharedInstance

donnée stockée dans le singleton

```
#include <CShared.h>
```

Graphe de collaboration de ayoaron ::core ::CSharedInstance :



Fonctions membres publiques

- `CSharedInstance ()`
constructeur
- `~CSharedInstance ()`

Attributs publics

- **CLogger log**
le logger utilisé
- **boost ::mutex mutex_gui**
mutex de l'interface utilisateur

7.18.1 Description détaillée

donnée stockée dans le singleton
on peut ajouter ce qu'on veut

7.18.2 Documentation des constructeurs et destructeur**7.18.2.1 ayoaron ::core ::CSharedInstance ::CSharedInstance ()**

constructeur

7.18.2.2 ayoaron ::core ::CSharedInstance ::~CSharedInstance ()

destructeur

7.18.3 Documentation des données membres**7.18.3.1 CLogger ayoaron ::core ::CSharedInstance ::log**

le logger utilisé

7.18.3.2 boost ::mutex ayoaron ::core ::CSharedInstance ::mutex_gui

mutex de l'interface utilisateur

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

- [CShared.h](#)
- [CShared.cpp](#)

7.19 Référence de la classe ayoaron ::core ::CXMLExtractor

cree les interfaces XML

```
#include <CXMLExtractor.h>
```

Fonctions membres publiques

- **CXMLExtractor (const std ::string &)**
constructeur
- **CAvailableAi * getAi ()**
cree une ai depuis un fichier xml
- **CAvailableMap * getMap ()**

- `~CXMLExtractor ()`
destructeur

Attributs privés

- `CEGUI ::XMLHandler hdl`
conteneur xml, dépendant du parserxml choisi

7.19.1 Description détaillée

cree les interfaces XML

Classe d'usine. Cree toutes les instances d'objet xml depuis un fichier xml. toute les acquisitions xml sont faite ici.

7.19.2 Documentation des constructeurs et destructeur

7.19.2.1 ayoaron ::core ::CXMLExtractor ::CXMLExtractor (const std ::string & filename)

constructeur

Paramètres

in	<i>filename</i>	nom du fichier xml à traiter
----	-----------------	------------------------------

Références hdl.

7.19.2.2 ayoaron ::core ::CXMLExtractor ::~CXMLExtractor ()

destructeur

7.19.3 Documentation des fonctions membres

7.19.3.1 CAvailableAi * ayoaron ::core ::CXMLExtractor ::getAi ()

cree une ai depuis un fichier xml

essaye de trouver toutes les données requises pour construire une available_ai si une donnée requise ne peut être fournie, on jette une exception

7.19.3.2 CAvailableMap * ayoaron ::core ::CXMLExtractor ::getMap ()

cree une map depuis un fichier xml

essaye de trouver toutes les données requises pour construire une available_map si une donnée requise ne peut être fournie, on jette une exception

Références hdl.

7.19.4 Documentation des données membres

7.19.4.1 CEGUI ::XMLHandler ayoaron ::core ::CXMLExtractor ::hdl [private]

conteneur xml, dépendant du parserxml choisi

Référencé par CXMLExtractor(), et getMap().

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

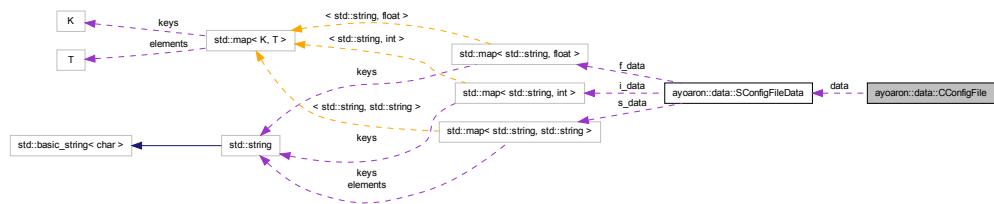
- [CXMLExtractor.h](#)
- [CXMLExtractor.cpp](#)

7.20 Référence de la classe ayoaron ::data ::CConfigFile

Traitement des fichiers de config.

```
#include <CConfigFile.h>
```

Graphe de collaboration de ayoaron ::data ::CConfigFile :



Fonctions membres publiques

- **CConfigFile ()**
constructeur
- **SConfigFileData getData ()**
récupération des données du fichier
- **bool isIn (const std ::string &s)**
dit si une clé est présente dans les données
- **void parse (const std ::string &filename)**
examen de fichier
- **~CConfigFile ()**
destructeur

Fonctions membres privées

- template<typename T >
bool isOfType (const std ::string &)
renvoie le type du paramètre

Attributs privés

- [SConfigFileData data](#)
les données du fichier

7.20.1 Description détaillée

Traitement des fichiers de config.

7.20.2 Documentation des constructeurs et destructeur**7.20.2.1 ayoaron ::data ::CConfigFile ()**

constructeur

7.20.2.2 ayoaron ::data ::CConfigFile ::~CConfigFile ()

destructeur

7.20.3 Documentation des fonctions membres**7.20.3.1 SConfigFileData ayoaron ::data ::CConfigFile ::getData ()**

récupération des données du fichier

Renvoie

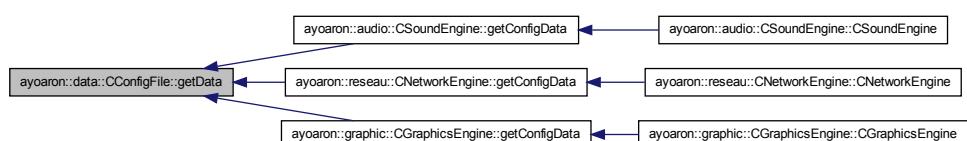
Liste des paramètres du fichier

renvoie de donnée basique

Références data.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::reseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

Voici le graphe d'appel pour cette fonction :



7.20.3.2 bool ayoaron ::data ::CConfigFile ::isIn (const std ::string & s)

dit si une clé est présente dans les données

Paramètres

in	s	la clé à rechercher
----	---	---------------------

Renvoie

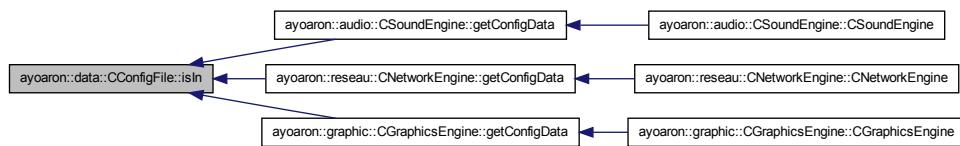
true si la clé est présente, false sinon

Revoie si la clé existe dans l'une des map

Références data, ayoaron ::data ::SConfigFileData ::f_data, ayoaron ::data ::SConfigFileData ::i_data, et ayoaron ::data ::SConfigFileData ::s_data.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::reseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

Voici le graphe d'appel pour cette fonction :



7.20.3.3 template<typename T > bool ayoaron ::data ::CConfigFile ::isOfType (const std ::string & Str) [private]

renvoie le type du paramètre

Paramètres

in	Str	la clé à évaluer
----	-----	------------------

Renvoie

le type de donnée

pris sur developpez.com pour savoir si le parametre float/int

7.20.3.4 void ayoaron ::data ::CConfigFile ::parse (const std ::string & filename)

examun de fichier

Paramètres

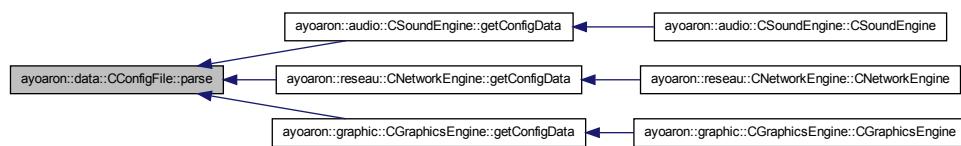
in	filename	nom du fichier à traiter
----	----------	--------------------------

Examine le fichier et stocke les donnée dans les différente map en fonction de leur type

Références data, ayoaron ::data ::SConfigFileData ::f_data, ayoaron ::data ::SConfigFileData ::i_data, et ayoaron ::data ::SConfigFileData ::s_data.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::reseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

Voici le graphe d'appel pour cette fonction :



7.20.4 Documentation des données membres

7.20.4.1 SConfigFileData ayoaron ::data ::CConfigFile ::data [private]

les données du fichier

Référencé par `getData()`, `isIn()`, et `parse()`.

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

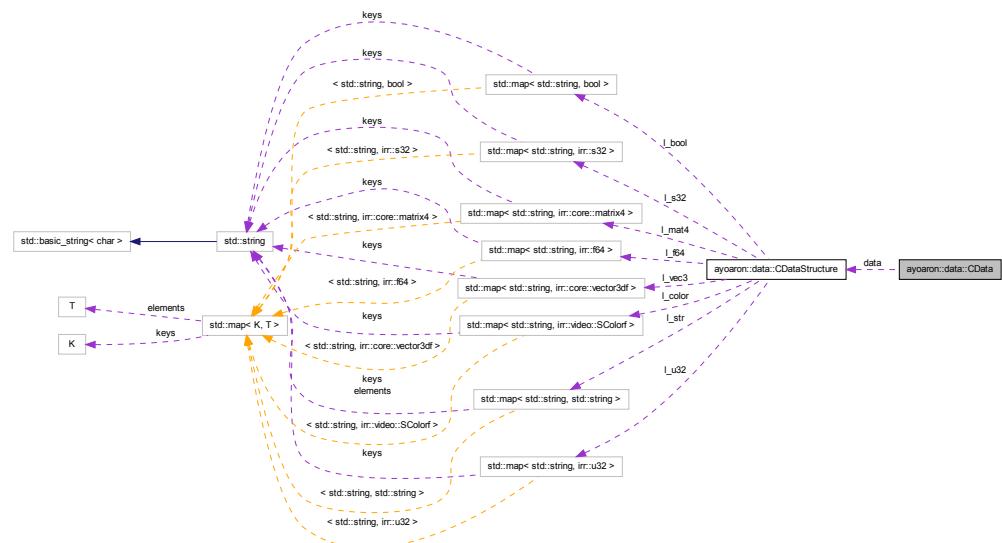
- [CConfigFile.h](#)
- [CConfigFile.cpp](#)

7.21 Référence de la classe ayoaron ::data ::CData

classe de manipulation de données

```
#include <CData.h>
```

Graphe de collaboration de ayoaron : :data : :CData :



Fonctions membres publiques

- **CData ()**
Constructeur.
 - **void drop (std :: string key="")**
detruit la donnée key
 - **bool getbool (std :: string key)**
renvoie la valeur bool correspondant à la clé
 - **irr :: video :: SColorf getcol (std :: string key)**
renvoie la valeur color correspondant à la clé
 - **std :: list< std :: string > getDataList ()**
renvoie une liste de chaîne des clés de donnée
 - **DATATYPE getDataType (std :: string key)**
renvoie le type de donnée
 - **irr :: f64 getf64 (std :: string key)**
renvoie la valeur float correspondant à la clé
 - **irr :: core :: matrix4 getmat4 (std :: string key)**
renvoie la valeur matrix correspondant à la clé
 - **irr :: s32 gets32 (std :: string key)**
renvoie la valeur sint correspondant à la clé
 - **std :: string getstr (std :: string key)**
renvoie la valeur str correspondant à la clé
 - **std :: string getStrFromData (std :: string key)**
renvoie une chaîne permettant l'écriture des données
 - **irr :: u32 getu32 (std :: string key)**
renvoie la valeur uint correspondant à la clé
 - **irr :: core :: vector3df getvec3 (std :: string key)**
renvoie la valeur vecteur correspondant à la clé
 - **void setData (std :: string key, irr :: core :: matrix4 val)**
ajoute ou modifie une donnée de type matrice 4x4. Les clés sont unique à un type
 - **void setData (std :: string key, irr :: s32 val)**

- ajoute ou modifie une donnée de type entier signé. Les clés sont unique à un type
- void **setData** (std ::string key, bool val)
 - ajoute ou modifie une donnée de type booléen. Les clés sont unique à un type
- void **setData** (std ::string key, irr ::core ::vector3df val)
 - ajoute ou modifie une donnée de type vecteur. Les clés sont unique à un type
- void **setData** (std ::string key, irr ::video ::SColorf val)
 - ajoute ou modifie une donnée de type couleur. Les clés sont unique à un type
- void **setData** (std ::string key, irr ::u32 val)
 - ajoute ou modifie une donnée de type entier non signé. Les clés sont unique à un type
- void **setData** (std ::string key, std ::string val)
 - ajoute ou modifie une donnée de type chaîne. Les clés sont unique à un type
- void **setData** (std ::string key, irr ::f64 val)
 - ajoute ou modifie une donnée de type réel. Les clés sont unique à un type
- void **setDataFromStr** (std ::string key, std ::string val, DATATYPE typ=NOTYPE)
 - lit une chaîne de caractère et en déduit la donnée à ajouter
- virtual **~CData** ()
 - Destructeur.

Attributs privés

- **CDataStructure data**
 - les données

7.21.1 Description détaillée

classe de manipulation de données

7.21.2 Documentation des constructeurs et destructeur

7.21.2.1 ayoaron ::data ::CData ::CData ()

Constructeur.

Références drop().

Voici le graphe d'appel pour cette fonction :



7.21.2.2 ayoaron ::data ::CData ::~CData () [virtual]

Destructeur.

Références drop().

Voici le graphe d'appel pour cette fonction :



7.21.3 Documentation des fonctions membres

7.21.3.1 void ayoaron ::data ::CData ::drop (std ::string key = " ")

détruit la donnée key

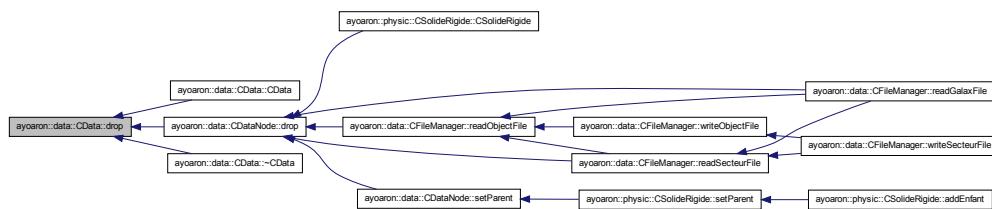
Paramètres

in	key	la clé vers la donnée, si vide, alors détruit tout
----	-----	--

Références data, ayoaron ::data ::CDataStructure ::l_bool, ayoaron ::data ::CDataStructure ::l_color, ayoaron ::data ::CDataStructure ::l_f64, ayoaron ::data ::CDataStructure ::l_mat4, ayoaron ::data ::CDataStructure ::l_s32, ayoaron ::data ::CDataStructure ::l_str, ayoaron ::data ::CDataStructure ::l_u32, et ayoaron ::data ::CDataStructure ::l_vec3.

Référencé par CData(), ayoaron ::data ::CDataNode ::drop(), et ~CData().

Voici le graphe d'appel pour cette fonction :



7.21.3.2 bool ayoaron ::data ::CData ::getbool (std ::string key)

renvoie la valeur bool correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

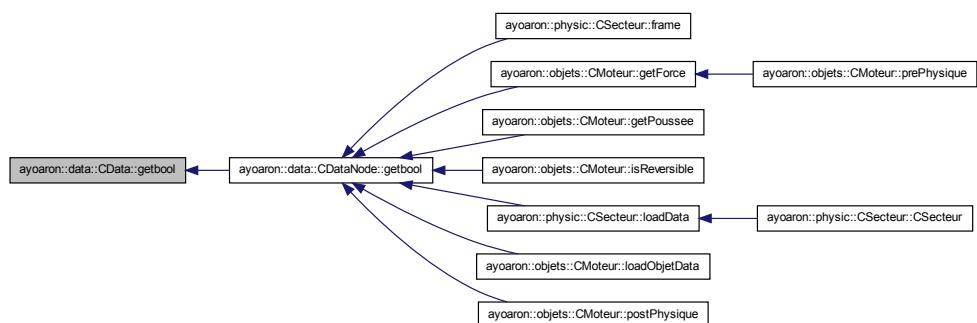
Renvoie

la valeur booléenne correspondant à la clé, renvoie false si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_bool.

Référencé par ayoaron ::data ::CDataNode ::getbool().

Voici le graphe d'appel pour cette fonction :

**7.21.3.3 irr ::video ::SColorf ayoaron ::data ::CData ::getcol (std ::string key)**

renvoie renvoie la valeur color correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur de couleur correspondant à la clé, renvoie la couleur noire (0,0,0,0) si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_color.

Référencé par ayoaron ::data ::CDataNode ::getcol().

Voici le graphe d'appel pour cette fonction :



7.21.3.4 std ::list< std ::string > ayoaron ::data ::CData ::getDataList()

renvoie une liste de chaîne des clés de donnée

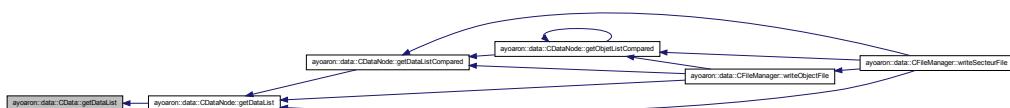
Renvoie

une liste de toutes les clés valides dans data

Références data, ayoaron ::data ::CDataStructure ::l_bool, ayoaron ::data ::CDataStructure ::l_color, ayoaron ::data ::CDataStructure ::l_f64, ayoaron ::data ::CDataStructure ::l_mat4, ayoaron ::data ::CDataStructure ::l_s32, ayoaron ::data ::CDataStructure ::l_str, ayoaron ::data ::CDataStructure ::l_u32, et ayoaron ::data ::CDataStructure ::l_vec3.

Référencé par ayoaron ::data ::CDataNode ::getDataList().

Voici le graphe d'appel pour cette fonction :



7.21.3.5 DATATYPE ayoaron ::data ::CData ::getDataType(std ::string key)

revoie le type de donnée

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

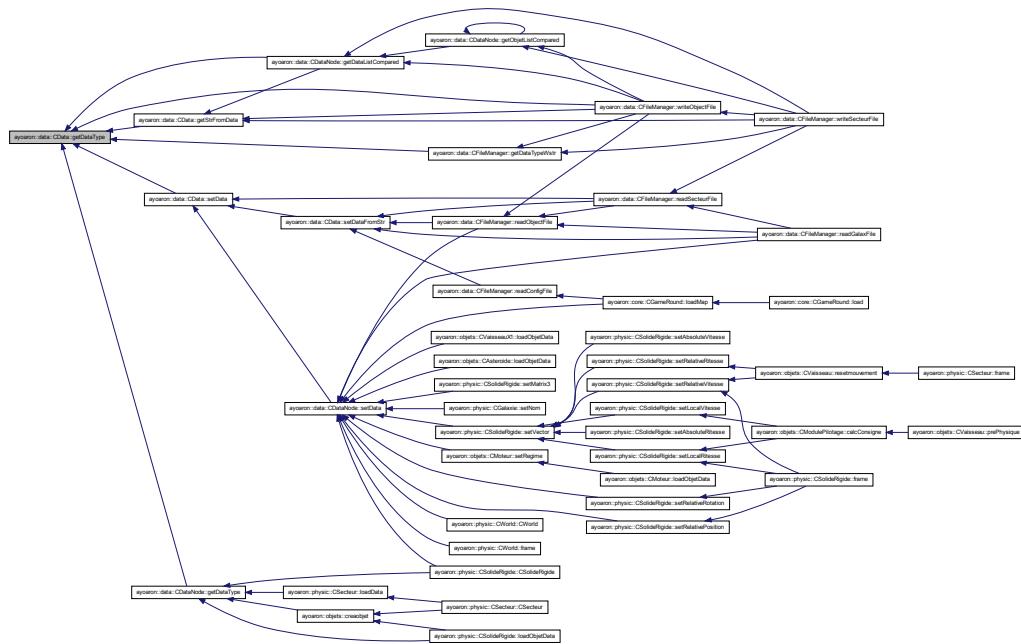
Renvoie

le type de donnée

Références ayoaron ::data ::BOOL, ayoaron ::data ::COLOR, data, ayoaron ::data ::FLOAT, ayoaron ::data ::CDataStructure ::l_bool, ayoaron ::data ::CDataStructure ::l_color, ayoaron ::data ::CDataStructure ::l_f64, ayoaron ::data ::CDataStructure ::l_mat4, ayoaron ::data ::CDataStructure ::l_s32, ayoaron ::data ::CDataStructure ::l_str, ayoaron ::data ::CDataStructure ::l_u32, ayoaron ::data ::CDataStructure ::l_vec3, ayoaron ::data ::MATRIX, ayoaron ::data ::NOTYPE, ayoaron ::data ::SINT, ayoaron ::data ::STRING, ayoaron ::data ::UINT, et ayoaron ::data ::VECTOR.

Référencé par ayoaron ::data ::CDataNode ::getDataListCompared(), ayoaron ::data ::CDataNode ::getDataType(), ayoaron ::data ::CFileManager ::getDataTypeWstr(), getStringFromData(), setData(), et ayoaron ::data ::CFileManager ::writeObjectFile().

Voici le graphe d'appel pour cette fonction :



7.21.3.6 irr : :f64 ayoaron ::data ::CData ::getf64 (std ::string key)

renvoie renvoie la valeur float correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

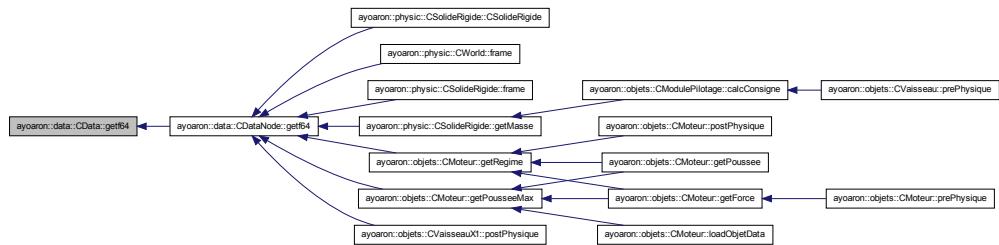
Renvoie

la valeur réelle correspondant à la clé, renvoie 0 si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::_f64.

Référencé par ayoaron ::data ::CDataNode ::getf64().

Voici le graphe d'appel pour cette fonction :



7.21.3.7 irr ::core ::matrix4 ayoaron ::data ::CData ::getmat4 (std ::string key)

renvoie renvoie la valeur matrix correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

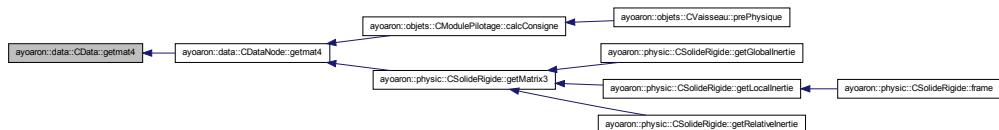
Renvoie

la valeur matricielle correspondant à la clé, renvoie la matrice nulle si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_mat4.

Référencé par ayoaron ::data ::CDataNode ::getmat4().

Voici le graphe d'appel pour cette fonction :



7.21.3.8 irr ::s32 ayoaron ::data ::CData ::gets32 (std ::string key)

renvoie renvoie la valeur sint correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur entière signée correspondant à la clé, renvoie 0 si la clé n'existe pas ou

n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_s32.

Référencé par ayoaron ::data ::CDataNode ::gets32().

Voici le graphe d'appel pour cette fonction :



7.21.3.9 std ::string ayoaron ::data ::CData ::getstr (std ::string key)

renvoie la valeur str correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

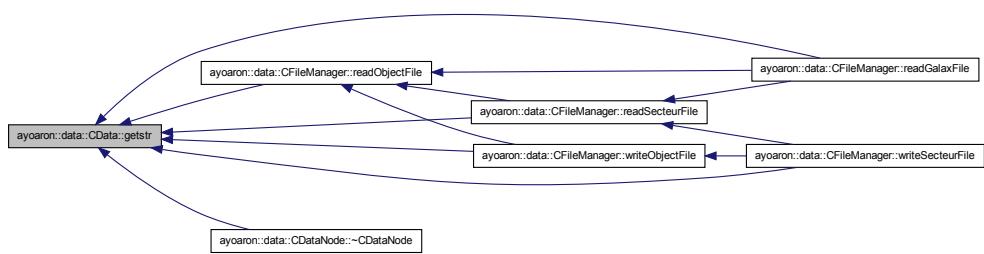
Renvoie

la valeur chaîne correspondant à la clé, renvoie une chaîne vide si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_str.

Référencé par ayoaron ::data ::CFileManager ::readGalaxyFile(), ayoaron ::data ::CFileManager ::readObjectFile(), ayoaron ::data ::CFileManager ::readSectorFile(), ayoaron ::data ::CFileManager ::writeObjectFile(), ayoaron ::data ::CFileManager ::writeSectorFile(), et ayoaron ::data ::CDataNode ::~CDataNode().

Voici le graphe d'appel pour cette fonction :



7.21.3.10 std ::string ayoaron ::data ::CData ::getStrFromData (std ::string key)

renvoie une chaîne permettant l'écriture des données

Paramètres

key	la clé à écrire
-----	-----------------

Renvoie

une chaîne correspondant à l'écriture de la clé

Références ayoaron ::data ::BOOL, ayoaron ::data ::COLOR, data, ayoaron ::data ::FLOAT, getDataType(), ayoaron ::data ::CDataStructure ::l_bool, ayoaron ::data ::CDataStructure ::l_color, ayoaron ::data ::CDataStructure ::l_f64, ayoaron ::data ::CDataStructure ::l_mat4, ayoaron ::data ::CDataStructure ::l_s32, ayoaron ::data ::CDataStructure ::l_str, ayoaron ::data ::CDataStructure ::l_u32, ayoaron ::data ::CDataStructure ::l_vec3, ayoaron ::data ::MATRIX, ayoaron ::data ::NOTYPE, ayoaron ::data ::SINT, ayoaron ::data ::STRING, ayoaron ::data ::UINT, et ayoaron ::data ::VECTOR.

Référencé par ayoaron ::data ::CDataNode ::getDataListCompared(), ayoaron ::data ::CFFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.21.3.11 irr ::u32 ayoaron ::data ::CData ::getu32 (std ::string key)

renvoie la valeur uint correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur entière non signée correspondant à la clé, renvoie 0 si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_u32.

Référencé par ayoaron ::data ::CDataNode ::getu32().

Voici le graphe d'appel pour cette fonction :

**7.21.3.12 irr ::core ::vector3df ayoaron ::data ::CData ::getvec3 (std ::string key)**

renvoie renvoie la valeur vecteur correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

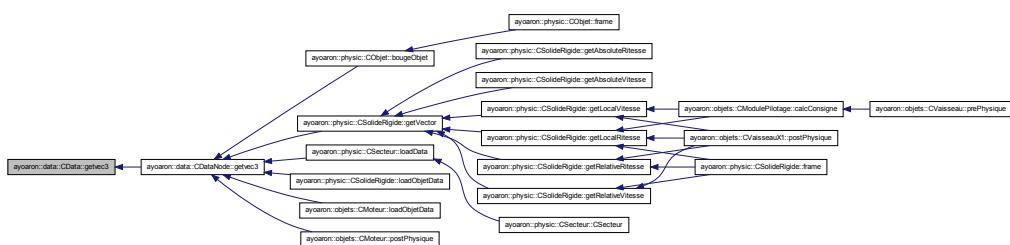
Renvoie

la valeur vectorielle correspondant à la clé, renvoie (0,0,0) si la clé n'existe pas ou n'est pas du bon type

Références data, et ayoaron ::data ::CDataStructure ::l_vec3.

Référencé par ayoaron ::data ::CDataNode ::getvec3().

Voici le graphe d'appel pour cette fonction :

**7.21.3.13 void ayoaron ::data ::CData ::setData (std ::string key, irr ::video ::SColorf val)**

ajoute ou modifie une donnée de type couleur. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

in	val	la valeur de couleur à affecter à la clé
----	-----	--

Références ayoaron ::data ::COLOR, data, getDataType(), ayoaron ::data ::CData-Structure ::l_color, et ayoaron ::data ::NOTYPE.

Voici le graphe d'appel pour cette fonction :



7.21.3.14 void ayoaron ::data ::CData ::setData (std ::string key, std ::string val)

ajoute ou modifie une donnée de type chaîne. Les clés sont uniques à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur chaîne à affecter à la clé

Références data, getDataType(), ayoaron ::data ::CDataStructure ::l_str, ayoaron ::data ::NOTYPE, et ayoaron ::data ::STRING.

Voici le graphe d'appel pour cette fonction :



7.21.3.15 void ayoaron ::data ::CData ::setData (std ::string key, irr ::core ::vector3df val)

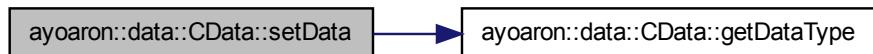
ajoute ou modifie une donnée de type vecteur. Les clés sont uniques à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur vectorielle à affecter à la clé

Références data, getDataType(), ayoaron ::data ::CDataStructure ::l_vec3, ayoaron ::data ::NOTYPE, et ayoaron ::data ::VECTOR.

Voici le graphe d'appel pour cette fonction :



7.21.3.16 void ayoaron ::data ::CData ::setData (std ::string key, irr ::f64 val)

ajoute ou modifie une donnée de type réel. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur réelle à affecter à la clé

Références data, ayoaron ::data ::FLOAT, getDataType(), ayoaron ::data ::CDataStructure ::l_f64, et ayoaron ::data ::NOTYPE.

Voici le graphe d'appel pour cette fonction :



7.21.3.17 void ayoaron ::data ::CData ::setData (std ::string key, irr ::core ::matrix4 val)

ajoute ou modifie une donnée de type matrice 4x4. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur matricielle à affecter à la clé

Références data, getDataType(), ayoaron ::data ::CDataStructure ::l_mat4, ayoaron ::data ::MATRIX, et ayoaron ::data ::NOTYPE.

Voici le graphe d'appel pour cette fonction :



7.21.3.18 void ayoaron ::data ::CData ::setData (std ::string key, irr ::s32 val)

ajoute ou modifie une donnée de type entier signé. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur entière à affecter à la clé

Références data, getDataType(), ayoaron ::data ::CDataStructure ::l_s32, ayoaron ::data ::NOTYPE, et ayoaron ::data ::SINT.

Voici le graphe d'appel pour cette fonction :



7.21.3.19 void ayoaron ::data ::CData ::setData (std ::string key, bool val)

ajoute ou modifie une donnée de type booléen. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur booléenne à affecter à la clé

Références ayoaron ::data ::BOOL, data, getDataType(), ayoaron ::data ::CDataStructure ::l_bool, et ayoaron ::data ::NOTYPE.

Voici le graphe d'appel pour cette fonction :



7.21.3.20 void ayoaron ::data ::CData ::setData (std ::string key, irr ::u32 val)

ajoute ou modifie une donnée de type entier non signé. Les clés sont unique à un type

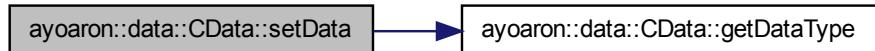
Paramètres

in	key	la clé vers la donnée
in	val	la valeur entière non signée à affecter à la clé

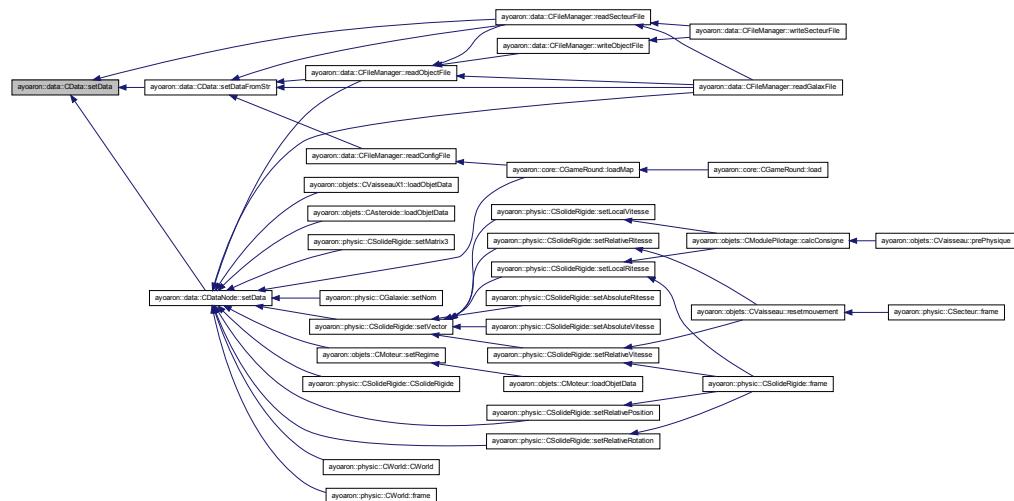
Références data, getDataType(), ayoaron ::data ::CDataStructure ::l_u32, ayoaron ::data ::NOTYPE, et ayoaron ::data ::UINT.

Référencé par ayoaron ::data ::CFileManager ::readSecteurFile(), ayoaron ::data ::CDa-
taNode ::setData(), et setDataFromStr().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.21.3.21 void ayoaron ::data ::CData ::setDataFromStr (std ::string *key*, std ::string *val*, DATATYPE *typ* = NOTYPE)

lit une chaine de caractère et en déduit la donnée à ajouter

Paramètres

<i>key</i>	le nom de la variable à ajouter
<i>val</i>	la chaine à lire
<i>typ</i>	on peut avec cela forcer le type

détermination du type : entier non signé = 0u entier signé = 0s réel = 0.0f vecteur = (0.0,0.0,0.0) matrice = [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0] couleur = {0.0,0.0,0.0,0.0} boolean = true ou false string = "<la chaine>"

on commence par retirer tous les espaces et tabulations : en début et fin de chaine

on stocke en fonction du type

on a un entier non signé on retire les caractère 'u'

on a un entier signé on retire les caractère 's'

on a un float on retire les caractère 'f'

on a un vecteur on retire les caractère '(' et ')'

on a une matrice on retire les caractère '(' et ')'

on a une couleur on retire les caractère '{' et '}'

on a un bool

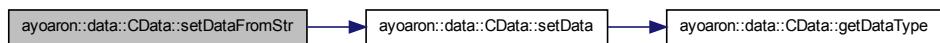
on a une chaine, on retire les guillements

si l'on a rien trouvé, on stocke tout de même la chaîne au cas où

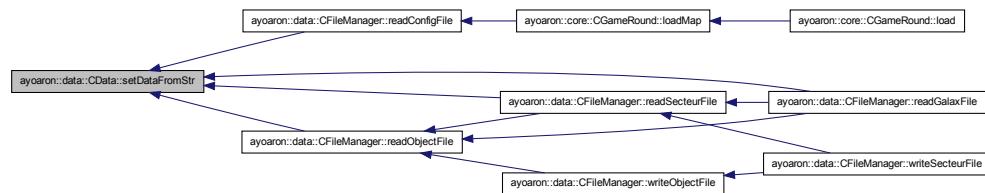
Références ayoaron ::data ::BOOL, ayoaron ::data ::COLOR, ayoaron ::data ::FLOAT, ayoaron ::data ::MATRIX, ayoaron ::data ::NOTYPE, setData(), ayoaron ::data ::SINT, ayoaron ::data ::STRING, ayoaron ::data ::UINT, et ayoaron ::data ::VECTOR.

Référencé par ayoaron ::data ::CFileManager ::readConfigFile(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), et ayoaron ::data ::CFileManager ::readSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.21.4 Documentation des données membres

7.21.4.1 CDataStructure ayoaron ::data ::CData ::data [private]

les données

Référencé par drop(), getbool(), getcol(), getDataList(), getDataType(), getf64(), getmat4(), gets32(), getstr(), getStrFromData(), getu32(), getvec3(), et setData().

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

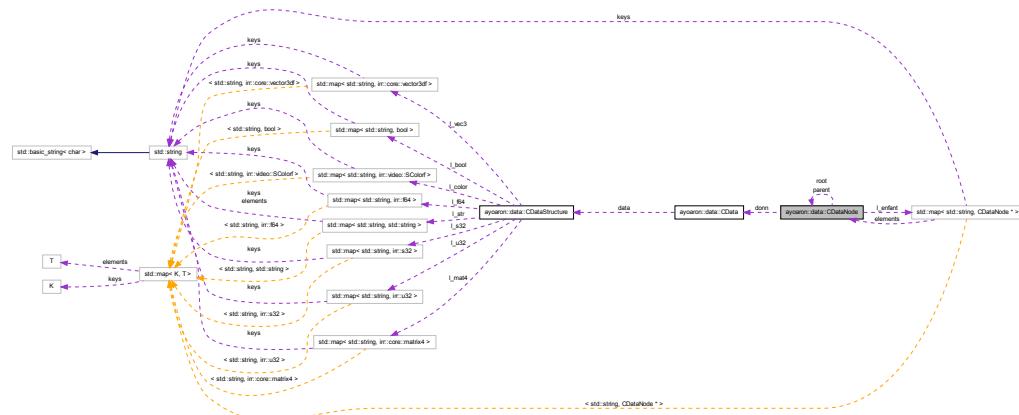
- [CData.h](#)
- [CData.cpp](#)

7.22 Référence de la classe ayoaron ::data ::CDataNode

un bloc de donnée noeud de l'arbre

```
#include <CDataNode.h>
```

Graphe de collaboration de ayoaron : :data : :CDataNode :



Fonctions membres publiques

- void **addEnfant** (std ::string nom, **CDataNode** *node)
 - ajoute un enfant existant à la liste*
 - **CDataNode** (**CDataNode** *p)
 - Constructeur.*
 - **CDataNode** * **createEnfant** (std ::string nom)
 - crée un enfant et renvoie un pointeur sur lui*
 - void **drop** (std ::string key="")
 - detruit toutes les données*
 - **CDataNode** * **dropEnfant** (std ::string nom)
 - supprime un enfant de la liste (mais ne le drop pas)*
 - bool **getbool** (std ::string key)
 - renvoie renvoie la valeur bool correspondant à la clé*
 - irr ::video ::SColor **getcol** (std ::string key)
 - renvoie renvoie la valeur color correspondant à la clé*
 - **CData** * **getData** ()
 - renvoie un pointeur vers les données*
 - std ::list< std ::string > **getDataList** ()
 - renvoie la liste des clés de données*
 - std ::list< std ::string > **getDataListCompared** (**CDataNode** *ref)
 - renvoie une liste des éléments différent d'une référence*
 - **DATATYPE** **getDataType** (std ::string key)
 - renvoie le type de donnée*
 - **CDataNode** * **getEnfant** (std ::string nom)
 - renvoie l'enfant nommé*
 - irr ::f64 **getf64** (std ::string key)
 - renvoie renvoie la valeur float correspondant à la clé*
 - std ::map< std ::string, **CDataNode** * > ::iterator **getIteratorEnfantBegin** ()
 - renvoie un itérateur sur les enfant pointant au début de la map*
 - std ::map< std ::string, **CDataNode** * > ::iterator **getIteratorEnfantEnd** ()
 - renvoie un itérateur sur les enfant pointant à la fin de la map*
 - irr ::core ::matrix4 **getmat4** (std ::string key)
 - renvoie renvoie la valeur matrix correspondant à la clé*
 - irr ::u32 **getNbEnfant** ()
 - renvoie le nombre d'enfant*

- irr : :u32 **getNiveau** ()
 - renvoie le niveau*
- std ::list< std ::string > **getObjetListCompared** (CDataNode *ref)
 - renvoie une liste de sous-objet modifié*
- CDataNode * **getParent** ()
 - renvoie le parent*
- CDataNode * **getRoot** ()
 - renvoie la racine de l'arbre*
- irr : :s32 **getS32** (std ::string key)
 - renvoie renvoie la valeur sint correspondant à la clé*
- std ::string **getStr** (std ::string key)
 - renvoie renvoie la valeur str correspondant à la clé*
- irr : :u32 **getU32** (std ::string key)
 - renvoie renvoie la valeur uint correspondant à la clé*
- irr : :core ::vector3df **getVec3** (std ::string key)
 - renvoie renvoie la valeur vecteur correspondant à la clé*
- void **setData** (std ::string key, irr : :core ::vector3df val)
 - ajoute ou modifie une donnée de type vecteur. Les clés sont unique à un type*
- void **setData** (std ::string key, irr : :u32 val)
 - ajoute ou modifie une donnée de type entier non signé. Les clés sont unique à un type*
- void **setData** (std ::string key, irr : :core ::matrix4 val)
 - ajoute ou modifie une donnée de type matrice 4x4. Les clés sont unique à un type*
- void **setData** (std ::string key, irr : :f64 val)
 - ajoute ou modifie une donnée de type réel. Les clés sont unique à un type*
- void **setData** (std ::string key, irr : :video ::SColorf val)
 - ajoute ou modifie une donnée de type couleur. Les clés sont unique à un type*
- void **setData** (std ::string key, bool val)
 - ajoute ou modifie une donnée de type booléen. Les clés sont unique à un type*
- void **setData** (std ::string key, irr : :s32 val)
 - ajoute ou modifie une donnée de type entier signé. Les clés sont unique à un type*
- void **setData** (std ::string key, std ::string val)
 - ajoute ou modifie une donnée de type chaîne. Les clés sont unique à un type*
- void **setParent** (CDataNode *p)
 - définit la structure de donnée parente*
- virtual ~**CDataNode** ()
 - Destructeur.*

Attributs privés

- CData * **donn**
 - les données*
- std ::map< std ::string, CDataNode * > **l_enfant**
 - liste des enfants*
- irr : :u32 **niveau**
 - le niveau dans l'arbre*
- CDataNode * **parent**
 - pointeur vers le parent*
- CDataNode * **root**
 - pointeur vers la racine de l'arbre*

7.22.1 Description détaillée

un bloc de donnée noeud de l'arbre

7.22.2 Documentation des constructeurs et destructeur

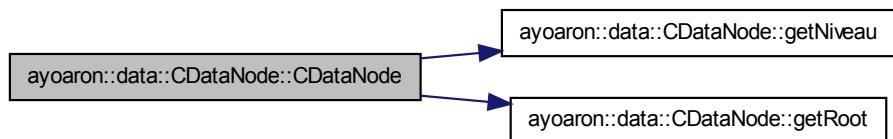
7.22.2.1 ayoaron ::data ::CDataNode ::CDataNode (**CDataNode * p**)

Constructeur.

Références donn, getNiveau(), getRoot(), niveau, parent, et root.

Référencé par createEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.2.2 ayoaron ::data ::CDataNode ::~CDataNode () [virtual]

Destructeur.

Références donn, ayoaron ::data ::CData ::getstr(), et l_enfant.

Voici le graphe d'appel pour cette fonction :



7.22.3 Documentation des fonctions membres

7.22.3.1 void ayoaron ::data ::CDataNode ::addEnfant (std ::string *nom*, CDataNode * *node*)

ajoute un enfant exsistant à la liste

Paramètres

<i>nom</i>	nom unique de l'enfant dans la liste
<i>node</i>	pointeur vers l'enfant à ajouter

Références l_enfant.

7.22.3.2 CDataNode * ayoaron ::data ::CDataNode ::createEnfant (std ::string *nom*)

crée un enfant et renvoie un pointeur sur lui

Paramètres

in	<i>nom</i>	le nom de l'enfant à créer (si le nom est vide, NULL sera renvoyé)
----	------------	--

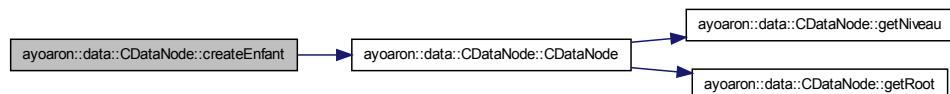
Renvoie

retourne un pointeur vers l'enfant nouvellement créé. Si l'enfant existe déjà, il n'est pas effacé.

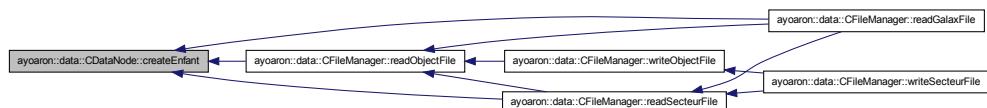
Références CDataNode(), et l_enfant.

Référencé par ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), et ayoaron ::data ::CFileManager ::readSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.3 void ayoaron ::data ::CDataNode ::drop (std ::string *key* = "") [inline]

detruit toue les données

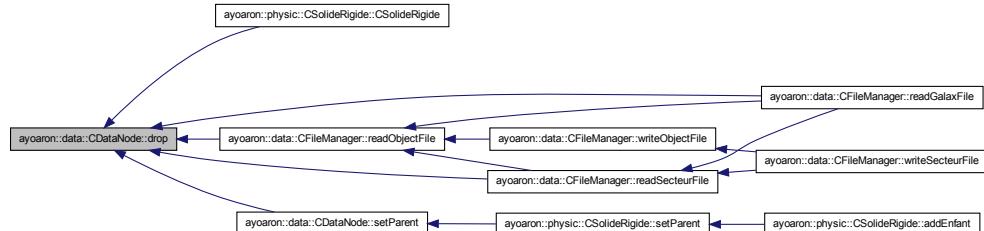
Références donn, et ayoaron ::data ::CData ::drop().

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), ayoaron ::data ::CFileManager ::readSecteurFile(), et setParent().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

7.22.3.4 CDataNode * ayoaron ::data ::CDataNode ::dropEnfant (std ::string *nom*)

supprime un enfant de la liste (mais ne le drop pas)

Paramètres

<i>nom</i>	nom de l'enfant à oublier
------------	---------------------------

Renvoie

pointeur vers cet enfant (pour un eventuel drop)

Références l_enfant.

7.22.3.5 bool ayoaron ::data ::CDataNode ::getbool (std ::string key) [inline]

renvoie renvoie la valeur bool correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur booléenne correspondant à la clé, renvoie false si la clé n'existe pas ou n'est pas du bon type

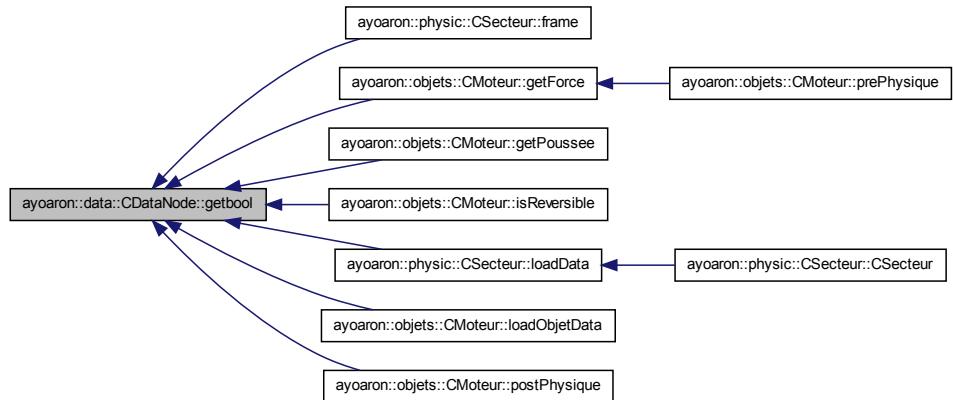
Références donn, et ayoaron ::data ::CData ::getbool().

Référencé par ayoaron ::physic ::CSecteur ::frame(), ayoaron ::objets ::CMoteur ::getForce(), ayoaron ::objets ::CMoteur ::getPoussee(), ayoaron ::objets ::CMoteur ::isReversible(), ayoaron ::physic ::CSecteur ::loadData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.6 `irr ::video ::SColorf ayoaron ::data ::CDataNode ::getcol (std ::string key)`
 [inline]

renvoie renvoie la valeur color correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur de couleur correspondant à la clé, renvoie la couleur noire (0,0,0,0) si la clé n'existe pas ou n'est pas du bon type

Références donn, et ayoaron ::data ::CData ::getcol().

Référencé par ayoaron ::physic ::CSecteur ::loadData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.7 `CData* ayoaron ::data ::CDataNode ::getData ()` [inline]

renvoie un pointeur vers les données

Renvoie

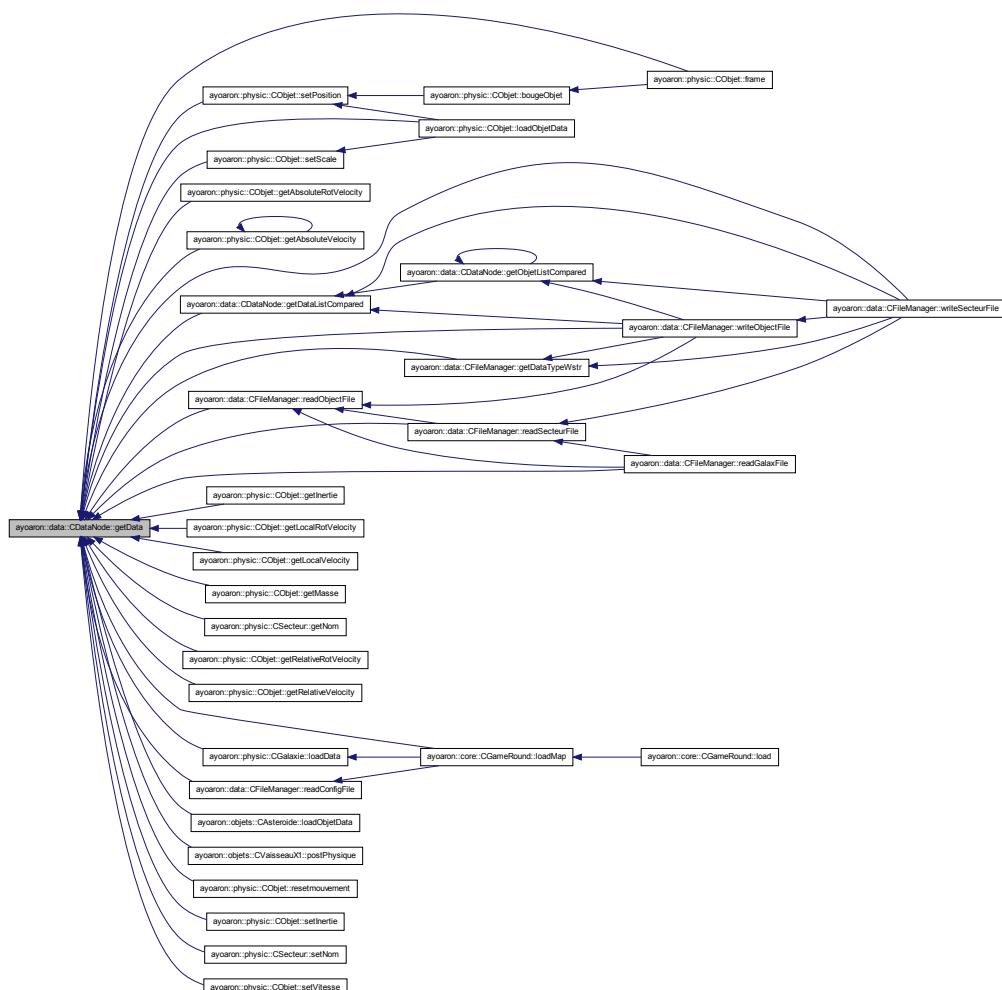
pointeur direct vers la structure de donnée

Références donn.

Référencé par ayoaron ::physic ::CObjet ::frame(), ayoaron ::physic ::CObjet ::getAbsoluteRotVelocity(), ayoaron ::physic ::CObjet ::getAbsoluteVelocity(), getDataListCompared(), ayoaron ::data ::CFileManager ::getDataTypeWstr(), ayoaron ::physic ::CObjet ::getInertie(), ayoaron ::physic ::CObjet ::getLocalRotVelocity(), ayoaron ::physic ::CObjet ::getLocalVelocity(), ayoaron ::physic ::CObjet ::getMasse(),

ayoaron ::physic ::CSecteur ::getNom(), ayoaron ::physic ::CObjet ::getRelativeRotVelocity(), ayoaron ::physic ::CObjet ::getRelativeVelocity(), ayoaron ::physic ::CGalaxie ::loadData(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::physic ::CObjet ::loadObjetData(), ayoaron ::objets ::CAsteroide ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::data ::CFileManager ::readConfigFile(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), ayoaron ::data ::CFileManager ::readSecteurFile(), ayoaron ::physic ::CObjet ::resetmouvement(), ayoaron ::physic ::CObjet ::setInertie(), ayoaron ::physic ::CSecteur ::setNom(), ayoaron ::physic ::CObjet ::setPosition(), ayoaron ::physic ::CObjet ::setScale(), ayoaron ::physic ::CObjet ::setVitesse(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



7.22.3.8 std ::list<std ::string> ayoaron ::data ::CDataNode ::getDataList() [inline]

renvoie la liste des clés de données

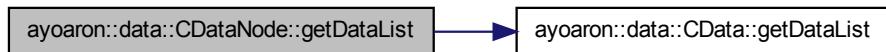
Renvoie

une liste des clés valides dans les données

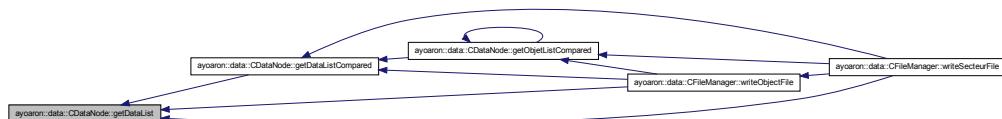
Références donn, et ayoaron ::data ::CData ::getDataList().

Référencé par getDataListCompared(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.9 std ::list< std ::string > ayoaron ::data ::CDataNode ::getDataListCompared (CDataNode * ref)

renvoie une liste des éléments différent d'une référence

Paramètres

in	ref	la structure de donnée de référence
----	-----	-------------------------------------

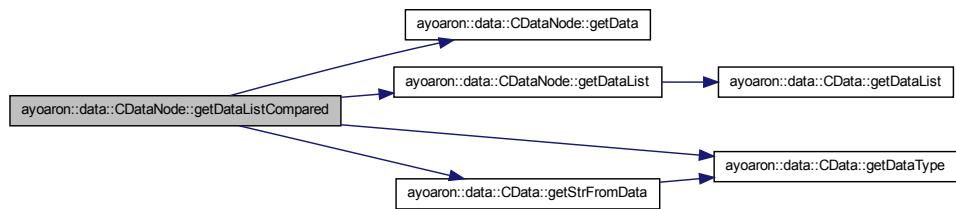
Renvoie

une liste des objets qui sont différents ou en plus d'une référence

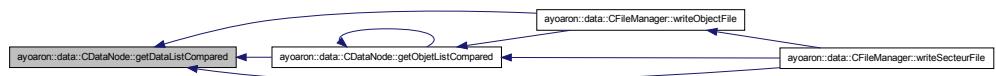
Références donn, getData(), getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::NOTYPE.

Référencé par getObjectListCompared(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.10 DATATYPE ayoaron ::data ::CDataNode ::getDataType (std ::string key) [inline]

revoie le type de donnée

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

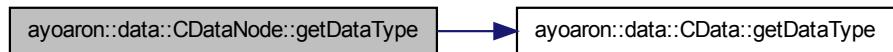
Renvoie

le type de donnée

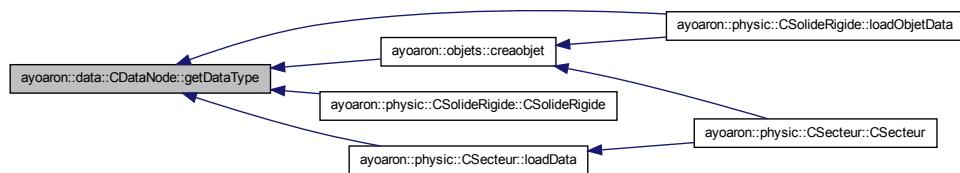
Références donn, et ayoaron ::data ::CData ::getDataType().

Référencé par ayoaron ::objets ::creaobjet(), ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSecteur ::loadData(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.11 CDataNode * ayoaron ::data ::CDataNode ::getEnfant (std ::string nom)

renvoie l'enfant nommé

Paramètres

in	nom	le nom de l'enfant désiré
----	-----	---------------------------

Renvoie

un pointeur vers l'enfant nomé, renvoie NULL si l'enfant n'existe pas

Références l_enfant.

Référencé par getObjetListCompared().

Voici le graphe d'appel pour cette fonction :



7.22.3.12 irr ::f64 ayoaron ::data ::CDataNode ::getf64 (std ::string key) [inline]

renvoie renvoie la valeur float correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

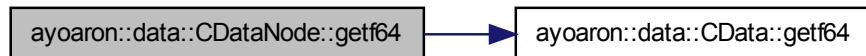
Renvoie

la valeur réelle correspondant à la clé, renvoie 0 si la clé n'existe pas ou n'est pas du bon type

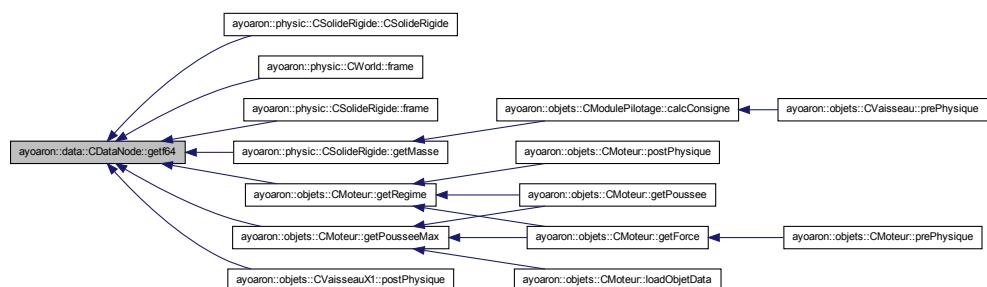
Références donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CWorld ::frame(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getMasse(), ayoaron ::objets ::CMoteur ::getPousseMax(), ayoaron ::objets ::CMoteur ::getRegime(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.13 std ::map< std ::string, CDataNode * > ::iterator ayoaron ::data ::CDataNode ::getIteratorEnfantBegin ()

renvoie un itérator sur les enfant pointant au début de la map

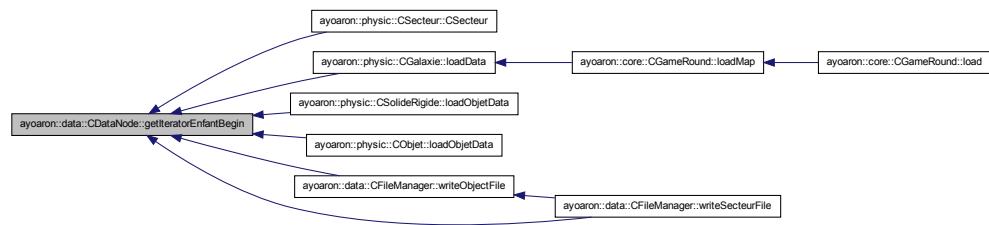
Renvoie

iterator pointant au début de la liste des enfant

Références l_enfant.

Référencé par ayoaron ::physic ::CSecteur ::CSecteur(), ayoaron ::physic ::CGalaxie ::loadData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



7.22.3.14 std ::map< std ::string, CDataNode * > ::iterator ayoaron ::data ::CDataNode ::getIteratorEnfantEnd ()

renvoie un itérator sur les enfant pointant à la fin de la map

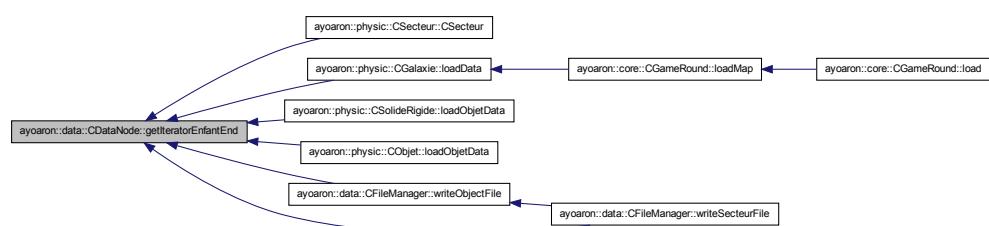
Renvoie

iterator pointant à la fin de la liste des enfant

Références l_enfant.

Référencé par ayoaron ::physic ::CSecteur ::CSecteur(), ayoaron ::physic ::CGalaxie ::loadData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



7.22.3.15 irr ::core ::matrix4 ayoaron ::data ::CDataNode ::getmat4 (std ::string *key*)
 [inline]

renvoie renvoie la valeur matrix correspondant à la clé

Paramètres

in	<i>key</i>	la clé vers la donnée
----	------------	-----------------------

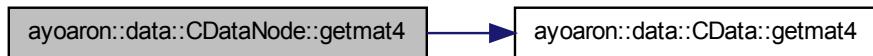
Renvoie

la valeur matricielle correspondant à la clé, renvoie la matrice nulle si la clé n'existe pas ou n'est pas du bon type

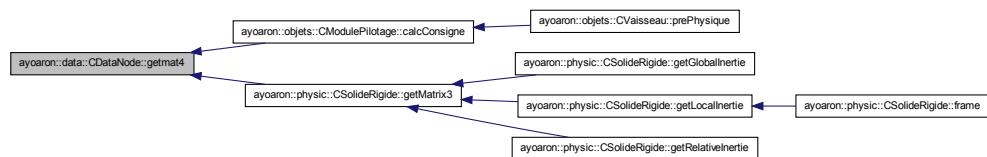
Références donn, et ayoaron ::data ::CData ::getmat4().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::getMatrix3().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.16 irr ::u32 ayoaron ::data ::CDataNode ::getNbEnfant () [inline]

renvoie le nombre d'enfant

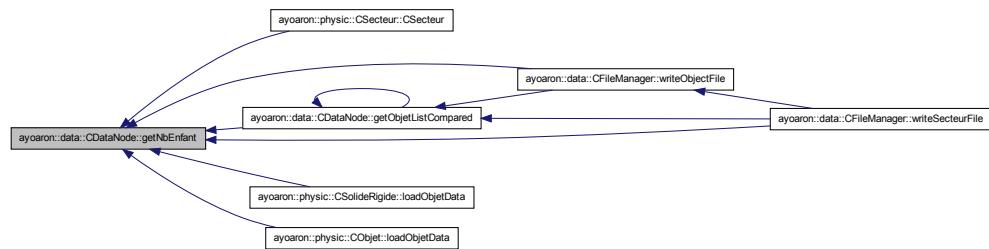
Renvoie

renvoie le nombre de sous-structures enfant

Références l_enfant.

Référencé par ayoaron ::physic ::CSecteur ::CSecteur(), getObjetListCompared(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), ayoaron ::data ::CFileManager ::writeObjectFile(), et ayoaron ::data ::CFileManager ::writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



7.22.3.17 irr ::u32 ayoaron ::data ::CDataNode ::getNiveau() [inline]

renvoie le niveau

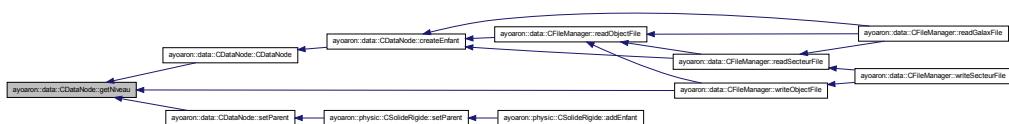
Renvoie

le niveau dans l'arbre de données (0= racine)

Références niveau.

Référencé par CDataNode(), setParent(), et ayoaron ::data ::CFileManager ::writeObjectFile().

Voici le graphe d'appel pour cette fonction :



7.22.3.18 std ::list< std ::string > ayoaron ::data ::CDataNode ::getObjetListCompared (CDataNode * ref)

renvoie une liste de sous-objet modifié

Paramètres

in	ref	la structure de donnée de référence
----	-----	-------------------------------------

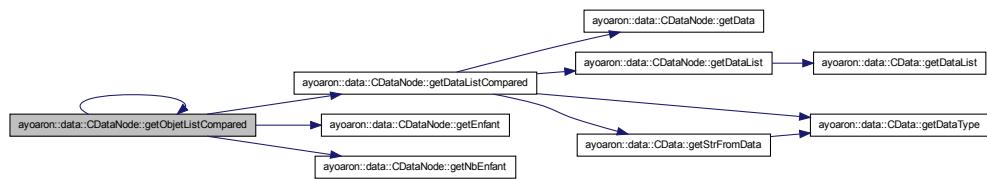
Renvoie

une liste des sous-structure différente des sous-structures de la référence

Références `getDataListCompared()`, `getEnfant()`, `getNbEnfant()`, `getObjetListCompared()`, et `l_enfant`.

Référencé par `getObjetListCompared()`, `ayoaron ::data ::CFileManager ::writeObjectFile()`, et `ayoaron ::data ::CFileManager ::writeSecteurFile()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

**7.22.3.19 CDataNode* ayoaron ::data ::CDataNode ::getParent() [inline]**

renvoie le parent

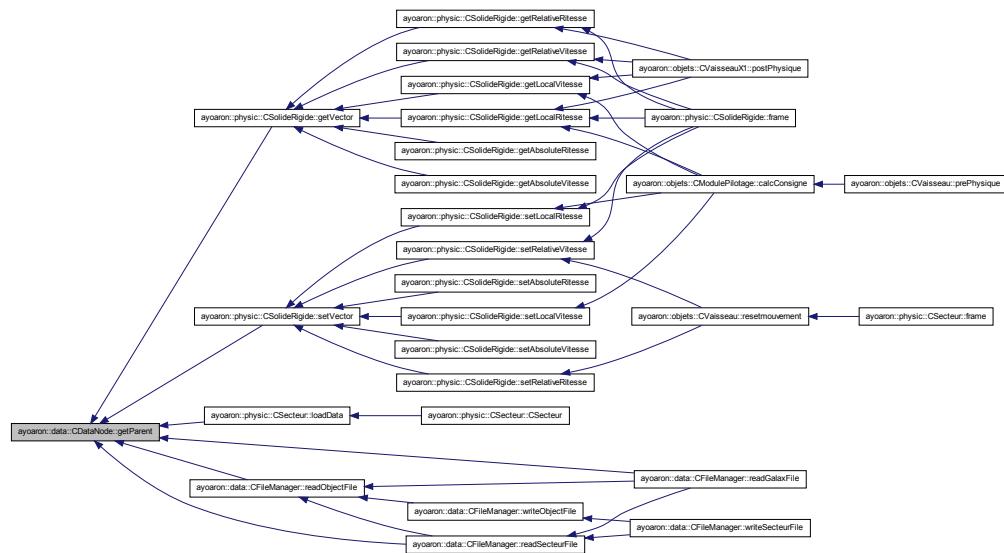
Renvoie

pointeur vers la structure parente, renvoie NULL si la structure est la racine

Références parent.

Référencé par `ayoaron ::physic ::CSolideRigide ::getVector()`, `ayoaron ::physic ::CSecteur ::loadData()`, `ayoaron ::data ::CFileManager ::readGalaxFile()`, `ayoaron ::data ::CFileManager ::readObjectFile()`, `ayoaron ::data ::CFileManager ::readSecteurFile()`, et `ayoaron ::physic ::CSolideRigide ::setVector()`.

Voici le graphe d'appel pour cette fonction :



7.22.3.20 CDataNode* ayoaron ::data ::CDataNode ::getRoot () [inline]

revoie la racine de l'arbre

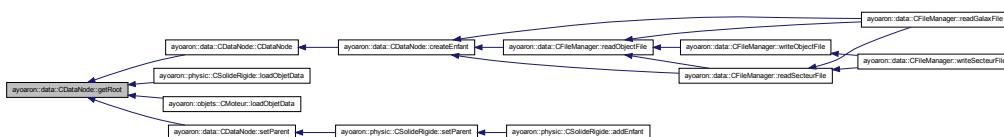
Renvoie

pointeur vers la structure racine, renvoi this si le node est déjà la racine !!!

Références root.

Référencé par CDataNode(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et setParent().

Voici le graphe d'appel pour cette fonction :



7.22.3.21 irr ::s32 ayoaron ::data ::CDataNode ::getS32 (std ::string key) [inline]

renvoie renvoie la valeur sint correspondant à la clé

Paramètres

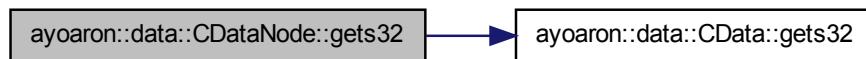
in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur entière signée correspondant à la clé, renvoie 0 si la clé n'existe pas ou n'est pas du bon type

Références donn, et ayoaron ::data ::CData ::gets32().

Voici le graphe d'appel pour cette fonction :

**7.22.3.22 std ::string ayoaron ::data ::CDataNode ::getstr (std ::string key) [inline]**

renvoie renvoie la valeur str correspondant à la clé

Paramètres

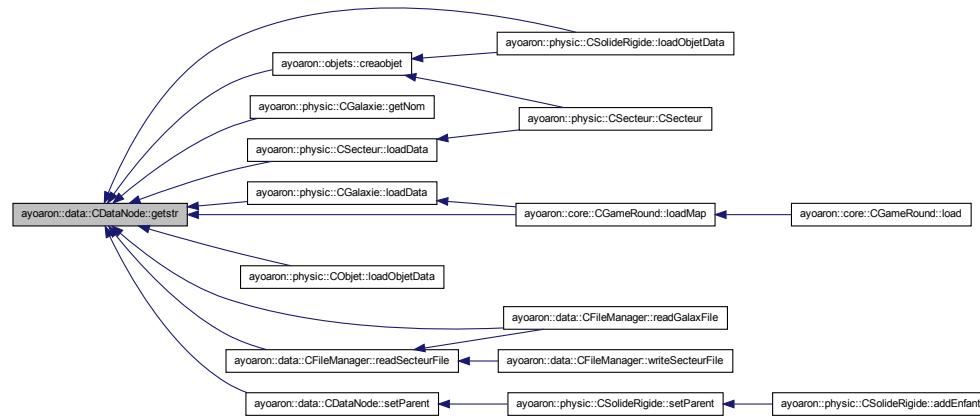
in	key	la clé vers la donnée
----	-----	-----------------------

Renvoie

la valeur chaîne correspondant à la clé, renvoie une chaîne vide si la clé n'existe pas ou n'est pas du bon type

Référencé par ayoaron ::objets ::creaobjet(), ayoaron ::physic ::CGalaxie ::getNom(), ayoaron ::physic ::CSecteur ::loadData(), ayoaron ::physic ::CGalaxie ::loadData(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readSecteurFile(), et setParent().

Voici le graphe d'appel pour cette fonction :



7.22.3.23 irr ::u32 ayoaron ::data ::CDataNode ::getu32 (std ::string key) [inline]

renvoie renvoie la valeur uint correspondant à la clé

Paramètres

in	key	la clé vers la donnée
----	-----	-----------------------

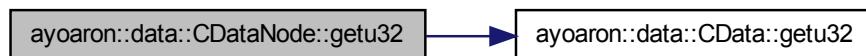
Renvoie

la valeur entière non signée correspondant à la clé, renvoie 0 si la clé n'existe pas ou n'est pas du bon type

Références donn, et ayoaron ::data ::CData ::getu32().

Référencé par ayoaron ::physic ::CSecteur ::loadData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.24 `irr ::core ::vector3df ayoaron ::data ::CDataNode ::getvec3(std ::string key)`
[inline]

renvoie renvoie la valeur vecteur correspondant à la clé

Paramètres

in	<code>key</code>	la clé vers la donnée
----	------------------	-----------------------

Renvoie

la valeur vectorielle correspondant à la clé, renvoie (0,0,0) si la clé n'existe pas ou n'est pas du bon type

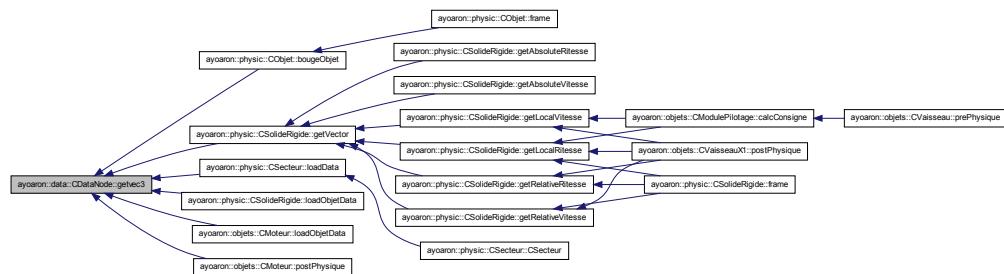
Références donn, et ayoaron ::data ::CData ::getvec3().

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::CSecteur ::loadData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.22.3.25 void ayoaron ::data ::CDataNode ::setData (std ::string *key*, irr ::u32 *val*)
[inline]**

ajoute ou modifie une donnée de type entier non signé. Les clés sont unique à un type

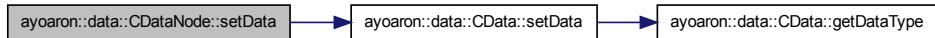
Paramètres

in	<i>key</i>	la clé vers la donnée
in	<i>val</i>	la valeur entière non signée à affecter à la clé

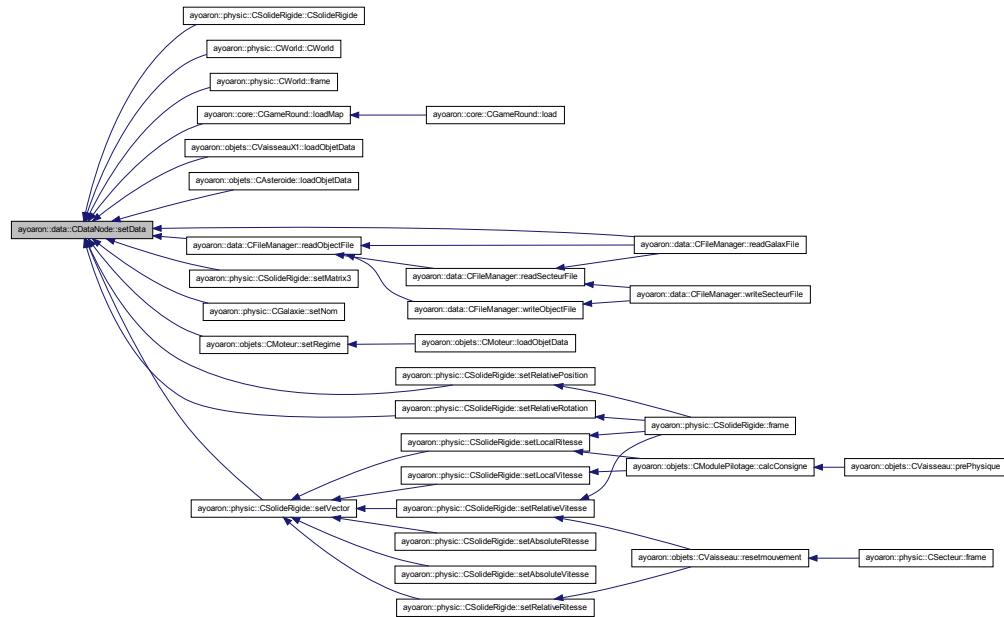
Références donn, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CWorld ::CWorld(), ayoaron ::physic ::CWorld ::frame(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::objets ::CAsteroid ::loadObjetData(), ayoaron ::data ::CFileManager ::readGalaxFile(), ayoaron ::data ::CFileManager ::readObjectFile(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), ayoaron ::physic ::CGalaxie ::setNom(), ayoaron ::objets ::CMoteur ::setRegime(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.3.26 void ayoaron ::data ::CDataNode ::setData (std ::string key, irr ::video ::SColorf val) [inline]

ajoute ou modifie une donnée de type couleur. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur de couleur à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.27 void ayoaron ::data ::CDataNode ::setData (std ::string *key*, irr ::s32 *val*)
 [inline]

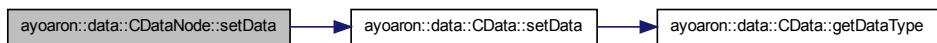
ajoute ou modifie une donnée de type entier signé. Les clés sont unique à un type

Paramètres

in	<i>key</i>	la clé vers la donnée
in	<i>val</i>	la valeur entière à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.28 void ayoaron ::data ::CDataNode ::setData (std ::string *key*, bool *val*)
 [inline]

ajoute ou modifie une donnée de type booléen. Les clés sont unique à un type

Paramètres

in	<i>key</i>	la clé vers la donnée
in	<i>val</i>	la valeur booléenne à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.29 void ayoaron ::data ::CDataNode ::setData (std ::string *key*, std ::string *val*)
 [inline]

ajoute ou modifie une donnée de type chaîne. Les clés sont unique à un type

Paramètres

in	<i>key</i>	la clé vers la donnée
in	<i>val</i>	la valeur chaîne à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.30 void ayoaron ::data ::CDataNode ::setData (std ::string key, irr ::core ::matrix4 val) [inline]

ajoute ou modifie une donnée de type matrice 4x4. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur matricielle à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.31 void ayoaron ::data ::CDataNode ::setData (std ::string key, irr ::f64 val) [inline]

ajoute ou modifie une donnée de type réel. Les clés sont unique à un type

Paramètres

in	key	la clé vers la donnée
in	val	la valeur réelle à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.32 void ayoaron ::data ::CDataNode ::setData (std ::string *key*, irr ::core ::vector3df *val*) [inline]

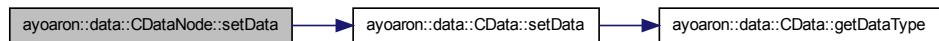
ajoute ou modifie une donnée de type vecteur. Les clés sont unique à un type

Paramètres

in	<i>key</i>	la clé vers la donnée
in	<i>val</i>	la valeur vectorielle à affecter à la clé

Références donn, et ayoaron ::data ::CData ::setData().

Voici le graphe d'appel pour cette fonction :



7.22.3.33 void ayoaron ::data ::CDataNode ::setParent (CDataNode * *p*)

définit la structure de donnée parente

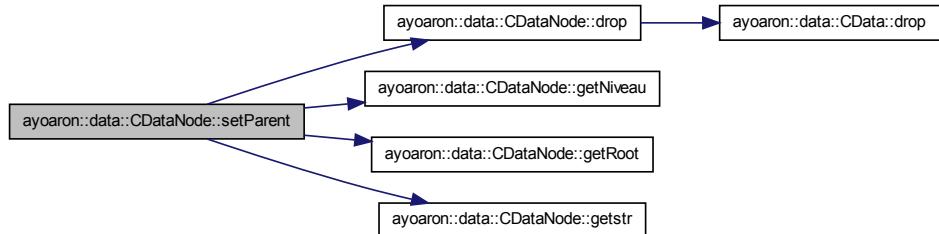
Paramètres

in	<i>p</i>	le nouveau parent
----	----------	-------------------

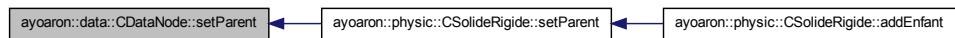
Références drop(), getNiveau(), getRoot(), getstr(), niveau, parent, et root.

Référencé par ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.22.4 Documentation des données membres

7.22.4.1 CData* ayoaron ::data ::CDataNode ::donn [private]

les données

Référencé par CDataNode(), drop(), getbool(), getcol(), getData(), getDataList(), getDataListCompared(), getDataType(), getf64(), getmat4(), gets32(), getu32(), getvec3(), setData(), et ~CDataNode().

7.22.4.2 std ::map<std ::string,CDataNode*> ayoaron ::data ::CDataNode ::l_- enfant [private]

liste des enfants

Référencé par addEnfant(), createEnfant(), dropEnfant(), getEnfant(), getIteratorEnfantBegin(), getIteratorEnfantEnd(), getNbEnfant(), getObjetListCompared(), et ~CDataNode().

7.22.4.3 irr ::u32 ayoaron ::data ::CDataNode ::niveau [private]

le niveau dans l'arbre

Référencé par CDataNode(), getNiveau(), et setParent().

7.22.4.4 CDataNode* ayoaron ::data ::CDataNode ::parent [private]

pointeur vers le parent

Référencé par CDataNode(), getParent(), et setParent().

7.22.4.5 CDataNode* ayoaron ::data ::CDataNode ::root [private]

pointeur vers la racine de l'arbre

Référencé par CDataNode(), getRoot(), et setParent().

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

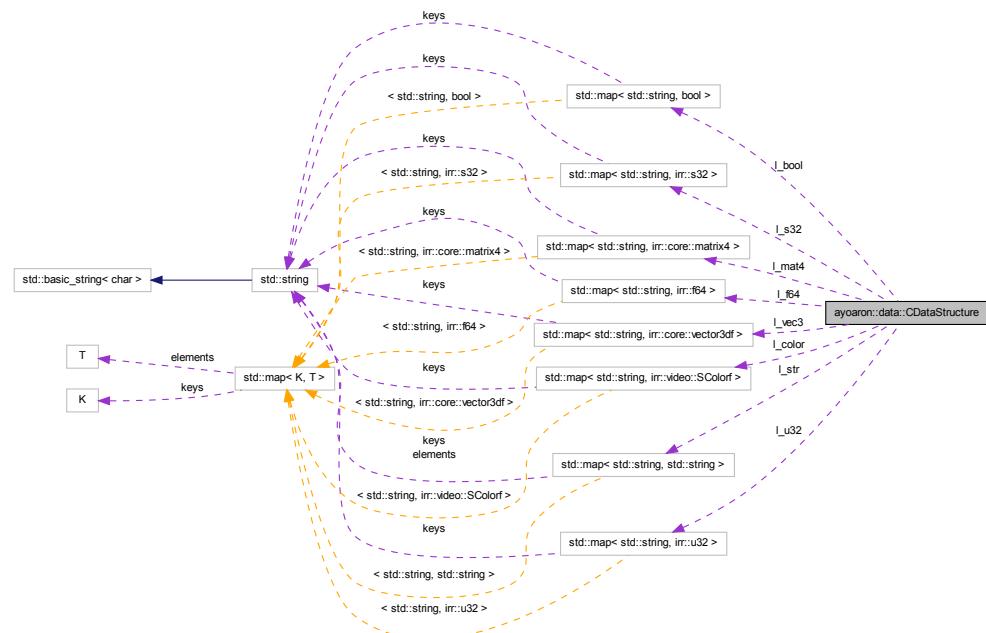
- [CDataNode.h](#)
- [CDataNode.cpp](#)

7.23 Référence de la structure ayoaron ::data ::CDataStructure

rassemblement des structures de données

```
#include <CData.h>
```

Graphe de collaboration de ayoaron ::data ::CDataStructure :



Attributs publics

- std::map<std::string, bool> [l_bool](#)
liste d'élément de type booléen
- std::map<std::string, irr::video::SColorf> [l_color](#)
liste d'élément de type couleur
- std::map<std::string, irr::f64> [l_f64](#)

- liste d'élément de type réel 64bits*
 - std ::map< std ::string, irr ::core ::matrix4 > [l_mat4](#)
- liste d'élément de type matrice 4x4*
 - std ::map< std ::string, irr ::s32 > [l_s32](#)
- liste d'élément de type entier signés 32bits*
 - std ::map< std ::string, std ::string > [l_str](#)
- liste d'élément de type chaîne de caractère*
 - std ::map< std ::string, irr ::u32 > [l_u32](#)
- liste d'élément de type entier non signés 32bits*
 - std ::map< std ::string, irr ::core ::vector3df > [l_vec3](#)
- liste d'élément de type vecteur 3d*

7.23.1 Description détaillée

rassemblement des structures de données

7.23.2 Documentation des données membres

7.23.2.1 std ::map<std ::string,bool> ayoaron ::data ::CDataStructure ::l_bool

liste d'élément de type booléen

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getbool(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.2 std ::map<std ::string,irr ::video ::SColorf> ayoaron ::data ::CDataStructure ::l_color

liste d'élément de type couleur

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getcol(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.3 std ::map<std ::string,irr ::f64> ayoaron ::data ::CDataStructure ::l_f64

liste d'élément de type réel 64bits

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getf64(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.4 std ::map<std ::string,irr ::core ::matrix4> ayoaron ::data ::CDataStructure ::l_mat4

liste d'élément de type matrice 4x4

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getmat4(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.5 std ::map<std ::string,irr ::s32> ayoaron ::data ::CDataStructure ::l_s32

liste d'élément de type entier signés 32bits

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::gets32(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.6 std ::map<std ::string,std ::string> ayoaron ::data ::CDataStructure ::l_str

liste d'élément de type chaîne de caractère

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getstr(), ayoaron ::data ::CData ::getStrFromData(), et ayoaron ::data ::CData ::setData().

7.23.2.7 std ::map<std ::string,irr ::u32> ayoaron ::data ::CDataStructure ::l_u32

liste d'élément de type entier non signés 32bits

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getStrFromData(), ayoaron ::data ::CData ::getu32(), et ayoaron ::data ::CData ::setData().

**7.23.2.8 std ::map<std ::string,irr ::core ::vector3df>
ayoaron ::data ::CDataStructure ::l_vec3**

liste d'élément de type vecteur 3d

Référencé par ayoaron ::data ::CData ::drop(), ayoaron ::data ::CData ::getDataList(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::CData ::getStrFromData(), ayoaron ::data ::CData ::getvec3(), et ayoaron ::data ::CData ::setData().

La documentation de cette structure a été générée à partir du fichier suivant :

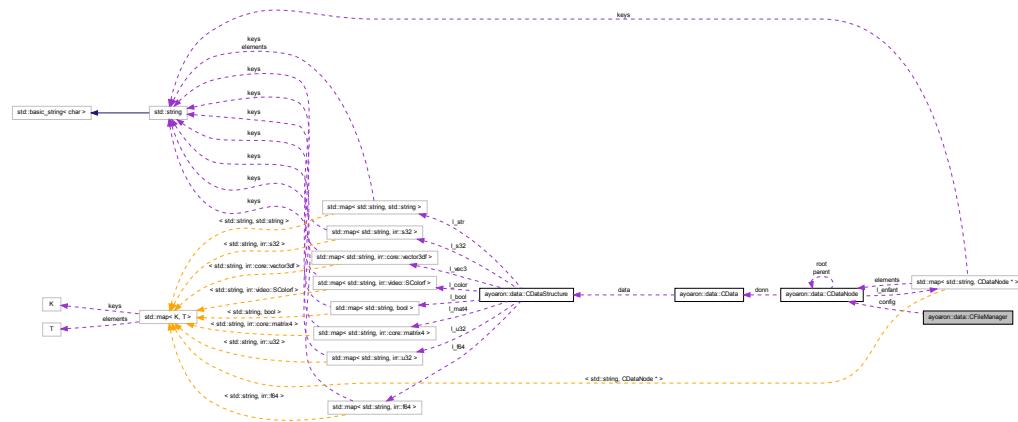
– [CData.h](#)

7.24 Référence de la classe ayoaron ::data ::CFileManager

classe de gestion des fichiers XML notamment

```
#include <CFileManager.h>
```

Graphe de collaboration de ayoaron : :data : :CFileManager :



Fonctions membres publiques

- **CFileManager** (irr : :io : :IFileSystem *fs)
Constructeur.
 - **CDataNode * getConfig ()**
renvoie un pointeur vers les données de configuration
 - **void readConfigFile (std : :string filename)**
lecture d'un fichier de config
 - **void readGalaxFile (std : :string filename, CDataNode *root)**
lit un fichier de galaxie
 - **void readObjectFile (std : :string filename, CDataNode *ici, irr : :io : :IXMLReader *xxmlr=NULL)**
lit un fichier objet
 - **void readSecteurFile (std : :string filename, CDataNode *ici, irr : :io : :IXMLReader *xmlww=NULL)**
lit un fichier de secteur
 - **void writeGalaxFile (std : :string filename, CDataNode *root)**
écrit un fichier de galaxie.
 - **void writeObjectFile (std : :string filename, CDataNode *ici, irr : :io : :IXMLWriter *xmlww=NULL)**
écrit un fichier objet.
 - **void writeSavegame (std : :string filename, CDataNode *root, std : :string reference-file)**
écrit une savegame (i.e. un fichier de galaxie unique avec tous les paramètres différent de la référence)
 - **void writeSecteurFile (std : :string filename, CDataNode *ici, irr : :io : :IXMLWriter *xmlww=NULL)**
écrit un fichier de secteur.
 - **virtual ~CFileManager ()**
Destructeur.

Fonctions membres privées

- std::wstring **getDataTypeWstr** (std::string key, CDataNode *ici)

renvoie une wstring de type

Attributs privés

- **CDataNode * config**
données de configuration
- **irr ::io ::IFileSystem * filesys**
pointeur vers le système de fichier

7.24.1 Description détaillée

classe de gestion des fichiers XML notamment

7.24.2 Documentation des constructeurs et destructeur

7.24.2.1 ayoaron ::data ::CFileManager ::CFileManager (irr ::io ::IFileSystem * fs)

Constructeur.

Références config, et filesys.

7.24.2.2 ayoaron ::data ::CFileManager ::~CFileManager () [virtual]

Destructeur.

Références config.

7.24.3 Documentation des fonctions membres

7.24.3.1 CDataNode* ayoaron ::data ::CFileManager ::getConfig () [inline]

renvoie un pointeur vers les données de configuration

Références config.

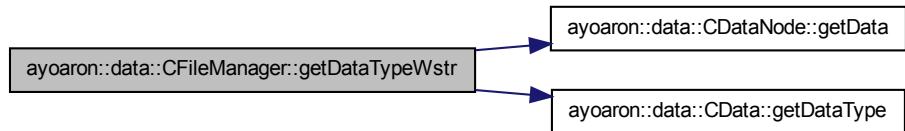
7.24.3.2 std ::wstring ayoaron ::data ::CFileManager ::getDataTypeWstr (std ::string key, CDataNode * ici) [private]

renvoie une wstring de type

Références ayoaron ::data ::BOOL, ayoaron ::data ::COLOR, ayoaron ::data ::FLOAT, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CData ::getDataType(), ayoaron ::data ::MATRIX, ayoaron ::data ::SINT, ayoaron ::data ::STRING, ayoaron ::data ::UINT, et ayoaron ::data ::VECTOR.

Référencé par writeObjectFile(), et writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.24.3.3 void ayoaron ::data ::CFileManager ::readConfigFile (std ::string *filename*)

lecture d'un fichier de config

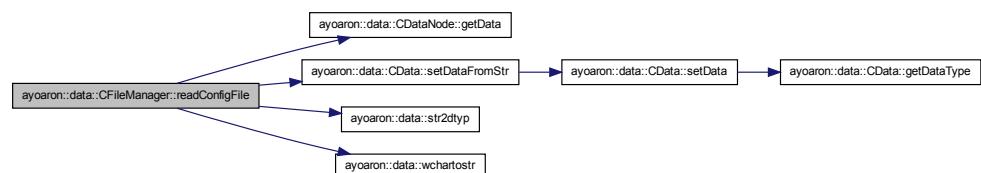
À faire

ajouter d'autres catégories

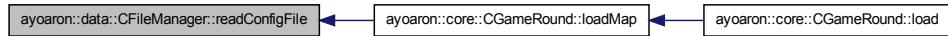
Références config, filesys, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CData ::setDataFromStr(), ayoaron ::data ::str2dtyp(), et ayoaron ::data ::wchartostr().

Référencé par ayoaron ::core ::CGameRound ::loadMap().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

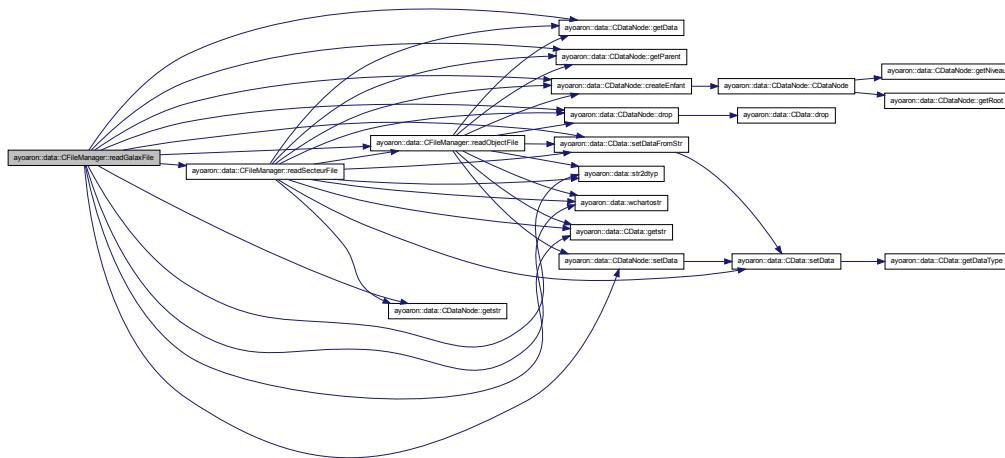


7.24.3.4 void ayoaron ::data ::CFileManager ::readGalaxyFile (std ::string *filename*, CDataNode * *root*)

lit un fichier de galaxie

Références config, ayoaron ::data ::CDataNode ::createEnfant(), ayoaron ::data ::CDataNode ::drop(), filesystem, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::data ::CDataNode ::getStr(), ayoaron ::data ::CDataNode ::getStr(), readObjectFile(), readSecteurFile(), ayoaron ::data ::CDataNode ::setData(), ayoaron ::data ::CDataNode ::setDataFromStr(), ayoaron ::data ::str2dtyp(), et ayoaron ::data ::wchartostr().

Voici le graphe d'appel pour cette fonction :



7.24.3.5 void ayoaron ::data ::CFileManager ::readObjectFile (std ::string *filename*, CDataNode * *ici*, irr ::io ::IXMLReader * *xxmlr* = NULL)

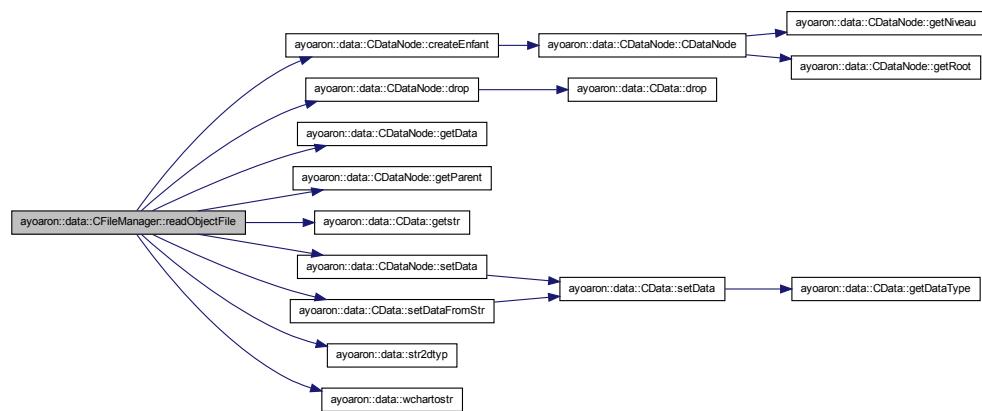
lit un fichier objet

Références config, ayoaron ::data ::CDataNode ::createEnfant(), ayoaron ::data ::CDataNode ::drop(), filesystem, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::data ::CDataNode ::getStr(), ayoaron ::data ::CDataNode ::setData(), ayoaron ::data ::CDataNode ::setDataFromStr(), ayoaron ::data ::str2dtyp(),

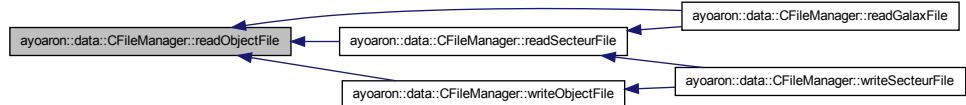
et ayoaron ::data ::wchartostr().

Référencé par readGalaxFile(), readSecteurFile(), et writeObjectFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



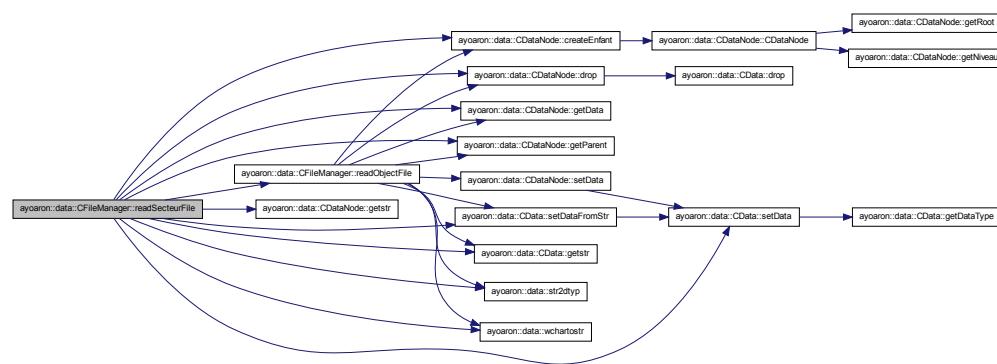
7.24.3.6 void ayoaron ::data ::CFileManager ::readSecteurFile (std ::string *filename*, CDataNode * *ici*, irr ::io ::IXMLReader * *xmlIww* = NULL)

lit un fichier de secteur

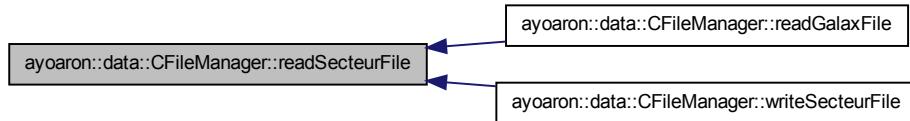
Références config, ayoaron ::data ::CDataNode ::createEnfant(), ayoaron ::data ::CDataNode ::drop(), filesystem, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::data ::CData ::getstr(), ayoaron ::data ::CDataNode ::getstr(), readObjectFile(), ayoaron ::data ::CData ::setData(), ayoaron ::data ::CData ::setDataFromStr(), ayoaron ::data ::str2dtyp(), et ayoaron ::data ::wchartostr().

Référencé par readGalaxFile(), et writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.24.3.7 void ayoaron ::data ::CFileManager ::writeGalaxFile (std ::string *filename*, CDataNode * *root*)

écrit un fichier de galaxie.

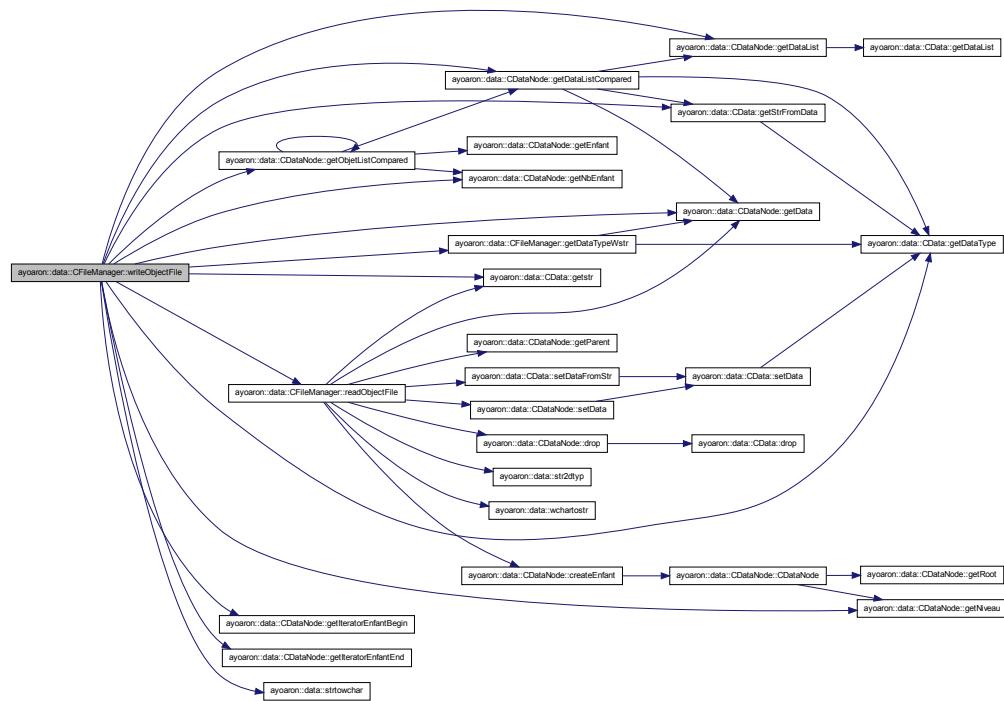
7.24.3.8 void ayoaron : :data : :CFileManager : :writeObjectFile (std : :string *filename*, CDataNode * *ici*, irr : :io : :IXMLWriter * *xmlww* = NULL)

écrit un fichier objet.

Références filesys, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CData-
Node ::getDataList(), ayoaron ::data ::CDataNode ::getDataListCompared(), ayoa-
ron ::data ::CData ::getDataType(), getDataTypeWstr(), ayoaron ::data ::CData-
Node ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::getIteratorEnfantEnd(),
ayoaron ::data ::CDataNode ::getNbEnfant(), ayoaron ::data ::CDataNode ::getNi-
veau(), ayoaron ::data ::CDataNode ::getObjetListCompared(), ayoaron ::data ::CData ::getstr(),
ayoaron ::data ::CData ::getStrFromData(), ayoaron ::data ::NOTYPE, readObject-
File(), et ayoaron ::data ::strtowchar().

Référencé par writeSecteurFile().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.24.3.9 void ayoaron : :data : :CFileManager : :writeSavegame (std : :string *filename*, CDataNode * *root*, std : :string *referencefile*)

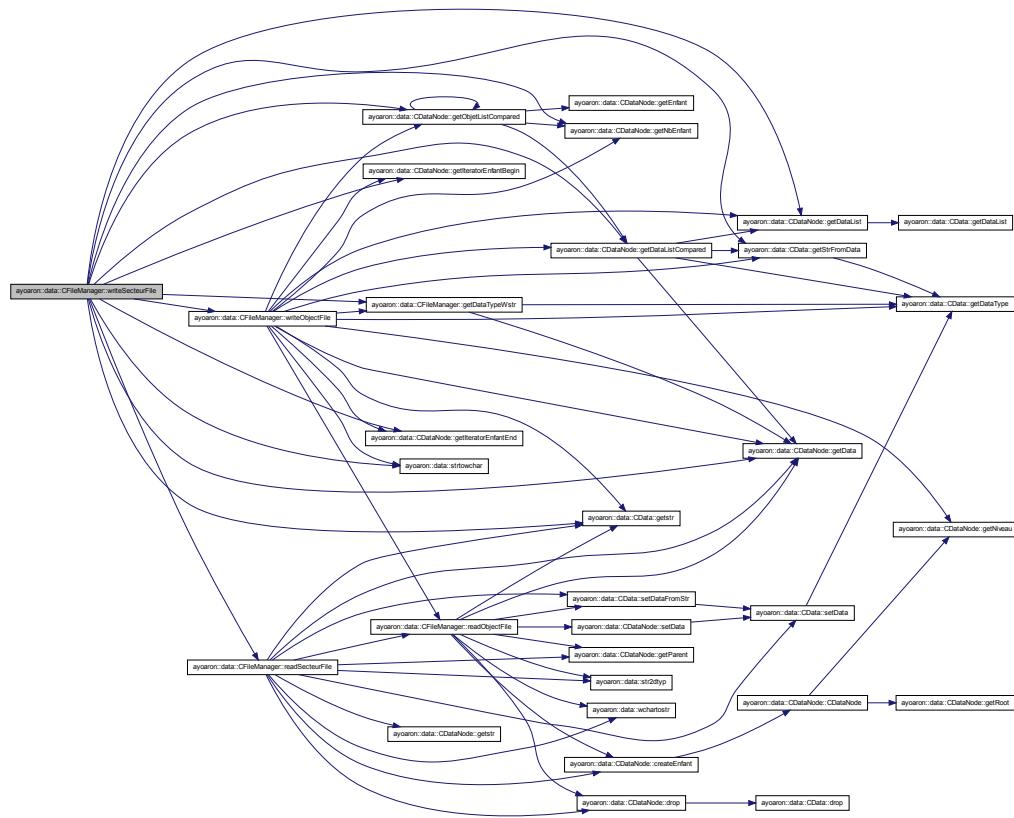
écrit une savegame (i.e. un fichier de galaxie unique avec tous les paramètres différent de la référence)

7.24.3.10 void ayoaron : :data : :CFileManager : :writeSecteurFile (std : :string *filename*, CDataNode * *ici*, irr : :io : :IXMLWriter * *xmlww* = NULL)

écrit un fichier de secteur.

Références config, filesys, ayoaron ::data ::CDataNode ::getData(), ayoaron ::data ::CDataNode ::getDataList(), ayoaron ::data ::CDataNode ::getDataListCompared(), getDataTypeWstr(), ayoaron ::data ::CDataNode ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::getIteratorEnfantEnd(), ayoaron ::data ::CDataNode ::getNbEnfant(), ayoaron ::data ::CDataNode ::getObjectListCompared(), ayoaron ::data ::CData ::getstr(), ayoaron ::data ::CData ::getStrFromData(), readSecteurFile(), ayoaron ::data ::strtowchar(), et writeObjectFile().

Voici le graphe d'appel pour cette fonction :



7.24.4 Documentation des données membres

7.24.4.1 CDataNode* ayoaron ::data ::CFileManager ::config [private]

données de configuration

Référencé par `CFileManager()`, `getConfig()`, `readConfigFile()`, `readGalaxFile()`, `readObjectFile()`, `readSecteurFile()`, `writeSecteurFile()`, et `~CFileManager()`.

7.24.4.2 irr ::io ::IFileSystem* ayoaron ::data ::CFileManager ::filesys [private]

pointeur vers le système de fichier

Référencé par CFileManager(), readConfigFile(), readGalaxFile(), readObjectFile(), readSecteurFile(), writeObjectFile(), et writeSecteurFile().

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

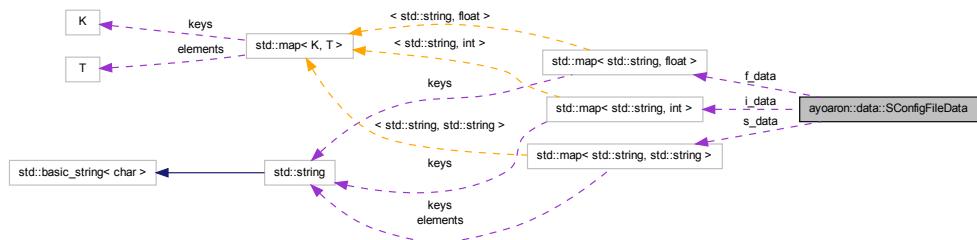
- [CFileManager.h](#)
- [CFileManager.cpp](#)

7.25 Référence de la structure ayoaron ::data ::SConfigFileData

conteneur de donnée fichier config

```
#include <CConfigFile.h>
```

Graphe de collaboration de ayoaron ::data ::SConfigFileData :



Attributs publics

- std ::map< std ::string, float > [f_data](#)
données de type float
- std ::map< std ::string, int > [i_data](#)
données de type entier
- std ::map< std ::string, std ::string > [s_data](#)
données de type chaîne

7.25.1 Description détaillée

conteneur de donnée fichier config

7.25.2 Documentation des données membres

7.25.2.1 std ::map<std ::string, float> ayoaron ::data ::SConfigFileData ::f_data

données de type float

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::data ::CConfigFile :::isIn(), et ayoaron ::data ::CConfigFile :::parse().

7.25.2.2 std ::map<std ::string,int> ayoaron ::data ::SConfigFileData ::i_data

données de type entier

Référencé par ayoaron ::reseau ::CNetworkEngine ::getConfigData(), ayoaron ::graphic ::CGraphicsEngine ::getConfigData(), ayoaron ::data ::CConfigFile :::isIn(), et ayoaron ::data ::CConfigFile :::parse().

7.25.2.3 std ::map<std ::string,std ::string> ayoaron ::data ::SConfigFileData ::s_data

données de type chaîne

Référencé par ayoaron ::data ::CConfigFile :::isIn(), et ayoaron ::data ::CConfigFile :::parse().

La documentation de cette structure a été générée à partir du fichier suivant :

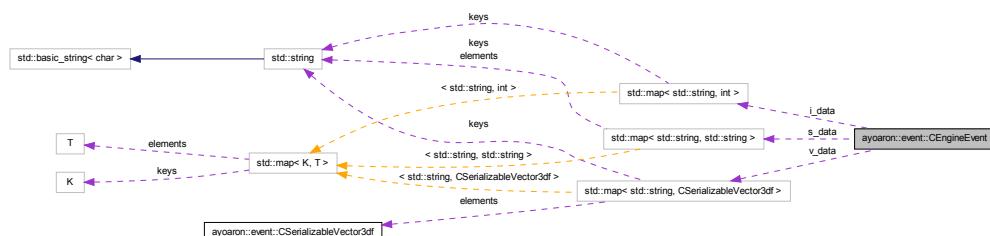
– [CConfigFile.h](#)

7.26 Référence de la classe ayoaron ::event ::CEngineEvent

classe générique pour manipuler les événements entre moteurs

```
#include <CEngineEvent.h>
```

Graphe de collaboration de ayoaron ::event ::CEngineEvent :



Types publics

```
- enum {
  MSG_NICK, GAME_PROP, WELCOME, REFUSED,
  NEW_PLAYER, RUN_MULTI, SEND_GAME_PROP, GAME_HAS_STARTED,
  CHOSEN_MAP, DEL_PLAYER, LOAD, ADD_NODE,
  REM_NODE, MOVE_NODE, REMOVE_MACHINE }
```

liste des types d'événements list of all graphics_events : type key value

Fonctions membres publiques

- bool **operator==** (const **CEngineEvent** &e)
opérateur de comparaison
- template<class Archive >
void serialize (Archive &ar, const unsigned int)
sérialise les données

Attributs publics

- std ::map< std ::string, int > **i_data**
donnée de type entier
- std ::map< std ::string, std ::string > **s_data**
données de type chaîne
- int **type**
type d'événement
- std ::map< std ::string, **CSerializableVector3df** > **v_data**
donnée de type liste de vecteur

7.26.1 Description détaillée

classe générique pour manipuler les événements entre moteurs

7.26.2 Documentation des énumérations membres

7.26.2.1 anonymous enum

liste des types d'événements list of all graphics_events : type key value

```
///////////////////////////// list of all network_events : type key value MSG_-  

NICK NICK player's nick PASS password to join the game GAME_PROP MAP map  

name NAME name of the server NB_PLAY max number of players WELCOME none  

REFUSED REASON explanation (nick already existing, wrong password, game full  

...) NEW_PLAYER NAME name of the player RUN_MULTI none SEND_GAME_PROP  

none GAME_HAS_STARTED none informs the network engine that the game has star-  

ted CHOSEN_MAP NAME name of the chosen map DEL_PLAYER NAME name of the  

player to remove LOAD none used for the game round to do the graphics engine load  

the round ADD_NODE ID id of the mesh to add POS position to put the node at ROT  

orientation to put the node at MOVE_NODE ID POS ROT REM_NODE ID id of the node  

to delete REMOVE_MACHINE ID id of the network machine to remove
```

Valeurs énumérées :

MSG_NICK
GAME_PROP
WELCOME
REFUSED
NEW_PLAYER
RUN_MULTI
SEND_GAME_PROP

GAME_HAS_STARTED
CHOSEN_MAP
DEL_PLAYER
LOAD
ADD_NODE
REM_NODE
MOVE_NODE
REMOVE_MACHINE

7.26.3 Documentation des fonctions membres

7.26.3.1 `bool ayoaron ::event ::CEngineEvent ::operator== (const CEngineEvent & e)`
[inline]

opérateur de comparaison

Paramètres

<code>e</code>	l'engine_event à comparer à celui-ci
----------------	--------------------------------------

Renvoie

true s'il sont identiques

Références `i_data`, `s_data`, `type`, et `v_data`.

7.26.3.2 `template<class Archive > void ayoaron ::event ::CEngineEvent ::serialize (Archive & ar, const unsigned int)` [inline]

sérialise les données

Références `i_data`, `s_data`, `type`, et `v_data`.

7.26.4 Documentation des données membres

7.26.4.1 `std ::map<std ::string, int> ayoaron ::event ::CEngineEvent ::i_data`

donnée de type entier

Référencé par `operator==()`, `ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents()`, `ayoaron ::reseau ::CNetworkInterface ::sendAdd()`, `ayoaron ::reseau ::CNetworkInterface ::sendMove()`, `ayoaron ::reseau ::CNetworkInterface ::sendRem()`, `serialize()`, et `ayoaron ::reseau ::CNetworkMachine ::tcpAsyncReceive()`.

7.26.4.2 `std ::map<std ::string, std ::string> ayoaron ::event ::CEngineEvent ::s_data`

données de type chaîne

Référencé par `ayoaron ::reseau ::CNetworkMachine ::connect()`, `operator==()`, `ayoar-`

ron : :reseau : :CNetworkEngine : :processReceivedEvents(), serialize(), et ayoaron : :reseau : :CNetworkEngine : :serverThreadTCPReceive().

7.26.4.3 int ayoaron : :event : :CEngineEvent : :type

type d'événement

Référencé par ayoaron : :core : :CGameRound : :CGameRound(), ayoaron : :reseau : :CNetworkMachine : :connect(), operator==(), ayoaron : :graphic : :CGraphicsEngine : :processEvent(), ayoaron : :reseau : :CNetworkEngine : :processReceivedEvents(), ayoaron : :reseau : :CNetworkInterface : :sendAdd(), ayoaron : :reseau : :CNetworkInterface : :sendMove(), ayoaron : :reseau : :CNetworkInterface : :sendRem(), serialize(), ayoaron : :reseau : :CNetworkEngine : :serverThreadTCPReceive(), et ayoaron : :reseau : :CNetworkMachine : :tcpAsyncReceive().

7.26.4.4 std : :map<std : :string, CSerializableVector3df> ayoaron : :event : :CEngineEvent : :v_data

donnée de type liste de vecteur

Référencé par operator==(), ayoaron : :reseau : :CNetworkEngine : :processReceivedEvents(), ayoaron : :reseau : :CNetworkInterface : :sendAdd(), ayoaron : :reseau : :CNetworkInterface : :sendMove(), et serialize().

La documentation de cette classe a été générée à partir du fichier suivant :

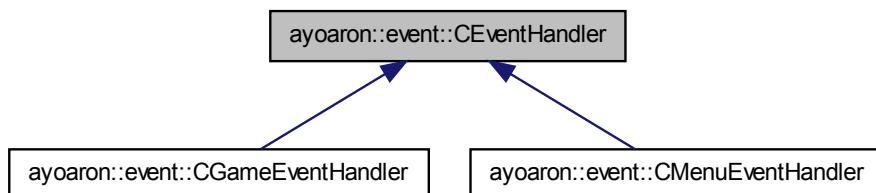
- [CEngineEvent.h](#)

7.27 Référence de la classe ayoaron : :event : :CEventHandler

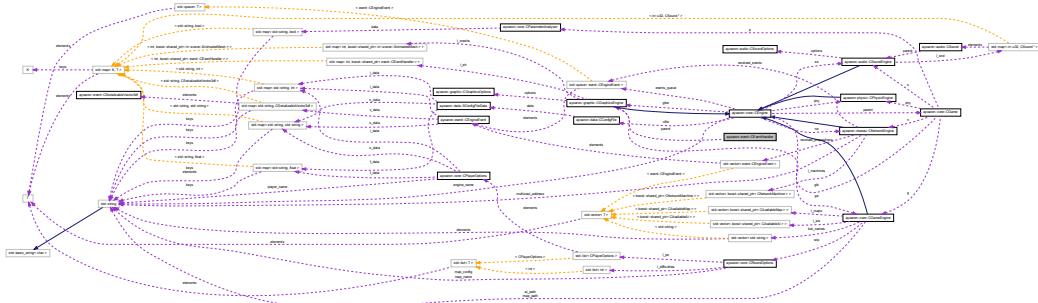
classe générique de gestionnaire d'événement

```
#include <CEventHandler.h>
```

Graphe d'héritage de ayoaron : :event : :CEventHandler :



Graphe de collaboration de ayoaron ::event ::CEventHandler :



Fonctions membres publiques

- **CEventHandler (graphic ::CGraphicsEngine *ge)**
constructeur
- **graphic ::CGraphicsEngine * getParent ()**
revoie le moteur graphique parent
- **bool isKeyPressed (irr ::EKEY_CODE i)**
renvoie si la touche est appuyée
- **virtual bool onEvent (const irr ::SEvent &e)=0**
traitement de l'événement
- **bool OnEvent (const irr ::SEvent &e)**
- **virtual ~CEventHandler ()**
destructeur

Attributs protégés

- **int HasJoystick**
le numéro du joystick à utiliser 0 si aucun
- **irr ::SEvent ::SJoystickEvent joy**
le joystick
- **bool KeyIsPressed [irr ::KEY_KEY_CODES_COUNT]**
si une touche est appuyée
- **irr ::SEvent ::SMouseInput mouse**
la souris
- **graphic ::CGraphicsEngine * parent**
le moteur graphique parent

7.27.1 Description détaillée

classe générique de gestionnaire d'événement

7.27.2 Documentation des constructeurs et destructeur

7.27.2.1 ayoaron ::event ::CEventHandler ::CEventHandler (**graphic** ::CGraphicsEngine * **ge**) [inline]

constructeur

Paramètres

ge	le moteur graphique parent
-----------	----------------------------

Références parent.

7.27.2.2 virtual ayoaron ::event ::CEventHandler ::~CEventHandler () [inline, virtual]

destructeur

7.27.3 Documentation des fonctions membres

7.27.3.1 **graphic** ::CGraphicsEngine* ayoaron ::event ::CEventHandler ::getParent () [inline]

revoie le moteur graphique parent

Renvoie

le moteur graphique parent

Références parent.

7.27.3.2 bool ayoaron ::event ::CEventHandler ::isKeyPressed (irr ::EKEY_CODE *i*) [inline]

renvoie si la touche est appuyée

Renvoie

l'état de la touche

Références KeyIsPressed.

7.27.3.3 virtual bool ayoaron ::event ::CEventHandler ::onEvent (const irr ::SEvent & **e**) [pure virtual]

traitement de l'événement

Paramètres

e	l'événement à traiter
----------	-----------------------

Renvoie

true si tout va bien

Implémenté dans [ayoaron ::event ::CGameEventHandler](#), et [ayoaron ::event ::CMENUEventHandler](#).

Référencé par OnEvent().

Voici le graphe d'appel pour cette fonction :



7.27.3.4 `bool ayoaron : :event : :CEventHandler : :OnEvent (const irr : :SEvent & e)`
[inline]

Références onEvent().

Voici le graphe d'appel pour cette fonction :



7.27.4 Documentation des données membres

7.27.4.1 `int ayoaron : :event : :CEventHandler : :HasJoystick` [protected]

le numéro du joystick à utiliser 0 si aucun

7.27.4.2 `irr : :SEvent : :SJoystickEvent ayoaron : :event : :CEventHandler : :joy`
[protected]

le joystick

Référencé par ayoaron : :event : :CGameEventHandler : :onEvent().

7.27.4.3 `bool ayoaron : :event : :CEventHandler : :KeyIsPressed[irr : :KEY_KEY_CODES_COUNT]` [protected]

si une touche est appuyée

Référencé par ayoaron : :event : :CGameEventHandler : :CGameEventHandler(), isKeyPressed(), et ayoaron : :event : :CGameEventHandler : :onEvent().

7.27.4.4 irr : :SEvent : :SMouseInput ayoaron : :event : :CEventHandler : :mouse
[protected]

la souris

Référencé par ayoaron : :event : :CGameEventHandler : :onEvent().

7.27.4.5 graphic : :CGraphicsEngine* ayoaron : :event : :CEventHandler : :parent
[protected]

le moteur graphique parent

Référencé par CEventHandler(), ayoaron : :event : :CGameEventHandler : :CGameEventHandler(), ayoaron : :event : :CMenuEventHandler : :CMenuEventHandler(), getParent(), ayoaron : :event : :CMenuEventHandler : :onEvent(), et ayoaron : :event : :CGameEventHandler : :onEvent().

La documentation de cette classe a été générée à partir du fichier suivant :

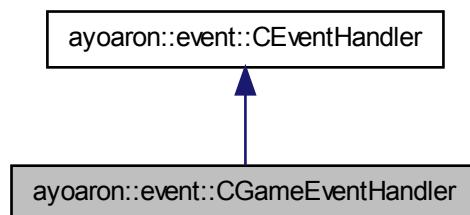
– [CEventHandler.h](#)

7.28 Référence de la classe ayoaron : :event : :CGameEventHandler

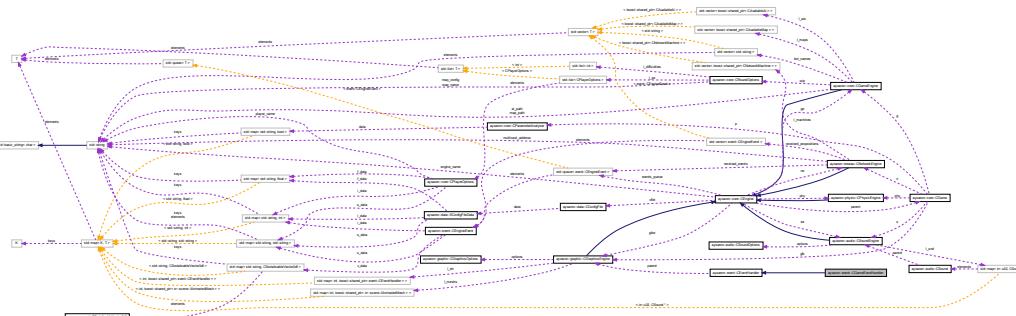
gestionnaire d'évenement du jeu

```
#include <CGameEventHandler.h>
```

Graphe d'héritage de ayoaron : :event : :CGameEventHandler :



Graphe de collaboration de ayoaron ::event ::CGameEventHandler :



Fonctions membres publiques

- **CGameEventHandler (graphic : :CGraphicsEngine *)**
constructeur
- **graphic : :CGraphicsEngine * getParent ()**
revoie le moteur graphique parent
- **bool isKeyPressed (irr : :EKEY_CODE i)**
renvoie si la touche est appuyée
- **bool onEvent (const irr : :SEvent &)**
traitement d'événement
- **bool OnEvent (const irr : :SEvent &e)**
- **virtual ~CGameEventHandler ()**
destructeur

Attributs protégés

- **int HasJoystick**
le numéro du joystick à utiliser 0 si aucun
- **irr : :SEvent : :SJoystickEvent joy**
le joystick
- **bool KeysPressed [irr : :KEY_KEY_CODES_COUNT]**
si une touche est appuyée
- **irr : :SEvent : :SMouseInput mouse**
la souris
- **graphic : :CGraphicsEngine * parent**
le moteur graphique parent

7.28.1 Description détaillée

gestionnaire d'événement du jeu

7.28.2 Documentation des constructeurs et destructeur

7.28.2.1 **virtual ayoaron ::event ::CGameEventHandler ::~CGameEventHandler ()**
[inline, virtual]

destructeur

7.28.2.2 **ayoaron ::event ::CGameEventHandler ::CGameEventHandler (**
graphic : :CGraphicsEngine * *p* **)**

constructeur

Paramètres

in	<i>p</i> moteur graphique parent
----	----------------------------------

Références ayoaron ::graphic ::CGraphicsEngine ::getGui(), ayoaron ::event ::CEventHandler ::KeyIsPressed, et ayoaron ::event ::CEventHandler ::parent.

Voici le graphe d'appel pour cette fonction :



7.28.3 Documentation des fonctions membres

7.28.3.1 **graphic : :CGraphicsEngine* ayoaron ::event ::CEventHandler ::getParent ()**
[inline, inherited]

revoie le moteur graphique parent

Renvoie

le moteur graphique parent

Références ayoaron ::event ::CEventHandler ::parent.

7.28.3.2 **bool ayoaron ::event ::CEventHandler ::isKeyPressed (irr ::EKEY_CODE *i*)**
[inline, inherited]

renvoie si la touche est appuyée

Renvoie

l'état de la touche

Références ayoaron ::event ::CEventHandler ::KeyIsPressed.

7.28.3.3 bool ayoaron ::event ::CGameEventHandler ::onEvent (const irr ::SEvent & e)
 [virtual]

traitement d'événement

Paramètres

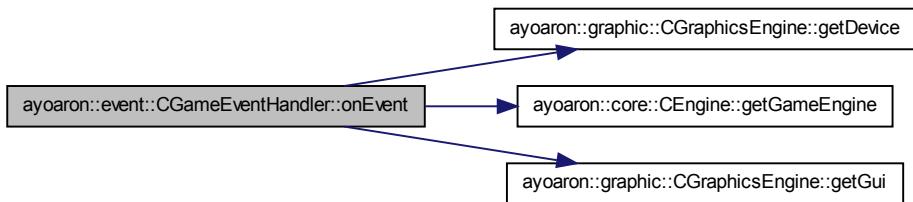
in	e l'événement
----	-----------------

envoie des données de la souris au GUI

Implémente [ayoaron ::event ::CEventHandler](#).

Références ayoaron ::graphic ::CGraphicsEngine ::getDevice(), ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::graphic ::CGraphicsEngine ::getGui(), ayoaron ::event ::CEventHandler ::joy, ayoaron ::event ::CEventHandler ::KeyPressed, ayoaron ::event ::CEventHandler ::mouse, et ayoaron ::event ::CEventHandler ::parent.

Voici le graphe d'appel pour cette fonction :



7.28.3.4 bool ayoaron ::event ::CEventHandler ::OnEvent (const irr ::SEvent & e)
 [inline, inherited]

Références ayoaron ::event ::CEventHandler ::onEvent().

Voici le graphe d'appel pour cette fonction :



7.28.4 Documentation des données membres

7.28.4.1 **int ayoaron : :event : :CEventHandler : :HasJoystick** [protected, inherited]

le numéro du joystick à utiliser 0 si aucun

7.28.4.2 **irr : :SEvent : :SJoystickEvent ayoaron : :event : :CEventHandler : :joy** [protected, inherited]

le joystick

Référencé par onEvent().

7.28.4.3 **bool ayoaron : :event : :CEventHandler : :KeyIsPressed[irr : :KEY_KEY_CODES_COUNT]** [protected, inherited]

si une touche est appuyée

Référencé par CGameEventHandler(), ayoaron : :event : :CEventHandler : :isKeyPressed(), et onEvent().

7.28.4.4 **irr : :SEvent : :SMouseInput ayoaron : :event : :CEventHandler : :mouse** [protected, inherited]

la souris

Référencé par onEvent().

7.28.4.5 **graphic : :CGraphicsEngine* ayoaron : :event : :CEventHandler : :parent** [protected, inherited]

le moteur graphique parent

Référencé par ayoaron : :event : :CEventHandler : :CEventHandler(), CGameEventHandler(), ayoaron : :event : :CMenuEventHandler : :CMenuEventHandler(), ayoaron : :event : :CEventHandler : :getParent(), ayoaron : :event : :CMenuEventHandler : :onEvent(), et onEvent().

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

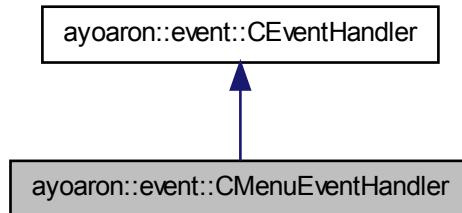
- [CGameEventHandler.h](#)
- [CGameEventHandler.cpp](#)

7.29 Référence de la classe ayoaron : :event : :CMenuEventHandler

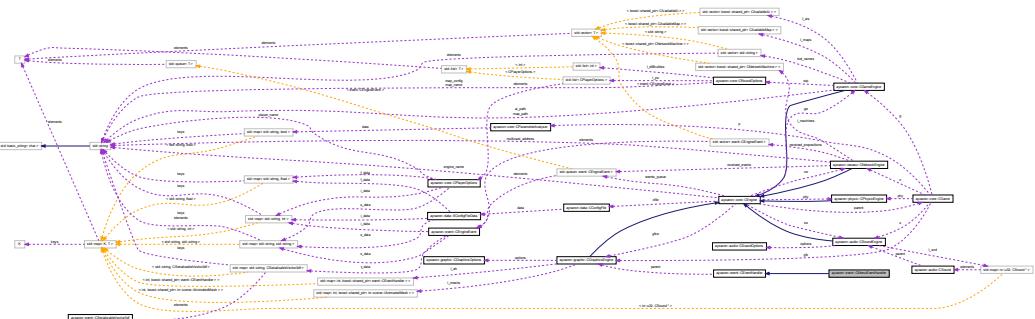
gestionnaire d'événement du menu

```
#include <CMenuEventHandler.h>
```

Graphe d'héritage de ayoaron : :event : :CMenuEventHandler :



Graphe de collaboration de ayoaron ::event ::CMenuEventHandler :



Fonctions membres publiques

- **CMenuEventHandler** (graphic : :CGraphicsEngine *)
constructeur
 - **graphic** : :CGraphicsEngine * **getParent** ()
revoie le moteur graphique parent
 - **bool isKeyPressed** (irr : :EKEY_CODE i)
renvoie si la touche est appuyée
 - **bool onEvent** (const irr : :SEvent &)
traitement d'événement
 - **bool OnEvent** (const irr : :SEvent &e)
 - **virtual ~CMenuEventHandler** ()
destructeur

Attributs protégés

- int **HasJoystick**
le numéro du joystick à utiliser 0 si aucun

- irr : :SEvent : :SJoystickEvent *joy*
le joystick
- bool **KeyIsPressed** [irr : :KEY_KEY_CODES_COUNT]
si une touche est appuyée
- irr : :SEvent : :SMouseInput *mouse*
la souris
- **graphic** : :CGraphicsEngine * *parent*
le moteur graphique parent

7.29.1 Description détaillée

gestionnaire d'évenement du menu

7.29.2 Documentation des constructeurs et destructeur

7.29.2.1 **virtual ayoaron : :event : :CMenuEventHandler : :~CMenuEventHandler ()**
[inline, virtual]

destructeur

7.29.2.2 **ayoaron : :event : :CMenuEventHandler : :CMenuEventHandler (graphic : :CGraphicsEngine * *p*)**

constructeur

Paramètres

in	<i>p</i>	moteur graphique parent
----	----------	-------------------------

Références ayoaron : :event : :CEventHandler : :parent.

7.29.3 Documentation des fonctions membres

7.29.3.1 **graphic : :CGraphicsEngine* ayoaron : :event : :CEventHandler : :getParent ()**
[inline, inherited]

revoie le moteur graphique parent

Renvoie

le moteur graphique parent

Références ayoaron : :event : :CEventHandler : :parent.

7.29.3.2 **bool ayoaron : :event : :CEventHandler : :isKeyPressed (irr : :EKEY_CODE *i*)**
[inline, inherited]

renvoie si la touche est appuyée

Renvoie

l'état de la touche

Références ayoaron ::event ::CEventHandler ::KeyIsPressed.

7.29.3.3 **bool ayoaron ::event ::CMenuEventHandler ::onEvent (const irr ::SEvent & e)**
[virtual]

traitement d'événement

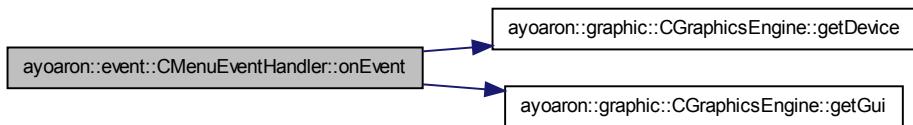
Paramètres

in	e	l'événement
----	---	-------------

Implémente ayoaron ::event ::CEventHandler.

Références ayoaron ::graphic ::CGraphicsEngine ::getDevice(), ayoaron ::graphic ::CGraphicsEngine ::getGui(), et ayoaron ::event ::CEventHandler ::parent.

Voici le graphe d'appel pour cette fonction :



7.29.3.4 **bool ayoaron ::event ::CEventHandler ::OnEvent (const irr ::SEvent & e)**
[inline, inherited]

Références ayoaron ::event ::CEventHandler ::onEvent().

Voici le graphe d'appel pour cette fonction :



7.29.4 Documentation des données membres

7.29.4.1 **int ayoaron ::event ::CEventHandler ::HasJoystick** [protected, inherited]

le numéro du joystick à utiliser 0 si aucun

7.29.4.2 **irr : :SEvent : :SJoystickEvent** ayoaron : :event : :CEventHandler : :joy
 [protected, inherited]

le joystick

Référencé par ayoaron : :event : :CGameEventHandler : :onEvent().

7.29.4.3 **bool ayoaron : :event : :CEventHandler : :KeyIsPressed**[irr : :KEY_KEY_CODES_COUNT] [protected, inherited]

si une touche est appuyée

Référencé par ayoaron : :event : :CGameEventHandler : :CGameEventHandler(), ayoaron : :event : :CEventHandler : :isKeyPressed(), et ayoaron : :event : :CGameEventHandler : :onEvent().

7.29.4.4 **irr : :SEvent : :SMouseInput** ayoaron : :event : :CEventHandler : :mouse
 [protected, inherited]

la souris

Référencé par ayoaron : :event : :CGameEventHandler : :onEvent().

7.29.4.5 **graphic : :GraphicsEngine*** ayoaron : :event : :CEventHandler : :parent
 [protected, inherited]

le moteur graphique parent

Référencé par ayoaron : :event : :CEventHandler : :CEventHandler(), ayoaron : :event : :CGameEventHandler : :CGameEventHandler(), CMuEventHandler(), ayoaron : :event : :CEventHandler : :getParent(), onEvent(), et ayoaron : :event : :CGameEventHandler : :onEvent().

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

- [CMuEventHandler.h](#)
- [CMuEventHandler.cpp](#)

7.30 Référence de la classe ayoaron : :event : :CSerializableVector3df

classe vecteur sérialisable

```
#include <CEngineEvent.h>
```

Fonctions membres publiques

- [CSerializableVector3df \(\)](#)
constructeur
- [CSerializableVector3df \(irr : :core : :vector3df &v\)](#)
constructeur
- template<class Archive >
 void [serialize \(Archive &ar, const unsigned int\)](#)
sérialise ce vecteur

7.30.1 Description détaillée

classe vecteur sérialisable

7.30.2 Documentation des constructeurs et destructeur

7.30.2.1 ayoaron ::event ::CSerializableVector3df ::CSerializableVector3df() [inline]

constructeur

7.30.2.2 ayoaron ::event ::CSerializableVector3df ::CSerializableVector3df(irr ::core ::vector3df & v) [inline]

constructeur

Paramètres

v	vecteur initialisant celui que l'on veut créer
---	--

7.30.3 Documentation des fonctions membres

7.30.3.1 template<class Archive> void ayoaron ::event ::CSerializableVector3df ::serialize(Archive & ar, const unsigned int) [inline]

sérialise ce vecteur

La documentation de cette classe a été générée à partir du fichier suivant :

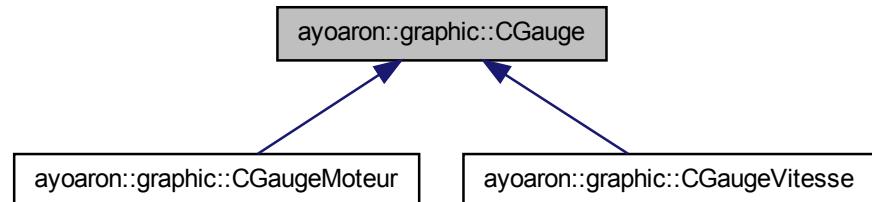
– [CEngineEvent.h](#)

7.31 Référence de la classe ayoaron ::graphic ::CGauge

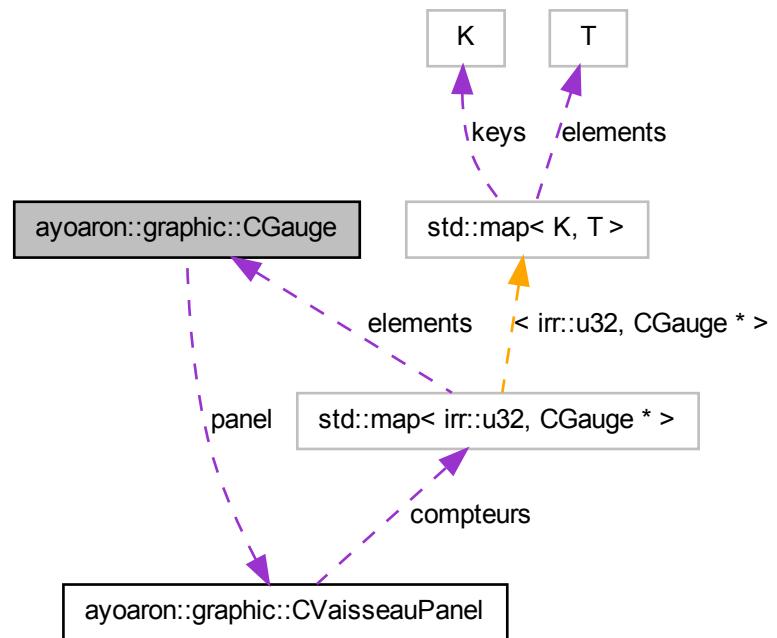
classe d'encapsulation des compteurs

```
#include <CGauge.h>
```

Graphe d'héritage de ayoaron ::graphic ::CGauge :



Graphe de collaboration de ayoaron ::graphic ::CGauge :



Fonctions membres publiques

- **CGauge (CVaisseauPanel *p)**
Constructeur.
- virtual void **draw ()**
dessine le compteur
- virtual void **setBackground (std ::string f="")**
définit l'image de fond
- virtual void **setPosition (irr ::core ::position2d< irr ::s32 > ppos=irr ::core ::position2d< irr ::s32 >(0, 0))**
définit la position du compteur (coin supérieur gauche)
- virtual void **setScale (irr ::core ::vector2df s=irr ::core ::vector2df(0, 0))**
définit l'échelle
- virtual **~CGauge ()**
Destructeur.

Attributs protégés

- irr ::video ::ITexture * **background**
nom de l'image de fond
- **CVaisseauPanel * panel**
pointeur vers le device d'irrlicht
- irr ::core ::position2d< irr ::s32 > **pos**
position de l'image background
- irr ::core ::vector2df **scale**
l'échelle

7.31.1 Description détaillée

classe d'encapsulation des compteurs

7.31.2 Documentation des constructeurs et destructeur

7.31.2.1 ayoaron ::graphic ::CGauge (CVaisseauPanel * p)

Constructeur.

Références ayoaron ::graphic ::CVaisseauPanel ::addGauge(), background, panel, pos, et scale.

Voici le graphe d'appel pour cette fonction :



7.31.2.2 ayoaron ::graphic ::CGauge ::~CGauge () [virtual]

Destructeur.

7.31.3 Documentation des fonctions membres

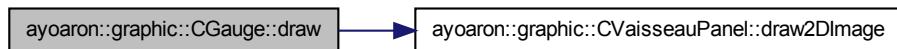
7.31.3.1 void ayoaron ::graphic ::CGauge ::draw () [virtual]

dessine le compteur

Réimplémentée dans [ayoaron ::graphic ::CGaugeMoteur](#), et [ayoaron ::graphic ::CGaugeVitesse](#).

Références background, ayoaron ::graphic ::CVaisseauPanel ::draw2DImage(), panel, et pos.

Voici le graphe d'appel pour cette fonction :



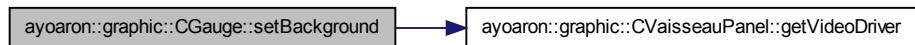
7.31.3.2 void ayoaron ::graphic ::CGauge ::setBackground (std ::string f = " ") [virtual]

définit l'image de fond

Références background, ayoaron ::graphic ::CVaisseauPanel ::getVideoDriver(), panel, et scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



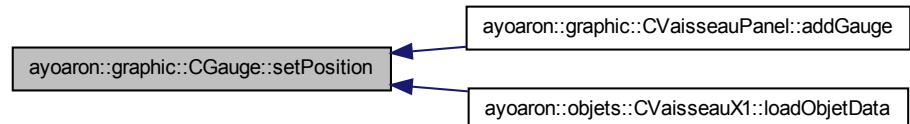
7.31.3.3 void ayoaron ::graphic ::CGauge ::setPosition (irr ::core ::position2d< irr ::s32 > ppos = irr ::core ::position2d< irr ::s32 > (0, 0)) [virtual]

définit la position du compteur (coin supérieur gauche)

Références background, pos, et scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.31.3.4 void ayoaron ::graphic ::CGauge ::setScale (irr ::core ::vector2df s = irr ::core ::vector2df (0, 0)) [virtual]

définit l'échelle

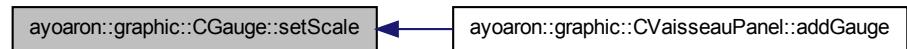
Références ayoaron ::graphic ::CVaisseauPanel ::getGlobalScale(), panel, et scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.31.4 Documentation des données membres

7.31.4.1 irr ::video ::ITexture* ayoaron ::graphic ::CGauge ::background [protected]

nom de l'image de fond

Référencé par CGauge(), ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), draw(), setBackground(), et setPosition().

7.31.4.2 CVaisseauPanel* ayoaron ::graphic ::CGauge ::panel [protected]

pointeur vers le device d'irrlicht

Référencé par CGauge(), ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), draw(), setBackground(), et setScale().

7.31.4.3 irr ::core ::position2d<irr ::s32> ayoaron ::graphic ::CGauge ::pos [protected]

position de l'image background

Référencé par CGauge(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), draw(), et setPosition().

7.31.4.4 irr ::core ::vector2df ayoaron ::graphic ::CGauge ::scale [protected]

l'echelle

Référencé par CGauge(), ayoaron ::graphic ::CGaugeMoteur ::draw(), setBackground(), setPosition(), et setScale().

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

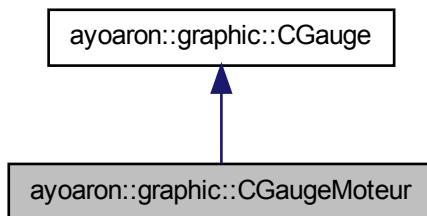
- [CGauge.h](#)
- [CGauge.cpp](#)

7.32 Référence de la classe ayoaron ::graphic ::CGaugeMoteur

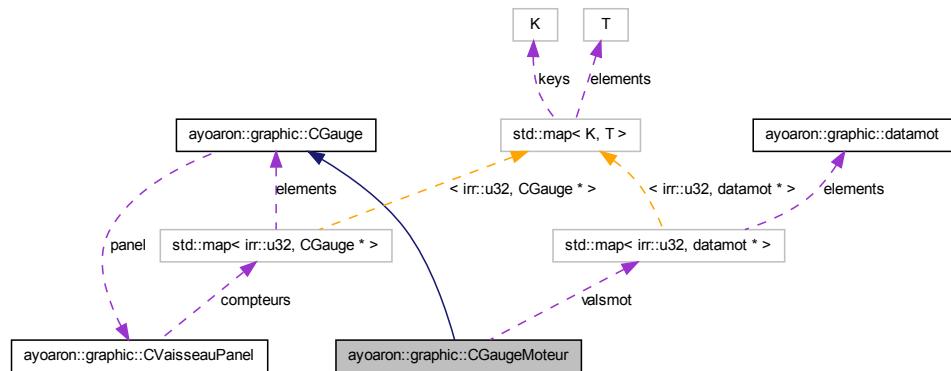
classe de dessin de l'écran moteur

```
#include <CGaugeMoteur.h>
```

Graphe d'héritage de ayoaron ::graphic ::CGaugeMoteur :



Graphe de collaboration de ayoaron ::graphic ::CGaugeMoteur :



Fonctions membres publiques

- **CGaugeMoteur** (CVaisseauPanel *p, std ::string back, std ::string perm, std ::string mot)

Constructeur.
- void **draw** ()

dessine le bazarre
- virtual void **setBackground** (std ::string f="")

définit l'image de fond
- void **setCadres** (irr ::u32 id, irr ::core ::rect< irr ::s32 > c1, irr ::core ::rect< irr ::s32 > c2=irr ::core ::rect< irr ::s32 >(0, 0, 0, 0))

Définit la position du moteur i dans l'écran virtuel.
- void **setNmoteur** (irr ::u32 N)

Définit le nombre de moteur.
- virtual void **setPosition** (irr ::core ::position2d< irr ::s32 > ppos=irr ::core ::position2d< irr ::s32 >(0, 0))

définit la position du compteur (coin supérieur gauche)
- void **setRegime** (irr ::u32 id, irr ::f64 reg)

Définit le régime actuel du moteur.
- virtual void **setScale** (irr ::core ::vector2df s=irr ::core ::vector2df(0, 0))

définit l'échelle
- virtual **~CGaugeMoteur** ()

Destructeur.

Attributs protégés

- irr ::video ::ITexture * **background**

nom de l'image de fond
- CVaisseauPanel * **panel**

pointeur vers le device d'irrlicht
- irr ::core ::position2d< irr ::s32 > **pos**

position de l'image background
- irr ::core ::vector2df **scale**

l'échelle

Attributs privés

- irr ::video ::ITexture * **affiche_perm**
l'image du fond de l'écran virtuel
- irr ::video ::ITexture * **moteurs**
l'image des bacs d'affichage
- irr ::u32 **Nmoteur**
le nombre des moteurs
- std ::map< irr ::u32, **datamot** * > **valsmtot**
la valeurs des moteurs

7.32.1 Description détaillée

classe de dessin de l'écran moteur

7.32.2 Documentation des constructeurs et destructeur

7.32.2.1 ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur (CVaisseauPanel * *p*, std ::string *back*, std ::string *perm*, std ::string *mot*)

Constructeur.

Références *affiche_perm*, *ayoaron ::graphic ::CGauge ::background*, *ayoaron ::graphic ::CVaisseauPanel ::getVideoDriver()*, *moteurs*, *Nmoteur*, *ayoaron ::graphic ::CGauge ::panel*, et *valsmtot*.

Voici le graphe d'appel pour cette fonction :



7.32.2.2 ayoaron ::graphic ::CGaugeMoteur ::~CGaugeMoteur () [virtual]

Destructeur.

7.32.3 Documentation des fonctions membres

7.32.3.1 void ayoaron ::graphic ::CGaugeMoteur ::draw () [virtual]

dessine le bazarre

Réimplémentée à partir de [ayoaron ::graphic ::CGauge](#).

Références *affiche_perm*, *ayoaron ::graphic ::CGauge ::background*, *ayoaron ::graphic ::CVaisseauPanel ::draw2DImage()*, *moteurs*, *Nmoteur*, *ayoaron ::graphic ::CGauge ::panel*

nel, ayoaron ::graphic ::CGauge ::pos, ayoaron ::graphic ::CGauge ::scale, et vals-
mot.

Voici le graphe d'appel pour cette fonction :



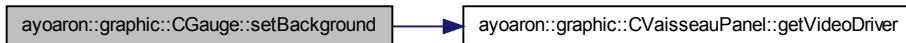
7.32.3.2 `void ayoaron ::graphic ::CGauge ::setBackground (std ::string f = " ")`
`[virtual, inherited]`

définit l'image de fond

Références ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CVais-
seauPanel ::getVideoDriver(), ayoaron ::graphic ::CGauge ::panel, et ayoaron ::gra-
phic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.32.3.3 `void ayoaron ::graphic ::CGaugeMoteur ::setCadres (irr ::u32 id,
irr ::core ::rect< irr ::s32 > c1, irr ::core ::rect< irr ::s32 > c2 =
irr ::core ::rect< irr ::s32 > (0, 0, 0, 0))`

Définit la position du moteur i dans l'écran virtuel.

Références Nmoteur, et valsmt.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.32.3.4 void ayoaron ::graphic ::CGaugeMoteur ::setNmoteur (irr ::u32 *N*)

Définit le nombre de moteur.

Références Nmoteur, et valsmt.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



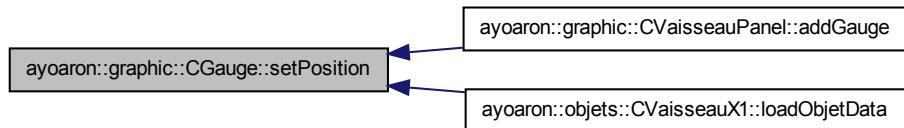
7.32.3.5 void ayoaron ::graphic ::CGauge ::setPosition (irr ::core ::position2d< irr ::s32 > *ppos* = irr ::core ::position2d< irr ::s32 > (0, 0)) [virtual, inherited]

définit la position du compteur (coin supérieur gauche)

Références ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CGauge ::pos, et ayoaron ::graphic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.32.3.6 void ayoaron ::graphic ::CGaugeMoteur ::setRegime (irr ::u32 id, irr ::f64 reg)

Définit le régime actuel du moteur.

Références Nmoteur, et valsmot.

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.32.3.7 void ayoaron ::graphic ::CGauge ::setScale (irr ::core ::vector2df s = irr ::core ::vector2df(0, 0)) [virtual, inherited]

définit l'échelle

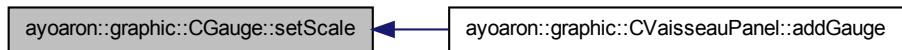
Références ayoaron ::graphic ::CVaisseauPanel ::getGlobalScale(), ayoaron ::graphic ::CGauge ::panel, et ayoaron ::graphic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.32.4 Documentation des données membres

7.32.4.1 irr : :video : :ITexture* ayoaron : :graphic : :CGaugeMoteur : :affiche_perm
[private]

l'image du fond de l'écran virtuel

Référencé par CGaugeMoteur(), et draw().

7.32.4.2 irr : :video : :ITexture* ayoaron : :graphic : :CGauge : :background
[protected, inherited]

nom de l'image de fond

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), CGaugeMoteur(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), draw(), ayoaron ::graphic ::CGauge ::draw(), ayoaron ::graphic ::CGauge ::setScale(), et ayoaron ::graphic ::CGauge ::setPosition().

7.32.4.3 irr : :video : :ITexture* ayoaron : :graphic : :CGaugeMoteur : :moteurs
[private]

l'image des bacs d'affichage

Référencé par CGaugeMoteur(), et draw().

7.32.4.4 irr : :u32 ayoaron : :graphic : :CGaugeMoteur : :Nmoteur [private]

le nombre des moteurs

Référencé par CGaugeMoteur(), draw(), setCadres(), setNmoteur(), et setRegime().

7.32.4.5 CVaisseauPanel* ayoaron : :graphic : :CGauge : :panel [protected, inherited]

pointeur vers le device d'irrlicht

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), CGaugeMoteur(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), draw(), ayoaron ::graphic ::CGauge ::draw(), ayoaron ::graphic ::CGauge ::setScale(), et ayoaron ::graphic ::CGauge ::setScale().

7.32.4.6 **irr ::core ::position2d<irr ::s32> ayoaron ::graphic ::CGauge ::pos** [protected, inherited]

position de l'image background

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), ayoaron ::graphic ::CGaugeVitesse ::draw(), draw(), ayoaron ::graphic ::CGauge ::draw(), et ayoaron ::graphic ::CGauge ::setPosition().

7.32.4.7 **irr ::core ::vector2df ayoaron ::graphic ::CGauge ::scale** [protected, inherited]

l'echelle

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), draw(), ayoaron ::graphic ::CGauge ::setPosition(), ayoaron ::graphic ::CGauge ::setPosition(), et ayoaron ::graphic ::CGauge ::setScale().

7.32.4.8 **std ::map<irr ::u32,datamot*> ayoaron ::graphic ::CGaugeMoteur ::valsmot** [private]

la valeurs des moteurs

Référencé par CGaugeMoteur(), draw(), setCadres(), setNmoteur(), et setRegime().

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

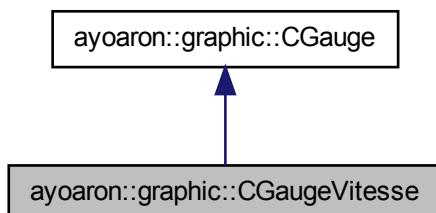
- [CGaugeMoteur.h](#)
- [CGaugeMoteur.cpp](#)

7.33 Référence de la classe ayoaron ::graphic ::CGaugeVitesse

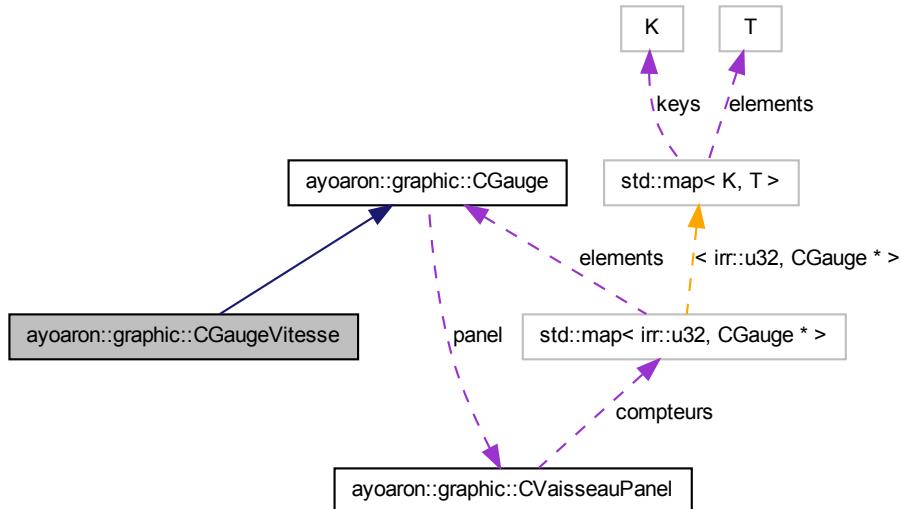
la classe de gestion du compteur de vitesse à six axes

```
#include <CGaugeVitesse.h>
```

Graphe d'héritage de ayoaron ::graphic ::CGaugeVitesse :



Graphe de collaboration de ayoaron ::graphic ::CGaugeVitesse :



Fonctions membres publiques

- **CGaugeVitesse** (CVaisseauPanel *p, std :: string back, std :: string aig)
constructeur
- void **draw** ()
dessine le cadran
- virtual void **setBackground** (std :: string f="")
définit l'image de fond
- void **setCurseur** (irr :: core :: vector3df vit, irr :: core :: vector3df vro)
defini les curseurs instantanés
- void **setLimit** (irr :: u8 i, irr :: core :: vector2df lim)
defini les max et min pour l'indice de vitesse i
- void **setLimitAngulaire** (irr :: core :: vector2df lim)
definit les limites angulaire du cadran tournant
- virtual void **setPosition** (irr :: core :: position2d< irr :: s32 > ppos=irr :: core :: position2d< irr :: s32 > (0, 0))
définit la position du compteur (coin supérieur gauche)
- void **setRectGRXY** (irr :: core :: rect< irr :: s32 > r)
definit le rectangle du compteur GRXY
- void **setRectGRZ** (irr :: core :: rect< irr :: s32 > r)
definit le rectangle du compteur GRZ
- void **setRectGVX** (irr :: core :: rect< irr :: s32 > r)
definit le rectangle du compteur GVX
- void **setRectGVY** (irr :: core :: rect< irr :: s32 > r)
definit le rectangle du compteur GYV
- void **setRectGVZ** (irr :: core :: rect< irr :: s32 > r)
definit le rectangle du compteur GVZ
- virtual void **setScale** (irr :: core :: vector2df s=irr :: core :: vector2df(0, 0))

- *définit l'échelle*
– virtual ~CGaugeVitesse ()
Destructeur.

Attributs protégés

- irr ::video ::ITexture * **background**
nom de l'image de fond
- **CVaisseauPanel** * **panel**
pointeur vers le device d'irrlicht
- irr ::core ::position2d< irr ::s32 > **pos**
position de l'image background
- irr ::core ::vector2df **scale**
l'échelle

Attributs privés

- irr ::video ::ITexture * **aiguille**
l'image de l'aiguille
- irr ::core ::vector2df **limang**
les limites angulaires pour le cadran tournant
- irr ::core ::vector2df **limits** [6]
les limites de vitesses
- irr ::core ::rect< irr ::s32 > **recGRXY**
rectangle ddu cadran GRXY
- irr ::core ::rect< irr ::s32 > **recGRZ**
rectangle ddu cadran GRZ
- irr ::core ::rect< irr ::s32 > **recGVX**
rectangle ddu cadran GVX
- irr ::core ::rect< irr ::s32 > **recGVY**
rectangle ddu cadran GVY
- irr ::core ::rect< irr ::s32 > **recGVZ**
rectangle ddu cadran GVZ
- irr ::core ::vector3df **vite**
le vecteur vitesse
- irr ::core ::vector3df **vrot**
le vecteur vitesse de rotation

7.33.1 Description détaillée

la classe de gestion du compteur de vitesse à six axes

7.33.2 Documentation des constructeurs et destructeur

7.33.2.1 ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse (CVaisseauPanel * *p*, std ::string *back*, std ::string *aig*)

constructeur

Références aiguille, ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CVaisseauPanel ::getVideoDriver(), limang, limits, ayoaron ::graphic ::CGauge ::panel, ayoaron ::graphic ::CGauge ::pos, recGRXY, recGRZ, recGVX, recGVY, recGVZ, vite, et vrot.

Voici le graphe d'appel pour cette fonction :



7.33.2.2 ayoaron ::graphic ::CGaugeVitesse ::~CGaugeVitesse () [virtual]

Destructeur.

7.33.3 Documentation des fonctions membres

7.33.3.1 void ayoaron ::graphic ::CGaugeVitesse ::draw () [virtual]

dessine le cadran

Réimplémentée à partir de [ayoaron ::graphic ::CGauge](#).

Références aiguille, ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CVaisseauPanel ::draw2DImage(), limang, limits, ayoaron ::graphic ::CGauge ::panel, ayoaron ::graphic ::CGauge ::pos, recGRXY, recGRZ, recGVX, recGVY, recGVZ, vite, et vrot.

Voici le graphe d'appel pour cette fonction :



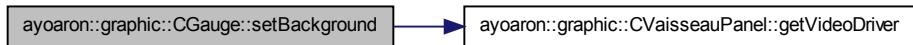
7.33.3.2 void ayoaron ::graphic ::CGauge ::setBackground (std ::string f = " ") [virtual, inherited]

définit l'image de fond

Références ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CVaisseauPanel ::getVideoDriver(), ayoaron ::graphic ::CGauge ::panel, et ayoaron ::graphic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.33.3.3 void ayoaron ::graphic ::CGaugeVitesse ::setCurseur (irr ::core ::vector3df *vit*,
irr ::core ::vector3df *vro*) [inline]**

defini les curseurs instantanés

Paramètres

<i>vit</i>	vitesse la vitesse à afficher
<i>vro</i>	vitesse de rotation à afficher

Références vite, et vrot.

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



**7.33.3.4 void ayoaron ::graphic ::CGaugeVitesse ::setLimit (irr ::u8 *i*, irr ::core ::vector2df
lim) [inline]**

defini les max et min pour l'indice de vitesse i

Paramètres

<i>i</i>	indice de coordonnée de vitesse 0 :horizontal, 1 :vertical, 2 : la partie rotative,3 :rotation autour de x,4 :rotation autour de y,5 :rotation autour de z
<i>lim</i>	la boren inf et sup de vitesse

Références limits.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.5 void ayoaron ::graphic ::CGaugeVitesse ::setLimitAngulaire (irr ::core ::vector2df *lim*) [inline]

définit les limites angulaire du cadran tournant

Paramètres

<i>lim</i>	les limites angulaire
------------	-----------------------

Références limang.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



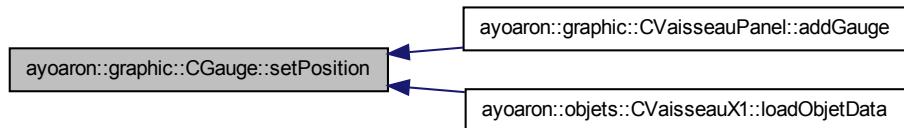
7.33.3.6 void ayoaron ::graphic ::CGauge ::setPosition (irr ::core ::position2d< irr ::s32 > *ppos* = irr ::core ::position2d< irr ::s32 > (0, 0)) [virtual, inherited]

définit la position du compteur (coin supérieur gauche)

Références ayoaron ::graphic ::CGauge ::background, ayoaron ::graphic ::CGauge ::pos, et ayoaron ::graphic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.7 void ayoaron ::graphic ::CGaugeVitesse ::setRectGRXY (irr ::core ::rect< irr ::s32 > r) [inline]

definit le rectangle du compteur GRXY

Paramètres

<i>r</i>	le nouveau rectangle
----------	----------------------

Références recGRXY.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.8 void ayoaron ::graphic ::CGaugeVitesse ::setRectGRZ (irr ::core ::rect< irr ::s32 > r) [inline]

definit le rectangle du compteur GRZ

Paramètres

<i>r</i>	le nouveau rectangle
----------	----------------------

Références recGRZ.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.9 void ayoaron ::graphic ::CGaugeVitesse ::setRectGVX (irr ::core ::rect< irr ::s32 > *r*) [inline]

definit le rectangle du compteur GVX

Paramètres

<i>r</i> le nouveau rectangle

Références recGVX.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.10 void ayoaron ::graphic ::CGaugeVitesse ::setRectGVY (irr ::core ::rect< irr ::s32 > *r*) [inline]

definit le rectangle du compteur GVY

Paramètres

<i>r</i> le nouveau rectangle

Références recGVY.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.11 void ayoaron ::graphic ::CGaugeVitesse ::setRectGVZ (irr ::core ::rect< irr ::s32 > r) [inline]

definit le rectangle du compteur GVZ

Paramètres

<i>r</i> le nouveau rectangle

Références recGVZ.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.33.3.12 void ayoaron ::graphic ::CGauge ::setScale (irr ::core ::vector2df s = irr ::core ::vector2df(0,0)) [virtual, inherited]

définit l'échelle

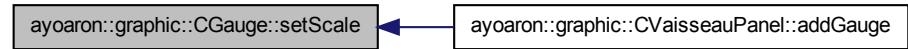
Références ayoaron ::graphic ::CVaisseauPanel ::getGlobalScale(), ayoaron ::graphic ::CGauge ::panel, et ayoaron ::graphic ::CGauge ::scale.

Référencé par ayoaron ::graphic ::CVaisseauPanel ::addGauge().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.33.4 Documentation des données membres

7.33.4.1 irr ::video ::ITexture* ayoaron ::graphic ::CGaugeVitesse ::aiguille
[private]

l'image de l'aiguille

Référencé par CGaugeVitesse(), et draw().

7.33.4.2 irr ::video ::ITexture* ayoaron ::graphic ::CGauge ::background
[protected, inherited]

nom de l'image de fond

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur(), CGaugeVitesse(), draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), ayoaron ::graphic ::CGauge ::draw(), ayoaron ::graphic ::CGauge ::setBackground(), et ayoaron ::graphic ::CGauge ::setPosition().

7.33.4.3 irr ::core ::vector2df ayoaron ::graphic ::CGaugeVitesse ::limang
[private]

les limites angulaires pour le cadran tournant

Référencé par CGaugeVitesse(), draw(), et setLimitAngulaire().

7.33.4.4 irr ::core ::vector2df ayoaron ::graphic ::CGaugeVitesse ::limits[6]
[private]

les limites de vitesses

Référencé par CGaugeVitesse(), draw(), et setLimit().

7.33.4.5 **CVaisseauPanel* ayoaron ::graphic ::CGauge ::panel** [protected, inherited]

pointeur vers le device d'irrlicht

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur(), CGaugeVitesse(), draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), ayoaron ::graphic ::CGauge ::draw(), ayoaron ::graphic ::CGauge ::setBackground(), et ayoaron ::graphic ::CGauge ::setScale().

7.33.4.6 **irr ::core ::position2d<irr ::s32> ayoaron ::graphic ::CGauge ::pos** [protected, inherited]

position de l'image background

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), CGaugeVitesse(), draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), ayoaron ::graphic ::CGauge ::draw(), et ayoaron ::graphic ::CGauge ::setPosition().

7.33.4.7 **irr ::core ::rect<irr ::s32> ayoaron ::graphic ::CGaugeVitesse ::recGRXY** [private]

rectangle ddu cadran GRXY

Référencé par CGaugeVitesse(), draw(), et setRectGRXY().

7.33.4.8 **irr ::core ::rect<irr ::s32> ayoaron ::graphic ::CGaugeVitesse ::recGRZ** [private]

rectangle ddu cadran GRZ

Référencé par CGaugeVitesse(), draw(), et setRectGRZ().

7.33.4.9 **irr ::core ::rect<irr ::s32> ayoaron ::graphic ::CGaugeVitesse ::recGVX** [private]

rectangle ddu cadran GVX

Référencé par CGaugeVitesse(), draw(), et setRectGVX().

7.33.4.10 **irr ::core ::rect<irr ::s32> ayoaron ::graphic ::CGaugeVitesse ::recGVY** [private]

rectangle ddu cadran GVY

Référencé par CGaugeVitesse(), draw(), et setRectGVY().

7.33.4.11 **irr ::core ::rect<irr ::s32> ayoaron ::graphic ::CGaugeVitesse ::recGVZ** [private]

rectangle ddu cadran GVZ

Référencé par CGaugeVitesse(), draw(), et setRectGVZ().

7.33.4.12 irr ::core ::vector2df ayoaron ::graphic ::CGauge ::scale
[protected, inherited]

l'échelle

Référencé par ayoaron ::graphic ::CGauge ::CGauge(), ayoaron ::graphic ::CGaugeMoteur ::draw(), ayoaron ::graphic ::CGauge ::setBackground(), ayoaron ::graphic ::CGauge ::setPosition(), et ayoaron ::graphic ::CGauge ::setScale().

7.33.4.13 irr ::core ::vector3df ayoaron ::graphic ::CGaugeVitesse ::vite
[private]

le vecteur vitesse

Référencé par CGaugeVitesse(), draw(), et setCurseur().

7.33.4.14 irr ::core ::vector3df ayoaron ::graphic ::CGaugeVitesse ::vrot
[private]

le vecteur vitesse de rotation

Référencé par CGaugeVitesse(), draw(), et setCurseur().

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

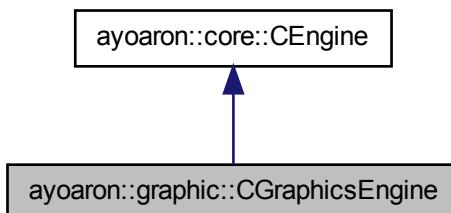
- [CGaugeVitesse.h](#)
- [CGaugeVitesse.cpp](#)

7.34 Référence de la classe ayoaron ::graphic ::CGraphicsEngine

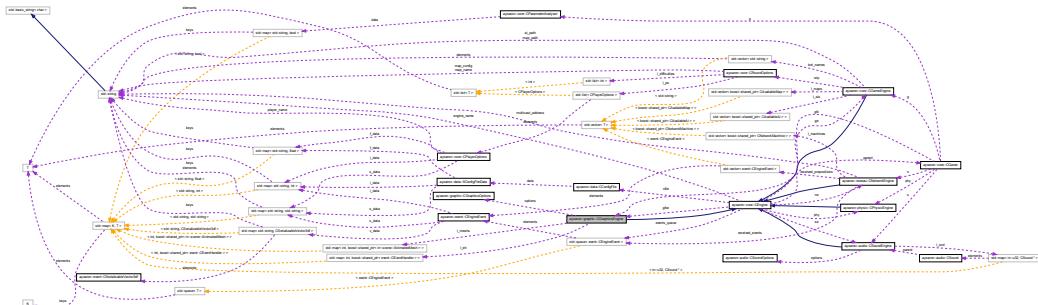
le moteur graphique

```
#include <CGraphicsEngine.h>
```

Graphe d'héritage de ayoaron ::graphic ::CGraphicsEngine :



Graphe de collaboration de ayoaron ::graphic ::CGraphicsEngine :



Types publics

- enum {

STATE NOTHING, STATE QUITTING, STATE WAITING PLAYERS, STATE PLAYING -

INTRO,

STATE LOADING ROUND, STATE PLAYING
 }

Constantes d'état.
- enum { ON MENUS, ON LOADING, ON GAME }

liste des type de gestionnaire d'événement

Fonctions membres publiques

- void attachGameEngine (CGameEngine *e)

lien avec le moteur de jeu
- void attachGraphicsEngine (graphic ::CGraphicsEngine *e)

lien avec le moteur de rendu graphique
- void attachNetworkEngine (reseau ::CNetworkEngine *e)

lien avec le moteur de réseau
- void attachPhysicEngine (physic ::CPhysicEngine *e)

lien avec le moteur de physique
- void attachSoundEngine (audio ::CSoundEngine *e)

lien avec le moteur de son
- CGraphicsEngine (core ::CGame *, bool)

constructeur
- void frame (double dt)

Méthode frame.
- int getDbgLevel ()

renvoie le niveau de debug
- irr ::IrrlichtDevice * getDevice ()

renvoie le device de irrlicht
- irr ::video ::VideoDriver * getDriver ()

renvoie le driver d'affichage
- CGameEngine * getGameEngine ()

lien vers le moteur de jeu
- graphic ::CGraphicsEngine * getGraphicsEngine ()

lien vers le moteur de rendu graphique
- boost ::shared_ptr< CGuiSystem > getGui ()

renvoie le système d'interface utilisateur
- CEGUI ::IrrlichtRenderer * getIrrRenderer ()

- **reseau** : `CNetworkEngine * getNetworkEngine ()`
revoie le moteur de rendu de l'interface utilisateur
- **reseau** : `CNetworkEngine * getNetworkEngine ()`
lien vers le moteur de réseau
- **CGame * getParent ()**
lien vers la classe game parente
- **boost : shared_ptr< CSceneGraph > getSceneGraph ()**
revoie le système de graph de scène
- **audio : CSoundEngine * getSoundEngine ()**
lien vers le moteur de son
- **void processQueue ()**
vidange de l'event_queue
- **void pushEvent (event : CEngineEvent &e)**
ajout d'un événement
- **void requestClose ()**
demande de fermeture
- **void sendMessageToGame (event : CEngineEvent &e)**
envoie un message au moteur de jeu
- **void sendMessageToGraphics (event : CEngineEvent &e)**
envoie un message au moteur de rendu graphique
- **void sendMessageToNetwork (event : CEngineEvent &e)**
envoie un message au moteur de réseau
- **void sendMessageToPhysic (event : CEngineEvent &e)**
envoie un message au moteur physique
- **void sendMessageToSound (event : CEngineEvent &e)**
envoie un message au moteur de son
- **void setState (int)**
définit l'état
- **virtual ~CGraphicsEngine ()**
destructeur

Attributs publics

- **std : map< int, boost : shared_ptr< event : CEventHandler > > l_eh**
liste des gestionnaires d'événement
- **std : map< int, boost : shared_ptr< irr : scene : IAnimatedMesh > > l_meshs**
listes des maillages des objets
- **int state**
l'état actuel
- **boost : mutex state_mutex**
Mutex de protection de l'état.

Fonctions membres protégées

- **void getConfigData ()**
récupération de la configuration actuelle
- **void processEvent (event : CEngineEvent &)**
traitement d'un événement

Attributs protégés

- **data : CConfigFile cfile**
Fichier de configuration.
- **irr : video : IVideoDriver * driver**
pointeur vers le device video
- **std : string engine_name**
nom de ce moteur

- std ::queue< [event](#) : :CEngineEvent > events_queue
 - liste des événement à traiter*
- CGameEngine * ge
 - pointeur vers le moteur de jeu*
- [graphic](#) : :CGraphicsEngine * gixe
 - pointeur vers le moteur de rendu graphique*
- boost ::shared_ptr< [CGuiSystem](#) > gui
 - pointeur vers l'interface utilisateur*
- irr : :IrrlichtDevice * idevice
 - pointeur vers le device de irrlicht*
- CEGUI : :IrrlichtRenderEngine * irrRenderEngine
 - pointeur vers le moteur de rendu de l'interface utilisateur*
- int lastFPS
 - nombre d'images par seconde*
- [reseau](#) : :CNetworkEngine * ne
 - pointeur vers le moteur de réseau*
- [CGraphicsOptions](#) options
 - options graphique*
- CGame * parent
 - pointeur vers le parent*
- bool passive_mode
 - true si mode passif activé*
- [physic](#) : :CPhysicsEngine * phy
 - pointeur vers le moteur de physique*
- boost ::mutex queue_mutex
 - Mutex de protection de la liste d'évenement event_queue.*
- [audio](#) : :CSoundEngine * se
 - pointeur vers le moteur de son*
- boost ::shared_ptr< [CSceneGraph](#) > sg
 - pointeur vers les graph de scène*
- CEGUI : :System * sys
 - pointeur vers le system CEGUI*

7.34.1 Description détaillée

le moteur graphique

7.34.2 Documentation des énumérations membres

7.34.2.1 anonymous enum [inherited]

Constantes d'état.

Valeurs énumérées :

STATE NOTHING
STATE QUITTING
STATE WAITING PLAYERS
STATE PLAYING INTRO
STATE LOADING ROUND
STATE PLAYING

7.34.2.2 anonymous enum

liste des type de gestionnaire d'évenement

Valeurs énumérées :

ON_MENUS

ON_LOADING

ON_GAME

7.34.3 Documentation des constructeurs et destructeur

7.34.3.1 ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine (*core* ::CGame * *g*, *bool* *daemon*)

constructeur

Paramètres

in	<i>g</i>	la classe game parente
in	<i>daemon</i>	définit le mode passif

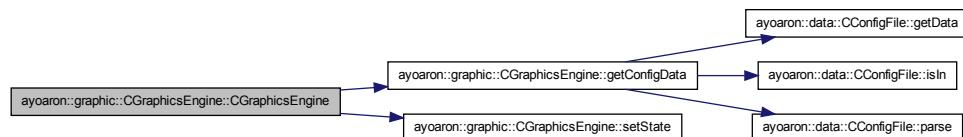
Cela crée les contextes Irrlicht & CEGUI gui_system est connecté à [CGraphicsEngine](#)

À faire

try to add the data directory to the filesystem or something like that

Références ayoaron ::graphic ::CGraphicsOptions ::bits, driver, ayoaron ::graphic ::CGraphicsOptions ::drv, ayoaron ::core ::CEngine ::engine_name, ayoaron ::graphic ::CGraphicsOptions ::fullscreen, getConfigData(), gui, idevice, irrRenderer, l_eh, lastFPS, ON_GAME, ON_MENUS, options, ayoaron ::core ::CEngine ::passive_mode, ayoaron ::graphic ::CGraphicsOptions ::resolution, setState(), sg, ayoaron ::graphic ::CGraphicsOptions ::shadows, et ayoaron ::graphic ::CGraphicsOptions ::Vsync.

Voici le graphe d'appel pour cette fonction :



7.34.3.2 ayoaron ::graphic ::CGraphicsEngine ::~CGraphicsEngine () [virtual]

destructeur

Références irrRenderer.

7.34.4 Documentation des fonctions membres

7.34.4.1 `void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e)`
[inline, inherited]

lien avec le moteur de jeu

Paramètres

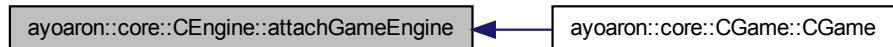
in	e	pointeur vers le moteur de jeu à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.34.4.2 `void ayoaron ::core ::CEngine ::attachGraphicsEngine (`
`graphic ::CGraphicsEngine * e)` [inline, inherited]

lien avec le moteur de rendu graphique

Paramètres

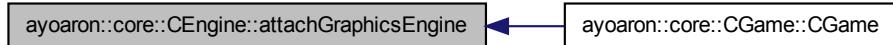
in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.34.4.3 void ayoaron ::core ::CEngine ::attachNetworkEngine (reseau ::CNetworkEngine * *e*) [inline, inherited]

lien avec le moteur de réseau

Paramètres

in	<i>e</i> pointeur vers le moteur de réseau à lier
----	---

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.34.4.4 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * *e*) [inline, inherited]

lien avec le moteur de physique

Paramètres

in	<i>e</i> pointeur vers le moteur de physique à lier
----	---

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références ayoaron ::core ::CEngine ::phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.34.4.5 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline, inherited]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.34.4.6 void ayoaron ::graphic ::CGraphicsEngine ::frame (double dt) [virtual]

Methode frame.

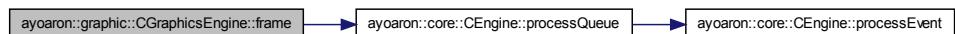
Cette méthode est utiliser pour mettre à jour le moteur de jeu
appelé à chaque frame

Implémente [ayoaron ::core ::CEngine](#).

Références driver, gui, idevice, lastFPS, ayoaron ::core ::CEngine ::parent, ayoaron ::core ::CEngine ::processQueue(), et sg.

Référencé par ayoaron ::core ::CGame ::run().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.34.4.7 void ayoaron ::graphic ::CGraphicsEngine ::getConfigData () [protected, virtual]

récupération de la configuration actuelle

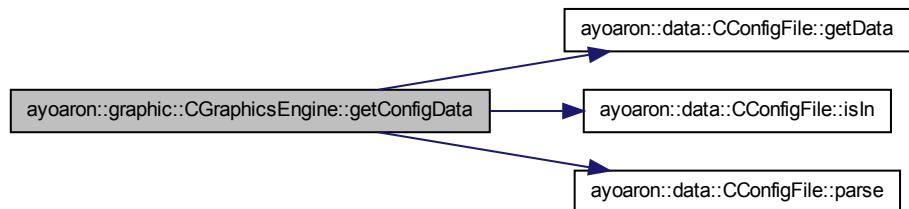
charge la config complète dans le moteur graphique

Implémente [ayoaron ::core ::CEngine](#).

Références ayoaron ::graphic ::CGraphicsOptions ::bits, ayoaron ::core ::CEngine ::cfile, ayoaron ::graphic ::CGraphicsOptions ::debuglvl, ayoaron ::graphic ::CGraphicsOptions ::drv, ayoaron ::graphic ::CGraphicsOptions ::fullscren, ayoaron ::data ::CConfigFile ::getData(), ayoaron ::data ::SConfigFileData ::i_data, ayoaron ::data ::CConfigFile ::isIn(), options, ayoaron ::data ::CConfigFile ::parse(), ayoaron ::graphic ::CGraphicsOptions ::resolution, ayoaron ::graphic ::CGraphicsOptions ::shadows, et ayoaron ::graphic ::CGraphicsOptions ::Vsync.

Référencé par CGraphicsEngine().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.34.4.8 int ayoaron ::graphic ::CGraphicsEngine ::getDbgLevel () [inline]

renvoie le niveau de debug

Références ayoaron ::graphic ::CGraphicsOptions ::debuglvl, et options.

Référencé par ayoaron ::graphic ::CGuiSystem ::displayGameGui(), ayoaron ::graphic ::CGuiSystem ::displayOptions(), et ayoaron ::objets ::CVaisseauX1 ::loadObjectData().

Voici le graphe d'appel pour cette fonction :



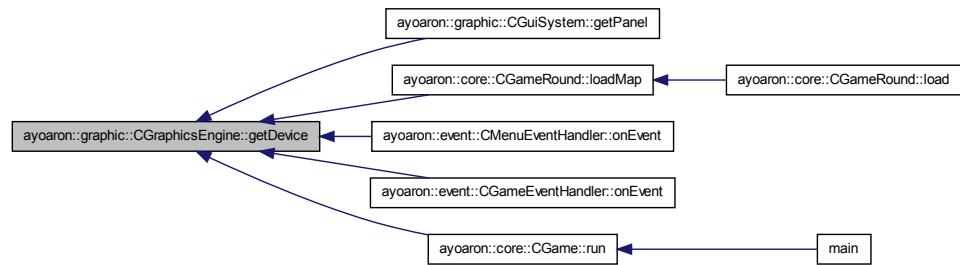
7.34.4.9 irr ::IrrlichtDevice* ayoaron ::graphic ::CGraphicsEngine ::getDevice () [inline]

revoie le device de irrlicht

Références idevice.

Référencé par ayoaron ::graphic ::CGuiSystem ::getPanel(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::event ::CMenuEventHandler ::onEvent(), ayoaron ::event ::CGameEventHandler ::onEvent(), et ayoaron ::core ::CGame ::run().

Voici le graphe d'appel pour cette fonction :



7.34.4.10 irr ::video ::IVideoDriver* ayoaron ::graphic ::CGraphicsEngine ::getDriver () [inline]

revoie le driver d'affichage

Références driver.

Référencé par ayoaron ::physic ::CObjet ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.34.4.11 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine () [inline, inherited]

lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect(), ayoaron ::graphic ::CGuiSystem ::handleProcessCommand(), ayoaron ::graphic ::CGuiSystem ::loadGame(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::event ::CGameEventHandler ::onEvent(), et processEvent().

Voici le graphe d'appel pour cette fonction :



7.34.4.12 `graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine () [inline, inherited]`

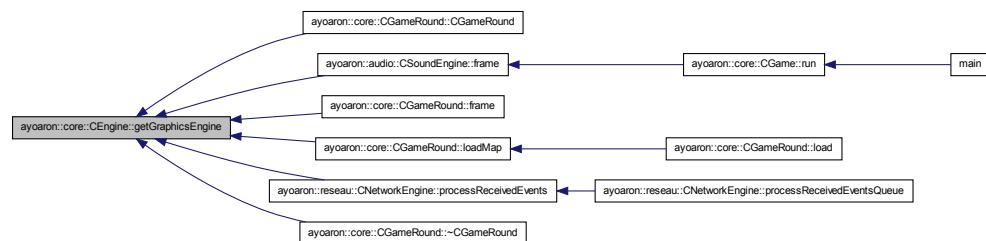
lien vers le moteur de rendu graphique

Cette méthode est utilisée pour obtenir un pointeur vers le moteur de rendu graphique

Références `ayoaron ::core ::CEngine ::gixe`.

Référencé par `ayoaron ::core ::CGameRound ::CGameRound()`, `ayoaron ::audio ::CSoundEngine ::frame()`, `ayoaron ::core ::CGameRound ::frame()`, `ayoaron ::core ::CGameRound ::loadMap()`, `ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents()`, et `ayoaron ::core ::CGameRound ::~CGameRound()`.

Voici le graphe d'appel pour cette fonction :



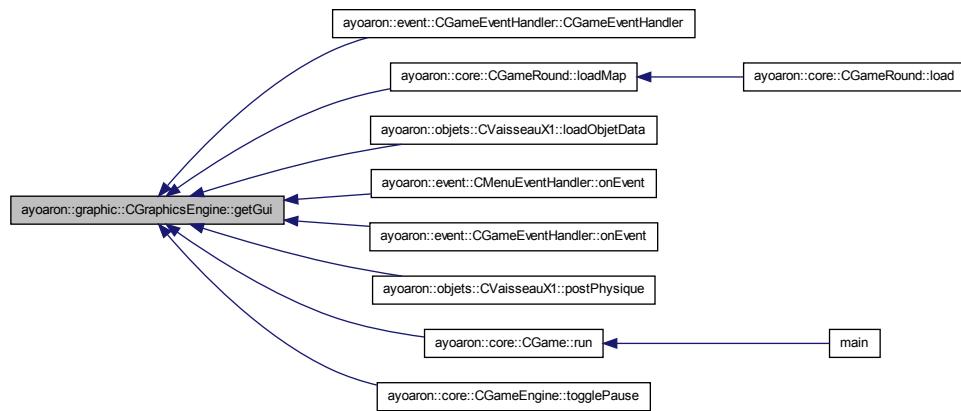
7.34.4.13 `boost ::shared_ptr<CGuiSystem> ayoaron ::graphic ::CGraphicsEngine ::getGui () [inline]`

revoie le système d'interface utilisateur

Références `gui`.

Référencé par `ayoaron ::event ::CGameEventHandler ::CGameEventHandler()`, `ayoaron ::core ::CGameRound ::loadMap()`, `ayoaron ::objets ::CVaisseauX1 ::loadObjetData()`, `ayoaron ::event ::CMENUEventHandler ::onEvent()`, `ayoaron ::event ::CGameEventHandler ::onEvent()`, `ayoaron ::objets ::CVaisseauX1 ::postPhysique()`, `ayoaron ::core ::CGame ::run()`, et `ayoaron ::core ::CGameEngine ::togglePause()`.

Voici le graphe d'appel pour cette fonction :



7.34.4.14 CEGUI::IrrlichtRenderer* ayoaron::graphic::CGraphicsEngine::getIrrRenderer ()
[inline]

revoie le moteur de rendu de l'interface utilisateur

Références irrRenderers.

7.34.4.15 **reseau** : **!CNetworkEngine*** ayoaron : **core : CEngine** : **getNetworkEngine ()**
[**inline, inherited**]

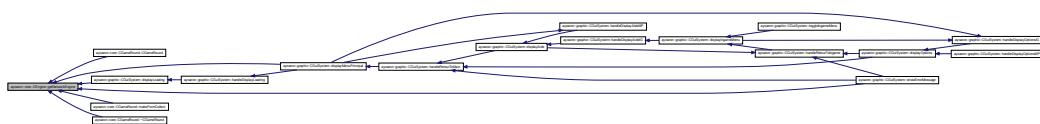
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.34.4.16 `CGame* ayoaron ::core ::CEngine ::getParent() [inline, inherited]`

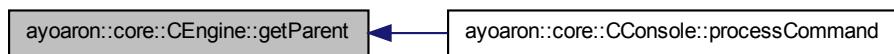
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :

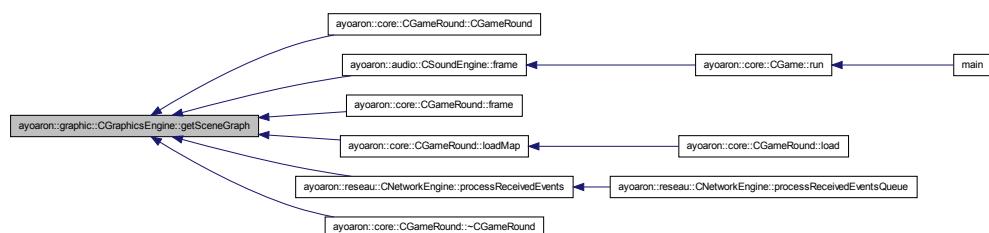
7.34.4.17 `boost ::shared_ptr<CSceneGraph> ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph() [inline]`

revoie le système de graph de scène

Références sg.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::core ::CGameRound ::frame(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :

7.34.4.18 `audio ::CSoundEngine* ayoaron ::core ::CEngine ::getSoundEngine() [inline, inherited]`

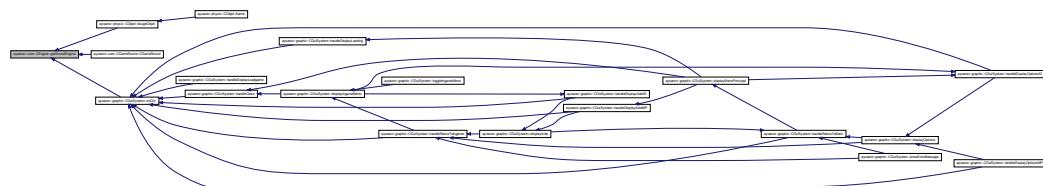
lien vers le moteur de son

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::core ::CGameRound ::CGameRound(), et ayoaron ::graphic ::CGuiSystem ::onClic().

Voici le graphe d'appel pour cette fonction :



7.34.4.19 void ayoaron ::graphic ::CGraphicsEngine ::processEvent (event ::CEngineEvent & e) [protected, virtual]

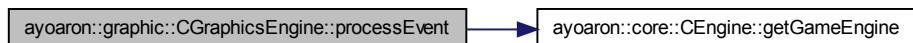
traitement d'un événement

Traite les événement arrivant au moteur graphique

Implémente [ayoaron ::core ::CEngine](#).

Références ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::event ::CEngineEvent ::LOAD, et ayoaron ::event ::CEngineEvent ::type.

Voici le graphe d'appel pour cette fonction :



7.34.4.20 void ayoaron ::core ::CEngine ::processQueue () [inherited]

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

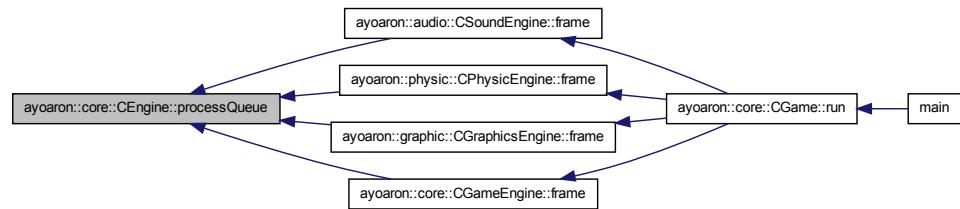
Références ayoaron ::core ::CEngine ::events_queue, ayoaron ::core ::CEngine ::processEvent(), et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::physic ::CPhysicsEngine ::frame(), frame(), et ayoaron ::core ::CGameEngine ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.34.4.21 void ayoaron ::core ::CEngine ::pushEvent (event ::CEngineEvent & e)
 [inherited]

ajout d'un événement

Paramètres

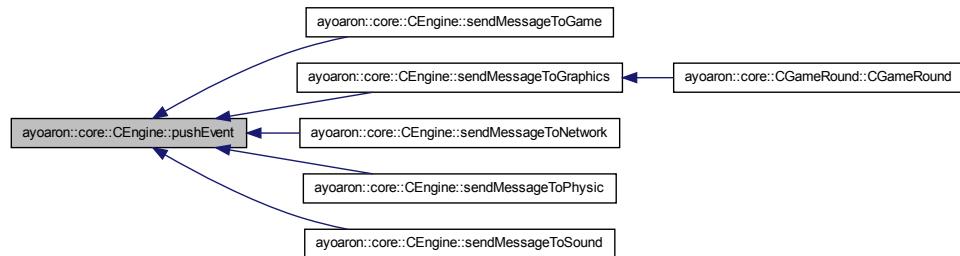
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références ayoaron ::core ::CEngine ::events_queue, et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par ayoaron ::core ::CEngine ::sendMessageToGame(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::core ::CEngine ::sendMessageToPhysic(), et ayoaron ::core ::CEngine ::sendMessageToSound().

Voici le graphe d'appel pour cette fonction :



7.34.4.22 void ayoaron ::graphic ::CGraphicsEngine ::requestClose ()

demande de fermeture

demande à tous les objet enfant de se terminer

Fonction à appelé en cas de fermeture du programme

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::graphic ::CGuiSystem ::close().

Voici le graphe d'appel pour cette fonction :



7.34.4.23 void ayoaron ::core ::CEngine ::sendMessageToGame (event : ::CEngineEvent & e) [inherited]

envoie un message au moteur de jeu

Paramètres

in	e	evenement à envoyer vers le moteur de jeu
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ayoaron ::core ::CEngine ::ge, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.34.4.24 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gffe, et ayoaron ::core ::CEngine ::pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.34.4.25 void ayoaron ::core ::CEngine ::sendMessageToNetwork(event ::CEngineEvent & e) [inherited]

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ayoaron ::core ::CEngine ::ne, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.34.4.26 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e) [inherited]

envoie un message au moteur physique

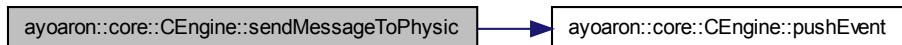
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de physique

Références ayoaron ::core ::CEngine ::phy, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.34.4.27 void ayoaron ::core ::CEngine ::sendMessageToSound (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de son

Références ayoaron ::core ::CEngine ::pushEvent(), et ayoaron ::core ::CEngine ::se.

Voici le graphe d'appel pour cette fonction :

**7.34.4.28 void ayoaron ::graphic ::CGraphicsEngine ::setState (int s)**

définit l'état

Paramètres

in	s	le nouvel état
----	---	----------------

Références idevice, l_eh, et ayoaron ::core ::CEngine ::state.

Référencé par CGraphicsEngine(), ayoaron ::graphic ::CGuiSystem ::loadGame(), et ayoaron ::core ::CGameRound ::loadMap().

Voici le graphe d'appel pour cette fonction :

**7.34.5 Documentation des données membres****7.34.5.1 data ::CConfigFile ayoaron ::core ::CEngine ::cfile [protected, inherited]**

Fichier de configuration.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::réseau ::CNetworkEngine ::getConfigData(), et getConfigData().

7.34.5.2 irr ::video ::IVideoDriver* ayoaron ::graphic ::CGraphicsEngine ::driver [protected]

pointeur vers le device video

Référencé par CGraphicsEngine(), frame(), et getDriver().

7.34.5.3 `std ::string ayoaron ::core ::CEngine ::engine_name` [protected, inherited]

nom de ce moteur

Référencé par ayoaron ::core ::CGameEngine ::CGameEngine(), CGraphicsEngine(), et ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine().

7.34.5.4 `std ::queue<event ::CEngineEvent> ayoaron ::core ::CEngine ::events_queue` [protected, inherited]

liste des événement à traiter

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.34.5.5 `CGameEngine* ayoaron ::core ::CEngine ::ge` [protected, inherited]

pointeur vers le moteur de jeu

Référencé par ayoaron ::core ::CEngine ::attachGameEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::core ::CEngine ::sendMessageToGame(), et ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive().

7.34.5.6 `graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::gixe` [protected, inherited]

pointeur vers le moteur de rendu graphique

Référencé par ayoaron ::core ::CEngine ::attachGraphicsEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), et ayoaron ::core ::CGameEngine ::togglePause().

7.34.5.7 `boost ::shared_ptr<CGuiSystem> ayoaron ::graphic ::CGraphicsEngine ::gui` [protected]

pointeur vers l'interface utilisateur

Référencé par CGraphicsEngine(), frame(), et getGui().

7.34.5.8 `irr ::IrrlichtDevice* ayoaron ::graphic ::CGraphicsEngine ::idevice` [protected]

pointeur vers le device de irrlicht

Référencé par CGraphicsEngine(), frame(), getDevice(), et setState().

7.34.5.9 **CEGUI : :IrrlichtRenderer* ayoaron : :graphic : :CGraphicsEngine : :IrrlichtRenderer** [protected]

pointeur vers le moteur de rendu de l'interface utilisateur

Référencé par CGraphicsEngine(), getIrrlichtRenderer(), et ~CGraphicsEngine().

7.34.5.10 **std : :map<int, boost : :shared_ptr<event : :CEventHandler> > ayoaron : :graphic : :CGraphicsEngine : :I_eh**

liste des gestionnaires d'évenement

Référencé par CGraphicsEngine(), ayoaron : :physic : :CSecteur : :frame(), et setState().

7.34.5.11 **std : :map<int, boost : :shared_ptr<irr : :scene : :IAnimatedMesh> > ayoaron : :graphic : :CGraphicsEngine : :I_meshs**

listes des maillages des objets

Référencé par ayoaron : :graphic : :CSceneGraph : :addNode().

7.34.5.12 **int ayoaron : :graphic : :CGraphicsEngine : :lastFPS** [protected]

nombre d'images par seconde

Référencé par CGraphicsEngine(), et frame().

7.34.5.13 **reseau : :CNetworkEngine* ayoaron : :core : :CEngine : :ne** [protected, inherited]

pointeur vers le moteur de réseau

Référencé par ayoaron : :core : :CEngine : :attachNetworkEngine(), ayoaron : :core : :CEngine : :CEngine(), ayoaron : :core : :CEngine : :getNetworkEngine(), ayoaron : :reseau : :CNetworkEngine : :sendEventTCP(), ayoaron : :reseau : :CNetworkEngine : :sendEventUDP(), ayoaron : :core : :CEngine : :sendMessageToNetwork(), ayoaron : :reseau : :CNetworkEngine : :serverThreadTCPReceive(), et ayoaron : :reseau : :CNetworkEngine : :udpAsyncRead().

7.34.5.14 **CGraphicsOptions ayoaron : :graphic : :CGraphicsEngine : :options** [protected]

options graphique

Référencé par CGraphicsEngine(), getConfigData(), et getDbgLevel().

7.34.5.15 **CGame* ayoaron : :core : :CEngine : :parent** [protected, inherited]

pointeur vers le parent

Référencé par frame(), ayoaron : :core : :CEngine : :getParent(), et requestClose().

7.34.5.16 **bool ayoaron ::core ::CEngine ::passive_mode** [protected, inherited]

true si mode passif activé

Référencé par ayoaron ::core ::CEngine ::CEngine(), CGraphicsEngine(), et ayoaron ::audio ::CSoundEngine ::CSoundEngine().

7.34.5.17 **physic ::CPhysicEngine* ayoaron ::core ::CEngine ::phy** [protected, inherited]

pointeur vers le moteur de physique

Référencé par ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::CEngine(), et ayoaron ::core ::CEngine ::sendMessageToPhysic().

7.34.5.18 **boost ::mutex ayoaron ::core ::CEngine ::queue_mutex** [protected, inherited]

Mutex de protection de la liste d'événement event_queue.

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.34.5.19 **audio ::CSoundEngine* ayoaron ::core ::CEngine ::se** [protected, inherited]

pointeur vers le moteur de son

Référencé par ayoaron ::core ::CEngine ::attachSoundEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getSoundEngine(), et ayoaron ::core ::CEngine ::sendMessageToSound().

7.34.5.20 **boost ::shared_ptr<CSceneGraph> ayoaron ::graphic ::CGraphicsEngine ::sg** [protected]

pointeur vers les graph de scène

Référencé par CGraphicsEngine(), frame(), et getSceneGraph().

7.34.5.21 **int ayoaron ::core ::CEngine ::state** [inherited]

l'état actuel

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::physic ::CSecteur ::frame(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::reseau ::CNetworkEngine ::multicastReceive(), setState(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.34.5.22 boost ::mutex ayoaron ::core ::CEngine ::state_mutex [inherited]

Mutex de protection de l'état.

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.34.5.23 CEGUI ::System* ayoaron ::graphic ::CGraphicsEngine ::sys [protected]

pointeur vers le system CEGUI

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

- [CGraphicsEngine.h](#)
- [CGraphicsEngine.cpp](#)

7.35 Référence de la classe ayoaron ::graphic ::CGraphicsOptions

résumé des options graphiques

```
#include <CGraphicsOptions.h>
```

Fonctions membres publiques

- [CGraphicsOptions \(\)](#)
constructeur
- [~CGraphicsOptions \(\)](#)
destructeur

Attributs publics

- [irr ::u32 bits](#)
nombre de bits de résolution
- [int debuglvl](#)
niveau de debug
- [irr ::video ::E_DRIVER_TYPE drv](#)
type de driver
- [bool fullscreen](#)
si plein écran
- [irr ::core ::dimension2d< irr ::u32 > resolution](#)
la résolution d'écran
- [bool shadows](#)
ombres dynamiques
- [bool Vsync](#)
Vsync.

7.35.1 Description détaillée

résumé des options graphiques

7.35.2 Documentation des constructeurs et destructeur

7.35.2.1 **ayoaron ::graphic ::CGraphicsOptions ::CGraphicsOptions()** [inline]

constructeur

7.35.2.2 **ayoaron ::graphic ::CGraphicsOptions ::~CGraphicsOptions()** [inline]

destructeur

7.35.3 Documentation des données membres

7.35.3.1 **irr ::u32 ayoaron ::graphic ::CGraphicsOptions ::bits**

nombre de bits de résolution

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.35.3.2 **int ayoaron ::graphic ::CGraphicsOptions ::debuglvl**

niveau de debug

Référencé par ayoaron ::graphic ::CGraphicsEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getDbgLevel().

7.35.3.3 **irr ::video ::E_DRIVER_TYPE ayoaron ::graphic ::CGraphicsOptions ::drv**

type de driver

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.35.3.4 **bool ayoaron ::graphic ::CGraphicsOptions ::fullscren**

si plein écran

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.35.3.5 **irr ::core ::dimension2d<irr ::u32> ayoaron ::graphic ::CGraphicsOptions ::resolution**

la résolution d'écran

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.35.3.6 **bool ayoaron ::graphic ::CGraphicsOptions ::shadows**

ombres dynamiques

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.35.3.7 bool ayoaron ::graphic ::CGraphicsOptions ::Vsync

Vsinc.

Référencé par ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

La documentation de cette classe a été générée à partir du fichier suivant :

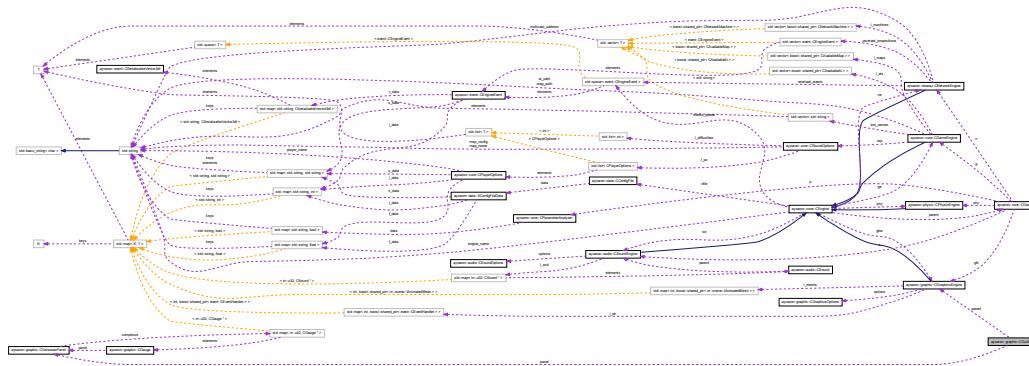
- [CGraphicsOptions.h](#)

7.36 Référence de la classe ayoaron ::graphic ::CGuiSystem

class traitant entièrement l'interface utilisateur (avec CEGUI) chaque événement déclenche un handle_XX , appelant une fonction XX

```
#include <CGuiSystem.h>
```

Graphe de collaboration de ayoaron ::graphic ::CGuiSystem :



Types publics

- enum {
 [MAIN_MENU](#), [OPTIONS_MENU](#), [INGAME_MENU](#), [LOAD_MENU](#),
[SAVE_MENU](#), [GAME](#) }
- liste des différents menus*

Fonctions membres publiques

- [CGuiSystem \(\)](#)
Constructeur.
- void [close \(\)](#)
ferme l'application
- void [displayGameGui \(\)](#)
affiche la console
- void [displayMenuPrincipal \(\)](#)
fonction d'affichage du menu1
- void [displaySelectPlayers \(\)](#)

- **CVaisseauPanel * getPanel ()**
renvoie le tableau de bord
- **bool handleClose (const CEGUI ::EventArgs &)**
handle vers la fermeture de l'application
- **bool handleDisplayAideIG (const CEGUI ::EventArgs &)**
handle d'affichage de l'aide depuis le menu en jeu
- **bool handleDisplayAideMP (const CEGUI ::EventArgs &)**
handle d'affichage de l'aide depuis le menu principal
- **bool handleDisplayLoadgame (const CEGUI ::EventArgs &)**
handle d'affichage du menu de chargement de sauvegarde
- **bool handleDisplayLoading (const CEGUI ::EventArgs &)**
handle vers le menu de chargement d'un jeu
- **bool handleDisplayOptionsIG (const CEGUI ::EventArgs &)**
handle d'affichage les options depuis le menu en jeu
- **bool handleDisplayOptionsMP (const CEGUI ::EventArgs &)**
handle d'affichage les options depuis le menu principal
- **bool handleProcessCommand (const CEGUI ::EventArgs &)**
handle de résolution des commandes
- **bool handleRetourTolngame (const CEGUI ::EventArgs &)**
handle d'affichage du menu en jeu
- **bool handleRetourToMain (const CEGUI ::EventArgs &)**
handle de retour au menu principal
- **bool isConsoleVisible (void)**
renvoie si la console est visible
- **bool isPanelVisible ()**
renvoie true si le panneau est visible
- **void loadGame ()**
charge un jeu
- **void render ()**
affiche l'interface
- **void setParent (CGraphicsEngine *)**
définit la graphics_engine parente
- **void showErrorMessage (std ::string, int)**
affiche le message d'erreur
- **void toggleConsole ()**
active ou désactive la console
- **void toggleIngameMenu ()**
lance l'affichage du menu 'in game'
- **void togglePanel ()**
change la visibilité du panneau
- **void txtVaiss (std ::string s)**
affiche un texte dans le gui du vaisseau
- **std ::string vectortoString (irr ::core ::vector3df a)**
transforme un vecteur en string pour etre afficher
- **~CGuiSystem ()**
Destructeur.

Fonctions membres privées

- **void displayAide (int)**
fonction d'affiche de la fenetre d'aide
- **void displayIngameMenu ()**
fonction d'affiche du menu en jeu
- **void displayLoadgame ()**
fonction d'affichage du jeu à charger
- **void displayLoading ()**
fonction d'affichage de l'écran de chargement

- void **displayOptions** (int)
fonction d'affichage de l'écran d'option
- std ::list< std ::string > **getMapList** ()
affiche la liste des maps
- void **onClic** ()
action par défaut à un click bouton (joue le son de clic)

Attributs privés

- bool **console_visible**
si l'on doit afficher la console
- bool **gaugepanel**
si l'on doit afficher le tableau de bord
- bool **ingame_menu**
définit si l'on est en mode menu in game
- **CVaisseauPanel * panel**
pointeur vers le tableau de bord
- **CGraphicsEngine * parent**
la graphics_engine parente

7.36.1 Description détaillée

class traitant entièrement l'interface utilisateur (avec CEGUI) chaque événement déclenche un handle_XX , appelant une fonction XX

7.36.2 Documentation des énumérations membres

7.36.2.1 anonymous enum

liste des différents menus

Valeurs énumérées :

MAIN_MENU
OPTIONS_MENU
INGAME_MENU
LOAD_MENU
SAVE_MENU
GAME

7.36.3 Documentation des constructeurs et destructeur

7.36.3.1 ayoaron ::graphic ::CGuiSystem ::CGuiSystem ()

Constructeur.

Références `console_visible`, `gaugepanel`, `ingame_menu`, et `panel`.

7.36.3.2 ayoaron ::graphic ::CGuiSystem ::~CGuiSystem ()

Destructeur.

7.36.4 Documentation des fonctions membres

7.36.4.1 void ayoaron ::graphic ::CGuiSystem ::close ()

ferme l'application

Références parent, et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

Référencé par handleClose().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.2 void ayoaron ::graphic ::CGuiSystem ::displayAide (int origine) [private]

fonction d'affiche de la fenêtre d'aide

À faire

gestion du ok

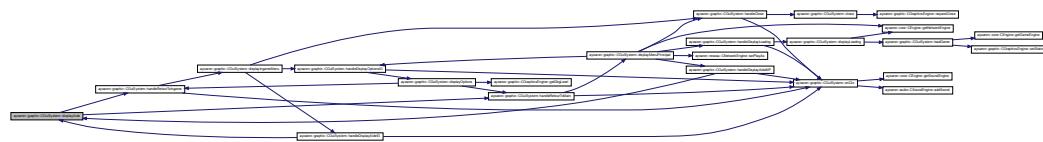
À faire

gestion du ok

Références handleRetourToIngame(), handleRetourToMain(), INGAME_MENU, et MAIN_MENU.

Référencé par handleDisplayAideIG(), et handleDisplayAideMP().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



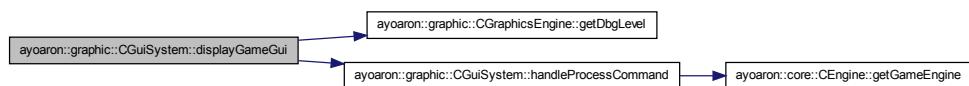
7.36.4.3 void ayoaron ::graphic ::CGuiSystem ::displayGameGui ()

affiche la console

Références console_visible, ayoaron ::graphic ::CGraphicsEngine ::getDbgLevel(), handleProcessCommand(), et parent.

Référencé par toggleIngameMenu().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.4 void ayoaron ::graphic ::CGuiSystem ::displayIngameMenu () [private]

fonction d'affiche du menu en jeu

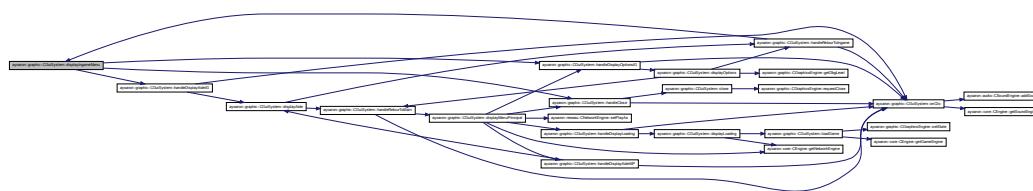
À faire

cabler les autres boutons

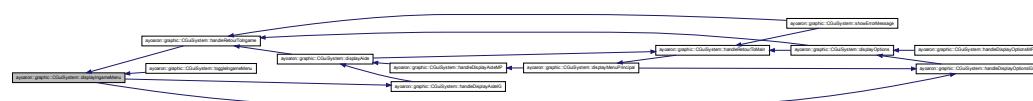
Références handleClose(), handleDisplayAideIG(), et handleDisplayOptionsIG().

Référencé par handleRetourTolngame(), et toggleIngameMenu().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.5 void ayoaron ::graphic ::CGuiSystem ::displayLoadgame () [private]

fonction d'affichage du jeu à charger

Référencé par handleDisplayLoadgame().

Voici le graphe d'appel pour cette fonction :



7.36.4.6 void ayoaron ::graphic ::CGuiSystem ::displayLoading () [private]

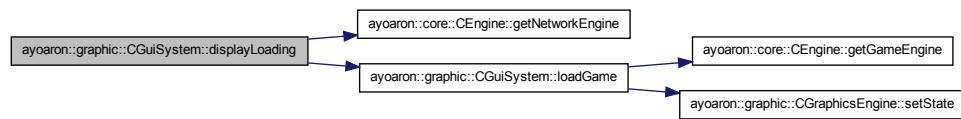
fonction d'affichage de l'écran de chargement

display the progress bar screen and calls the creation of the game round

Références ayoaron ::core ::CEngine ::getNetworkEngine(), loadGame(), parent, ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::STATE_LOADING_ROUND, et ayoaron ::core ::state_mutex.

Référencé par handleDisplayLoading().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.7 void ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal ()

fonction d'affichage du menu1

Affiche le menu principal

À faire

cabler les autres boutons

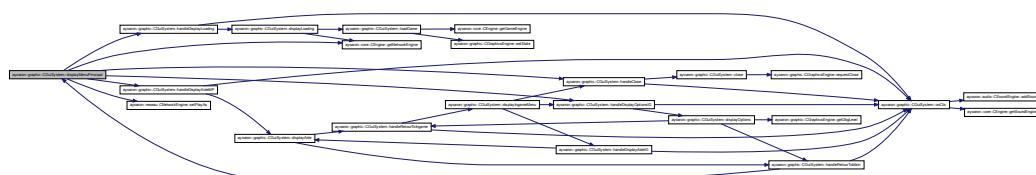
À faire

ajouter le menu de chargement

Références ayoaron ::core ::CEngine ::getNetworkEngine(), handleClose(), handleDisplayAideMP(), handleDisplayLoading(), handleDisplayOptionsIG(), ayoaron ::réseau ::CNetworkEngine ::NONE, parent, ayoaron ::réseau ::CNetworkEngine ::setPlayAs(), ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, et ayoaron ::core ::CEngine ::STATE NOTHING.

Référencé par handleRetourToMain().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.8 void ayoaron ::graphic ::CGuiSystem ::displayOptions (int origine) [private]

fonction d'affichage de l'écran d'option

À faire

rendre effective cette procédure

À faire

gestion des données d'options

À faire

gestion du ok

À faire

gestion du ok

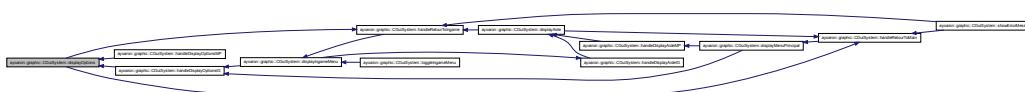
Références ayoaron : `:graphic` : `:CGraphicsEngine` : `:getDbgLevel()`, `handleRetourToln-game()`, `handleRetourToMain()`, `INGAME_MENU`, `MAIN_MENU`, et `parent`.

Référencé par handleDisplayOptionsIG(), et handleDisplayOptionsMP().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.9 void ayoaron ::graphic ::CGuiSystem ::displaySelectPlayers()

affiche les joueurs sélectionnés

7.36.4.10 `std ::list<std ::string> ayoaron ::graphic ::CGuiSystem ::getMapList ()`
 [private]

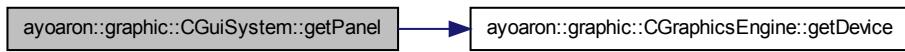
affiche la liste des maps

7.36.4.11 `CVaisseauPanel * ayoaron ::graphic ::CGuiSystem ::getPanel ()`

renvoie le tableau de bord

Références ayoaron ::graphic ::CGraphicsEngine ::getDevice(), panel, et parent.

Voici le graphe d'appel pour cette fonction :



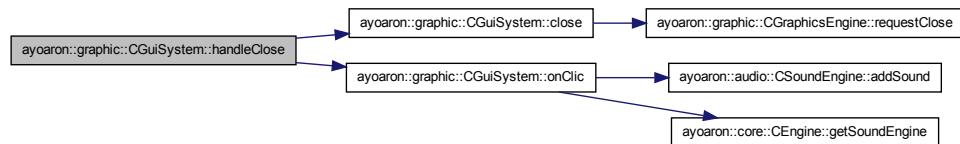
7.36.4.12 `bool ayoaron ::graphic ::CGuiSystem ::handleClose (const CEGUI ::EventArgs &)`
 [inline]

handle vers la fermeture de l'application

Références close(), et onClic().

Référencé par displayIngameMenu(), et displayMenuPrincipal().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



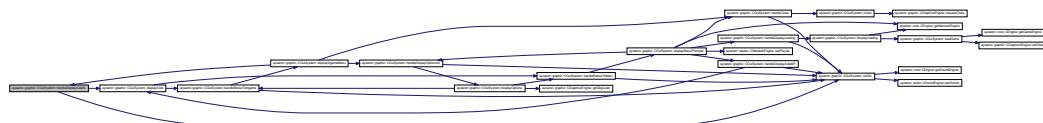
7.36.4.13 bool ayoaron ::graphic ::CGuiSystem ::handleDisplayAideIG (const CEGUI ::EventArgs &) [inline]

handle d'affichage de l'aide depuis le menu en jeu

Références displayAide(), INGAME_MENU, et onClic().

Référencé par displayIngameMenu().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



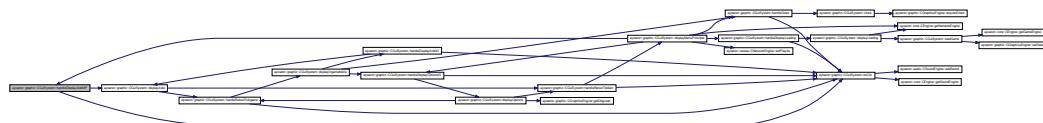
7.36.4.14 bool ayoaron ::graphic ::CGuiSystem ::handleDisplayAideMP (const CEGUI ::EventArgs &) [inline]

handle d'affichage de l'aide depuis le menu principal

Références displayAide(), MAIN_MENU, et onClic().

Référencé par displayMenuPrincipal().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

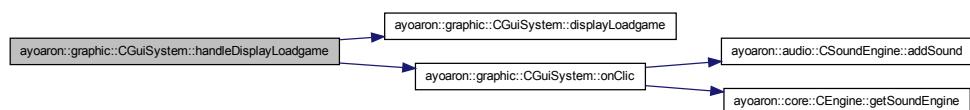


7.36.4.15 `bool ayoaron ::graphic ::CGuiSystem ::handleDisplayLoadgame (const CEGUI ::EventArgs &) [inline]`

handle d'affichage du menu de chargement de sauvegarde

Références `displayLoadgame()`, et `onClic()`.

Voici le graphe d'appel pour cette fonction :



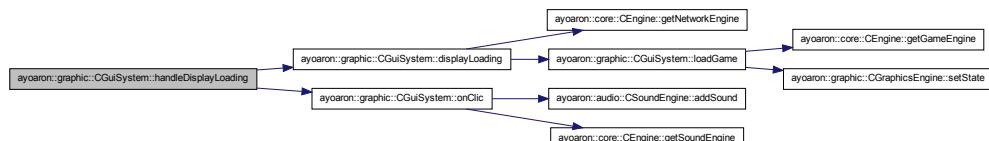
7.36.4.16 `bool ayoaron ::graphic ::CGuiSystem ::handleDisplayLoading (const CEGUI ::EventArgs &) [inline]`

handle vers le menu de chargement d'un jeu

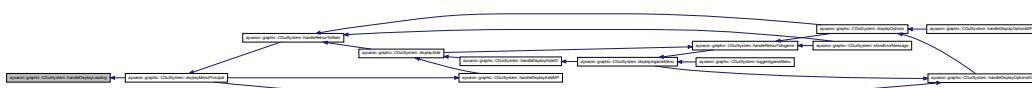
Références `displayLoading()`, et `onClic()`.

Référencé par `displayMenuPrincipal()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



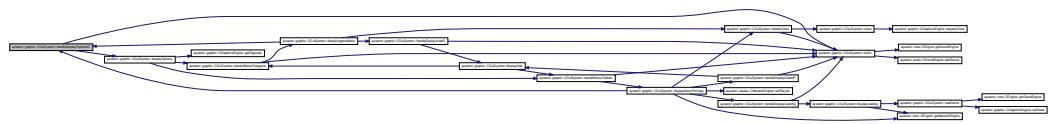
7.36.4.17 `bool ayoaron ::graphic ::CGuiSystem ::handleDisplayOptionsIG (const CEGUI ::EventArgs &) [inline]`

handle d'affichage les options depuis le menu en jeu

Références `displayOptions()`, `MAIN_MENU`, et `onClic()`.

Référencé par `displayIngameMenu()`, et `displayMenuPrincipal()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

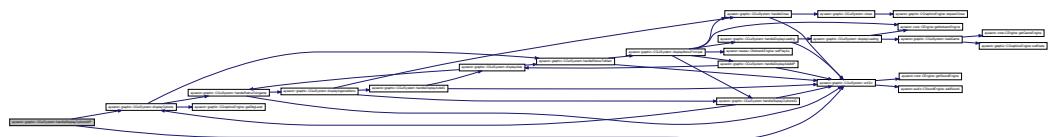


7.36.4.18 bool ayoaron ::graphic ::CGuiSystem ::handleDisplayOptionsMP (const CEGUI ::EventArgs &) [inline]

handle d'affichage les options depuis le menu principal

Références `displayOptions()`, `MAIN_MENU`, et `onClic()`.

Voici le graphe d'appel pour cette fonction :



7.36.4.19 bool ayoaron ::graphic ::CGuiSystem ::handleProcessCommand (const CEGUI ::EventArgs &)

handle de résolution des commandes

Références `ayoaron ::core ::CEngine ::getGameEngine()`, et `parent`.

Référencé par `displayGameGui()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



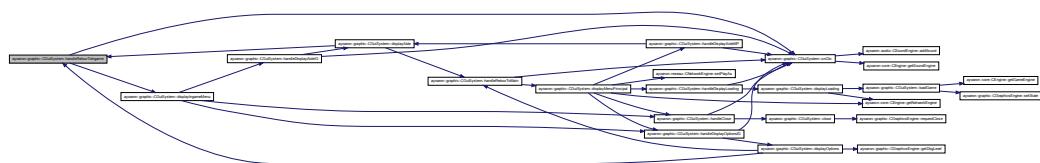
7.36.4.20 `bool ayoaron ::graphic ::CGuiSystem ::handleRetourToIngame (const CEGUI ::EventArgs &) [inline]`

handle d'affichage du menu enjeu

Références `displayIngameMenu()`, et `onClic()`.

Référencé par `displayAide()`, `displayOptions()`, et `showErrorMessage()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



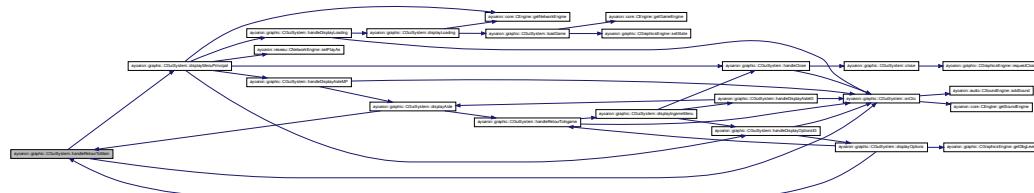
7.36.4.21 `bool ayoaron ::graphic ::CGuiSystem ::handleRetourToMain (const CEGUI ::EventArgs &) [inline]`

handle de retour au menu principal

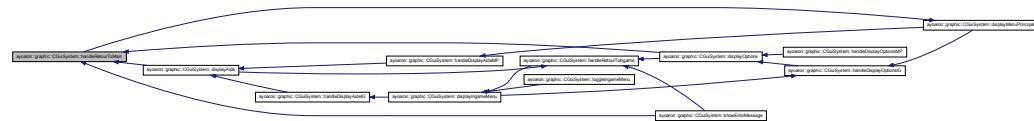
Références displayMenuPrincipal(), et onClic().

Référencé par displayAide(), displayOptions(), et showErrorMessage().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.22 bool ayoaron ::graphic ::CGuiSystem ::isConsoleVisible (void) [inline]

renvoie si la console est visible

*

Références console_visible.

7.36.4.23 bool ayoaron ::graphic ::CGuiSystem ::isPanelVisible () [inline]

renvoie true si le panneau est visible

Références gaugepanel.

7.36.4.24 void ayoaron ::graphic ::CGuiSystem ::loadGame ()

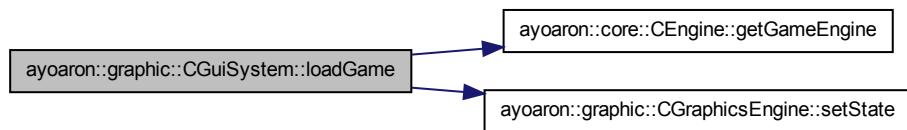
charge un jeu

thread function that loads the `game_round`

Références `ayoaron ::core ::CEngine ::getGameEngine()`, `ayoaron ::graphic ::CGraphicsEngine ::ON_LOADING`, parent, et `ayoaron ::graphic ::CGraphicsEngine ::setState()`.

Référencé par `displayLoading()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



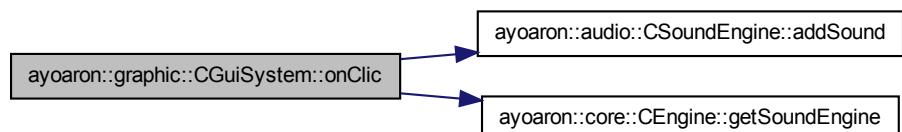
7.36.4.25 void ayoaron ::graphic ::CGuiSystem ::onClic () [private]

action par défaut à un click bouton (joue le son de clic)

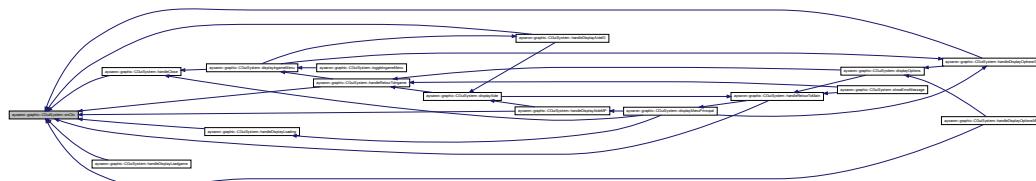
Références ayoaron ::audio ::CSoundEngine ::addSound(), ayoaron ::core ::CEngine ::getSoundEngine(), parent, et ayoaron ::audio ::SND_2D_INSTANT.

Référencé par handleClose(), handleDisplayAideIG(), handleDisplayAideMP(), handleDisplayLoadgame(), handleDisplayLoading(), handleDisplayOptionsIG(), handleDisplayOptionsMP(), handleRetourToIngame(), et handleRetourToMain().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.36.4.26 void ayoaron ::graphic ::CGuiSystem ::render ()

affiche l'interface

thread safe render function it displays the GUI

Références ayoaron ::graphic ::CVaisseauPanel ::drawAll(), gaugepanel, ayoaron ::core ::CShared ::get(), et panel.

Voici le graphe d'appel pour cette fonction :



7.36.4.27 void ayoaron ::graphic ::CGuiSystem ::setParent (CGraphicsEngine * p)

définit la graphics_engine parente

Paramètres

<i>p</i>	la graphics_engine parente
----------	----------------------------

Références parent.

7.36.4.28 void ayoaron ::graphic ::CGuiSystem ::showErrorMessage (std ::string msg, int screen)

affiche le message d'erreur

Paramètres

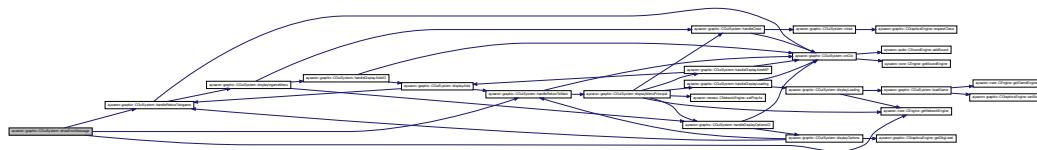
in	<i>msg</i>	le message à afficher
in	<i>screen</i>	le nuléro du menu où afficher

crée une fenetre popup d'erreur

basic screen showing a message and OK button clicking on OK returns to the screen whose number is given on parameter

Références ayoaron ::core ::CEngine ::getNetworkEngine(), handleRetourToIngame(), handleRetourToMain(), INGAME_MENU, MAIN_MENU, parent, ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, et ayoaron ::core ::CEngine ::STATE NOTHING.

Voici le graphe d'appel pour cette fonction :



7.36.4.29 void ayoaron ::graphic ::CGuiSystem ::toggleConsole ()

active ou désactive la console

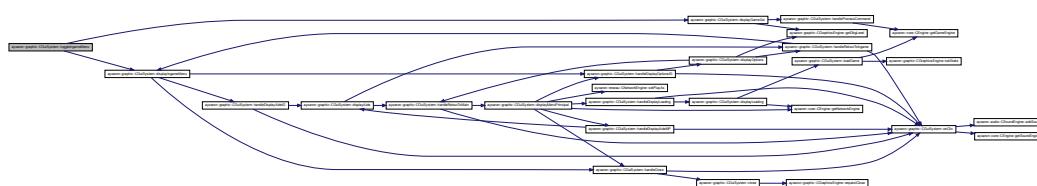
Références console_visible.

7.36.4.30 void ayoaron ::graphic ::CGuiSystem ::toggleIngameMenu ()

lance l'affichage du menu 'in game'

Références displayGameGui(), displayIngameMenu(), et ingame_menu.

Voici le graphe d'appel pour cette fonction :



7.36.4.31 void ayoaron ::graphic ::CGuiSystem ::togglePanel () [inline]

change la visibilité du panneau

Références gaugepanel.

7.36.4.32 void ayoaron ::graphic ::CGuiSystem ::txtVaiss (std ::string s)

affiche un texte dans le gui du vaisseau

7.36.4.33 std ::string ayoaron ::graphic ::CGuiSystem ::vectortoString (irr ::core ::vector3df a)

transforme un vecteur en string pour etre afficher

7.36.5 Documentation des données membres

7.36.5.1 `bool ayoaron ::graphic ::CGuiSystem ::console_visible [private]`

si l'on doit afficher la console

Référencé par CGuiSystem(), displayGameGui(), isConsoleVisible(), et toggleConsole().

7.36.5.2 `bool ayoaron ::graphic ::CGuiSystem ::gaugepanel [private]`

si l'on doit afficher le tableau de bord

Référencé par CGuiSystem(), isPanelVisible(), render(), et togglePanel().

7.36.5.3 `bool ayoaron ::graphic ::CGuiSystem ::ingame_menu [private]`

definit si l'on est en mode menu in game

Référencé par CGuiSystem(), et toggleIngameMenu().

7.36.5.4 `CVaisseauPanel* ayoaron ::graphic ::CGuiSystem ::panel [private]`

pointeur vers le tableau de bord

Référencé par CGuiSystem(), getPanel(), et render().

7.36.5.5 `CGraphicsEngine* ayoaron ::graphic ::CGuiSystem ::parent [private]`

la graphics_engine parente

Référencé par close(), displayGameGui(), displayLoading(), displayMenuPrincipal(), displayOptions(), getPanel(), handleProcessCommand(), loadGame(), onClic(), setParent(), et showErrorMessage().

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

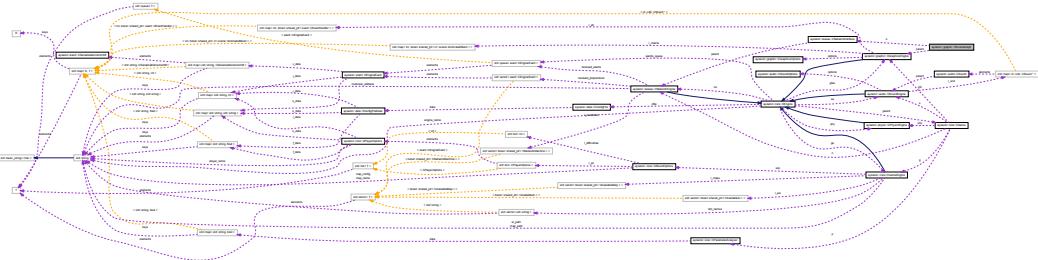
- [CGuiSystem.h](#)
- [CGuiSystem.cpp](#)

7.37 Référence de la classe ayoaron ::graphic ::CSceneGraph

le graph de scène

```
#include <CSceneGraph.h>
```

Graphe de collaboration de ayoaron ::graphic ::CSceneGraph :



Fonctions membres publiques

- void **addNode** (int id, irr ::core ::vector3df &pos, irr ::core ::vector3df &rot)
ajoute un élément
- void **addNodeAndPropagate** (int, irr ::core ::vector3df &, irr ::core ::vector3df &, int)
ajoute un élément et propage
- **CSceneGraph** (CGraphicsEngine *ge)
constructeur
- void **moveNode** (irr ::s32 id, irr ::core ::vector3df &pos, irr ::core ::vector3df &rot)
bouge un élément
- void **moveNodeAndPropagate** (int id, irr ::core ::vector3df &pos, irr ::core ::vector3df &rot, int expeditor)
bouge un élément et propage
- void **removeNode** (int id)
supprime un élément
- void **removeNodeAndPropagate** (int, int)
supprime un élément et propage
- void **removeObserver** ()
supprime l'observateur
- void **render** ()
calcul du rendu
- void **setObserver** (reseau ::CNetworkInterface *obs)
définit l'observateur
- void **tourneCamera** (irr ::core ::vector3df incr)
Translate la caméra.
- void **translateCamera** (irr ::core ::vector3df incr)
Translate la caméra.
- **~CSceneGraph** ()
destructeur

Attributs publics

- irr ::scene ::ISceneManager * **smgr**
manager de scène
- boost ::mutex **smgr_mutex**
mutex vers le manager de scène

Attributs privés

- **reseau** : `CNetworkInterface * o`
interface réseau
- **CGraphicsEngine * parent**
le moteur graphique parent

7.37.1 Description détaillée

le graph de scène

7.37.2 Documentation des constructeurs et destructeur**7.37.2.1 ayoaron ::graphic ::CSceneGraph ::CSceneGraph (CGraphicsEngine * ge)**

constructeur

Paramètres

in	ge	le moteur graphique parent
----	----	----------------------------

Références o.

7.37.2.2 ayoaron ::graphic ::CSceneGraph ::~CSceneGraph ()

destructeur

Références smgr, et smgr_mutex.

7.37.3 Documentation des fonctions membres**7.37.3.1 void ayoaron ::graphic ::CSceneGraph ::addNode (int id, irr ::core ::vector3df & pos, irr ::core ::vector3df & rot)**

ajoute un élément

Paramètres

in	id	id de l'objet à ajouter
in	pos	le vecteur position
in	rot	le vecteur rotation

ajoute l'élément de numéro id et mets en position pos & rot l'élément est copié depuis la base de donnée, stockée dans le moteur graphique

Références ayoaron ::graphic ::CGraphicsEngine ::l_meshs, parent, smgr, et smgr_mutex.

Référencé par addNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



**7.37.3.2 void ayoaron ::graphic ::CSceneGraph ::addNodeAndPropagate (int *id*,
irr ::core ::vector3df & *pos*, irr ::core ::vector3df & *rot*, int *expeditor*)**

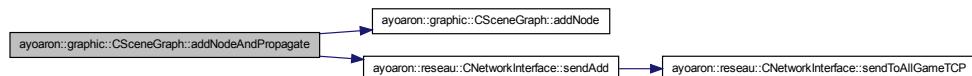
ajoute un élément et propage

Paramètres

in	<i>id</i>	id de l'élément à ajouter
in	<i>pos</i>	le vecteur position
in	<i>rot</i>	le vecteur rotation
in	<i>expeditor</i>	l'expéditeur

Références addNode(), o, et ayoaron ::reseau ::CNetworkInterface ::sendAdd().

Voici le graphe d'appel pour cette fonction :



**7.37.3.3 void ayoaron ::graphic ::CSceneGraph ::moveNode (irr ::s32 *id*, irr ::core ::vector3df
& *pos*, irr ::core ::vector3df & *rot*)**

bouge un élément

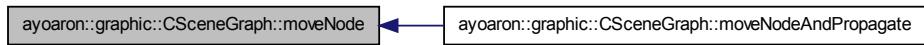
Paramètres

in	<i>id</i>	id de l'élément à bouger
in	<i>pos</i>	le vecteur position
in	<i>rot</i>	le vecteur rotation

Références smgr, et smgr_mutex.

Référencé par moveNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



7.37.3.4 `void ayoaron ::graphic ::CSceneGraph ::moveNodeAndPropagate (int id, irr ::core ::vector3df & pos, irr ::core ::vector3df & rot, int expeditor)`

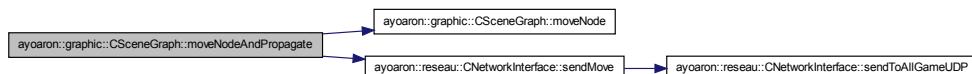
bouge un élément et propage

Paramètres

in	<i>id</i>	id de l'élément à bouger
in	<i>pos</i>	le vecteur position
in	<i>rot</i>	le vecteur rotation
in	<i>expeditor</i>	l'expéditeur

Références `moveNode()`, `o`, et `ayooron ::reseau ::CNetworkInterface ::sendMove()`.

Voici le graphe d'appel pour cette fonction :



7.37.3.5 `void ayoaron ::graphic ::CSceneGraph ::removeNode (int id)`

supprime un élément

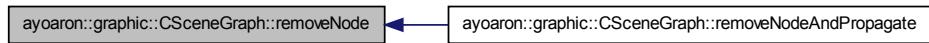
Paramètres

in	<i>id</i>	id de l'élément à supprimer
----	-----------	-----------------------------

Références `smgr`, et `smgr_mutex`.

Référencé par `removeNodeAndPropagate()`.

Voici le graphe d'appel pour cette fonction :



7.37.3.6 void ayoaron ::graphic ::CSceneGraph ::removeNodeAndPropagate (int *id*, int *expeditor*)

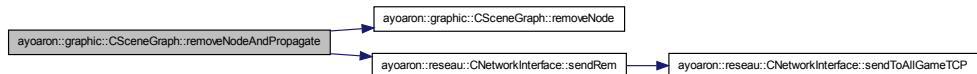
supprime un élément et propage

Paramètres

in	<i>id</i>	id de l'élément à supprimer
in	<i>expeditor</i>	l'expéditeur

Références o, removeNode(), et ayoaron ::reseau ::CNetworkInterface ::sendRem().

Voici le graphe d'appel pour cette fonction :



7.37.3.7 void ayoaron ::graphic ::CSceneGraph ::removeObserver ()

supprime l'observateur

Références o.

7.37.3.8 void ayoaron ::graphic ::CSceneGraph ::render ()

calcul du rendu

Références smgr, et smgr_mutex.

7.37.3.9 void ayoaron ::graphic ::CSceneGraph ::setObserver (reseau ::CNetworkInterface * *obs*)

définit l'observateur

Paramètres

in	<i>obs</i>	l'interface observée
----	------------	----------------------

Références o.

7.37.3.10 void ayoaron ::graphic ::CSceneGraph ::tourneCamera (irr ::core ::vector3df *incr*)

Translate la caméra.

Paramètres

in	<i>incr</i>	le vecteur de rotation
----	-------------	------------------------

Références smgr, et smgr_mutex.

7.37.3.11 void ayoaron ::graphic ::CSceneGraph ::translateCamera (irr ::core ::vector3df *incr*)

Translate la caméra.

Paramètres

in	<i>incr</i>	le vecteur de mouvement relatif (x : avance z : droite vers gauche z : bas en haut)
----	-------------	--

Références smgr, et smgr_mutex.

7.37.4 Documentation des données membres

7.37.4.1 **reseau** : CNetworkInterface* ayoaron ::graphic ::CSceneGraph ::
[private]

interface réseau

Référencé par addNodeAndPropagate(), CSceneGraph(), moveNodeAndPropagate(),
removeNodeAndPropagate(), removeObserver(), et setObserver().

7.37.4.2 CGraphicsEngine* ayoaron ::graphic ::CSceneGraph ::parent
[private]

le moteur graphique parent

Référencé par addNode().

7.37.4.3 irr ::scene ::ISceneManager* ayoaron ::graphic ::CSceneGraph ::smgr

manager de scène

Référencé par addNode(), moveNode(), removeNode(), render(), tourneCamera(), trans-
lateCamera(), et ~CSceneGraph().

7.37.4.4 boost ::mutex ayoaron ::graphic ::CSceneGraph ::smgr_mutex

mutex vers le manager de scène

Référencé par addNode(), moveNode(), removeNode(), render(), tourneCamera(), trans-
lateCamera(), et ~CSceneGraph().

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

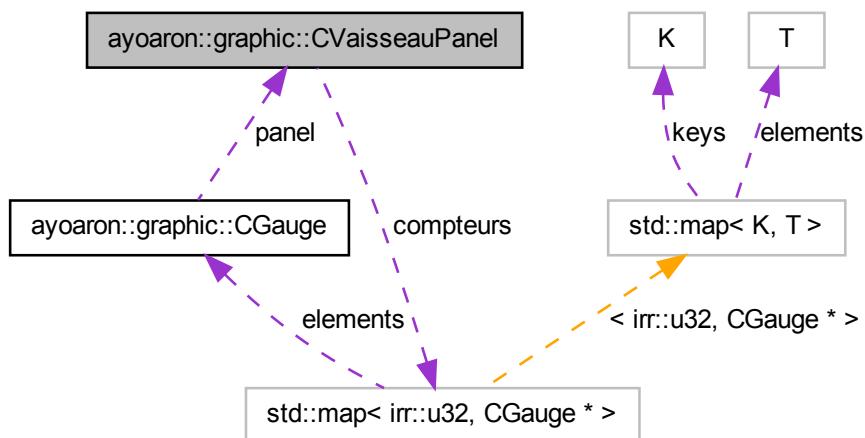
- [CSceneGraph.h](#)
- [CSceneGraph.cpp](#)

7.38 Référence de la classe ayoaron ::graphic ::CVaisseauPanel

une classe englobante permettant le dessin complet du tableau de bord

```
#include <CVaisseauPanel.h>
```

Graphe de collaboration de ayoaron ::graphic ::CVaisseauPanel :



Fonctions membres publiques

- `irr::u32 addGauge (CGauge *g)`
ajoute un compteur au HUD
- `irr::u32 addGauge (std::string backgr, irr::core::vector2d<irr::s32> pos=irr::core::vector2d<irr::s32>(0, 0), irr::core::vector2df scale=irr::core::vector2df(0, 0))`
cree et ajoute un compteur
- `CVaisseauPanel (irrlichtDevice *d)`
Constructeur.
- `void draw2DImage (irr::video::ITexture *tex, irr::core::position2d<irr::s32> position=irr::core::position2d<irr::s32>(0, 0), irr::f32 rotation=0, irr::core::position2d<irr::s32> rotationPoint=irr::core::position2d<irr::s32>(0, 0), irr::core::vector2df scale=irr::core::vector2df(0.0, 0.0), irr::core::rect<irr::s32> sourceRect=irr::core::rect<irr::s32>(0, 0, -1, -1), bool useAlphaChannel=true, irr::video::SColor color=irr::video::SColor(255, 255, 255, 255))`
dessinne une image
- `void drawAll ()`

- dessine l'intégralité du tableau de bord
- **CGauge * getGauge (irr ::u32 id)**
renvoie un pointeur vers le compteur
- irr ::core ::vector2df **getGlobalScale ()**
l'échelle globale
- irr ::video ::IVideoDriver * **getVideoDriver ()**
pour avoir le driver video
- void **setBackground (std ::string back)**
définit l'image de fond
- void **setOriginalSize (irr ::core ::vector2d< irr ::u32 > orisize)**
définit la taille originale du tableau de bord
- virtual **~CVaisseauPanel ()**
destructeur

Fonctions membres privées

- void **drawBackground ()**

Attributs privés

- irr ::video ::ITexture * **background**
nom de l'image de fond
- std ::map< irr ::u32, **CGauge * > compteurs**
les compteurs
- irr ::IrrlichtDevice * **device**
pointeur vers le device d'irrlicht
- irr ::core ::vector2df **scale**
le facteur d'échelle

7.38.1 Description détaillée

une classe englobante permettant le dessin complet du tableau de bord

7.38.2 Documentation des constructeurs et destructeur

7.38.2.1 ayoaron ::graphic ::CVaisseauPanel ::CVaisseauPanel (irr ::IrrlichtDevice * d)

Constructeur.

Références background, et device.

7.38.2.2 ayoaron ::graphic ::CVaisseauPanel ::~CVaisseauPanel () [virtual]

destructeur

Références compteurs.

7.38.3 Documentation des fonctions membres

7.38.3.1 irr : :u32 ayoaron ::graphic ::CVaisseauPanel ::addGauge (CGauge * g)

ajoute un compteur au HUD

Paramètres

<i>g</i>	le compteur à ajouter
----------	-----------------------

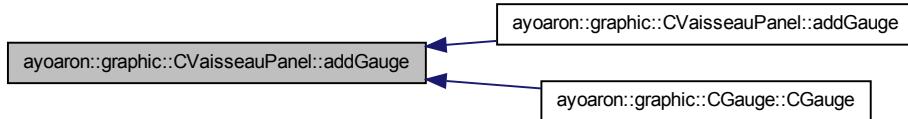
Renvoie

revoie le numéro dans la base

Références compteurs.

Référencé par addGauge(), et ayoaron ::graphic ::CGauge ::CGauge().

Voici le graphe d'appel pour cette fonction :

7.38.3.2 irr : :u32 ayoaron ::graphic ::CVaisseauPanel ::addGauge (std : :string *backgr*, irr : :core : :vector2d< irr : :s32 > *pos* = irr : :core : :vector2d< irr : :s32 > (0, 0), irr : :core : :vector2df *scale* = irr : :core : :vector2df (0, 0))

cree et ajoute un compteur

Paramètres

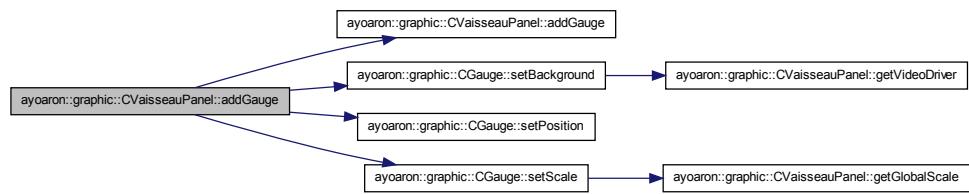
<i>backgr</i>	l'image de fond du compteur
<i>pos</i>	la position du compteur
<i>scale</i>	l'échelle du compteur

Renvoie

retourne le numéro dans la base

Références addGauge(), scale, ayoaron ::graphic ::CGauge ::setBackground(), ayoaron ::graphic ::CGauge ::setPosition(), et ayoaron ::graphic ::CGauge ::setScale().

Voici le graphe d'appel pour cette fonction :



```

7.38.3.3 void ayoaron ::graphic ::CVaisseauPanel ::draw2DImage (
    irr ::video ::ITexture * tex, irr ::core ::position2d< irr ::s32 >
    position = irr ::core ::position2d< irr ::s32 > (0, 0),
    irr ::f32 rotation = 0, irr ::core ::position2d< irr ::s32 > rotationPoint =
    irr ::core ::position2d< irr ::s32 > (0, 0), irr ::core ::vector2df
    scale = irr ::core ::vector2df(0.0, 0.0), irr ::core ::rect< irr ::s32
    > sourceRect = irr ::core ::rect< irr ::s32 > (0, 0, -1, -1),
    bool useAlphaChannel = true, irr ::video ::SColor color =
    irr ::video ::SColor(255, 255, 255, 255) )
  
```

dessinne une image

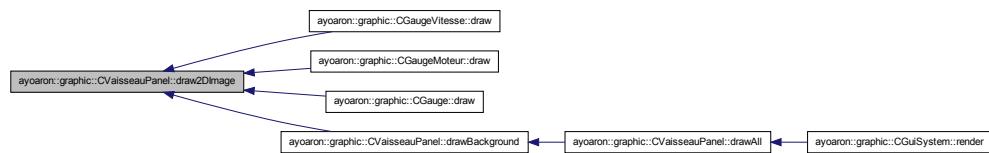
Paramètres

<i>tex</i>	L'image à dessiner
<i>position</i>	la position du coin supérieur gauche
<i>rotation</i>	l'angle de rotation
<i>rotationPoint</i>	la position du point de rotation
<i>scale</i>	L'échelle de dessin si l'échelle est 0, on utilise l'échelle globale
<i>sourceRect</i>	le rectangle de l'image source à dessiner si les valeurs sont négative, c'est l'ensemble de l'image qui est dessiner
<i>useAlpha- Channel</i>	si true, alors on utilise le canal alpha pour la transparence
<i>color</i>	la couleur de rendu (blanc pour un rendu tel que l'original)

Références device, et scale.

Référencé par ayoaron ::graphic ::CGaugeVitesse ::draw(), ayoaron ::graphic ::CGaugeMoteur ::draw(), ayoaron ::graphic ::CGauge ::draw(), et drawBackground().

Voici le graphe d'appel pour cette fonction :



7.38.3.4 void ayoaron ::graphic ::CVaisseauPanel ::drawAll ()

dessine l'intégralité du tableau de bord

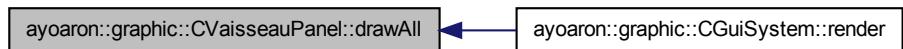
Références compteurs, et drawBackground().

Référencé par ayoaron ::graphic ::CGuiSystem ::render().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.38.3.5 void ayoaron ::graphic ::CVaisseauPanel ::drawBackground () [private]

dessine l'image de fond

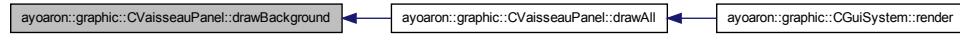
Références background, et draw2DImage().

Référencé par drawAll().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.38.3.6 CGauge * ayoaron ::graphic ::CVaisseauPanel ::getGauge (irr ::u32 id)

renvoie un pointeur vers le compteur

Paramètres

<i>id</i> numero du compteur à trouver
--

Renvoie

un pointeur sur le compteur selectionné, renvoie NULL s'il n'existe pas

Références compteurs.

7.38.3.7 irr ::core ::vector2df ayoaron ::graphic ::CVaisseauPanel ::getGlobalScale () [inline]

l'échelle globale

Renvoie

renvoie l'échelle globale du panneau

Références scale.

Référencé par ayoaron ::graphic ::CGauge ::setScale().

Voici le graphe d'appel pour cette fonction :



7.38.3.8 `irr ::video ::IVideoDriver* ayoaron ::graphic ::CVaisseauPanel ::getVideoDriver()`
 [inline]

pour avoir le driver video

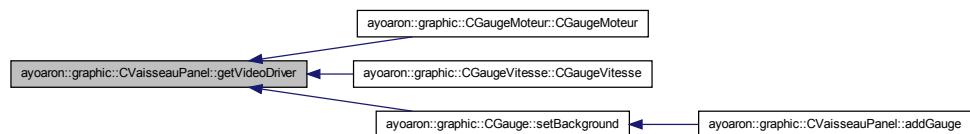
Renvoie

renvoie le driver video courrant

Références device.

Référencé par ayoaron ::graphic ::CGaugeMoteur ::CGaugeMoteur(), ayoaron ::graphic ::CGaugeVitesse ::CGaugeVitesse(), et ayoaron ::graphic ::CGauge ::setBackground().

Voici le graphe d'appel pour cette fonction :



7.38.3.9 `void ayoaron ::graphic ::CVaisseauPanel ::setBackground (std ::string back)`

definit l'image de fond

Paramètres

in	back	le nom de l'image de fond
----	------	---------------------------

Références background, device, et setOriginalSize().

Voici le graphe d'appel pour cette fonction :



7.38.3.10 `void ayoaron ::graphic ::CVaisseauPanel ::setOriginalSize (irr ::core ::vector2d< irr ::u32 > orisize)`

définit la taille originale du tableau de bord

Paramètres

in	orisize	la taille originale du tableau de bord. cette taille est généralement définie comme la taille de l'image de fond.
----	---------	---

Références device, et scale.

Référencé par setBackground().

Voici le graphe d'appel pour cette fonction :

**7.38.4 Documentation des données membres**

7.38.4.1 irr ::video ::ITexture* ayoaron ::graphic ::CVaisseauPanel ::background
[private]

nom de l'image de fond

Référencé par CVaisseauPanel(), drawBackground(), et setBackground().

7.38.4.2 std ::map<irr ::u32,CGauge*> ayoaron ::graphic ::CVaisseauPanel ::compteurs
[private]

les compteurs

Référencé par addGauge(), drawAll(), getGauge(), et ~CVaisseauPanel().

7.38.4.3 irr ::IrrlichtDevice* ayoaron ::graphic ::CVaisseauPanel ::device
[private]

pointeur vers le device d'irrlicht

Référencé par CVaisseauPanel(), draw2DImage(), getVideoDriver(), setBackground(), et setOriginalSize().

7.38.4.4 irr ::core ::vector2df ayoaron ::graphic ::CVaisseauPanel ::scale
[private]

le facteur d'ecchelle

Référencé par addGauge(), draw2DImage(), getGlobalScale(), et setOriginalSize().

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

- [CVaisseauPanel.h](#)
- [CVaisseauPanel.cpp](#)

7.39 Référence de la structure ayoaron ::graphic ::datamot

```
#include <CGaugeMoteur.h>
```

Attributs publics

- irr ::core ::rect< irr ::s32 > **cadre1**
le rectangle d'affichage poussee positive
- irr ::core ::rect< irr ::s32 > **cadre2**
le rectangle d'affichage poussee negative
- bool **reversible**
si le moteur est reversible
- irr ::f64 **valmot**
le régime actuel

7.39.1 Documentation des données membres

7.39.1.1 irr ::core ::rect<irr ::s32> ayoaron ::graphic ::datamot ::cadre1

le rectangle d'affichage poussee positive

7.39.1.2 irr ::core ::rect<irr ::s32> ayoaron ::graphic ::datamot ::cadre2

le rectangle d'affichage poussee negative

7.39.1.3 bool ayoaron ::graphic ::datamot ::reversible

si le moteur est reversible

7.39.1.4 irr ::f64 ayoaron ::graphic ::datamot ::valmot

le régime actuel

La documentation de cette structure a été générée à partir du fichier suivant :

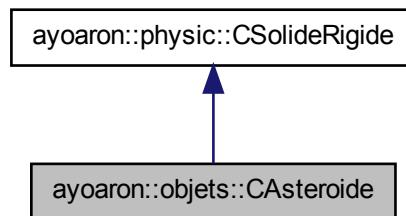
- [CGaugeMoteur.h](#)

7.40 Référence de la classe ayoaron ::objets ::CAsteroide

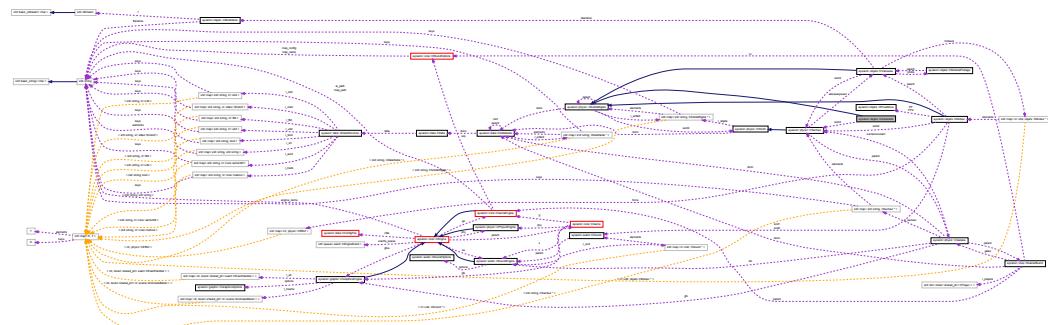
Une classe de description d'un astéroïde.

```
#include <CAsteroide.h>
```

Graphe d'héritage de ayoaron ::objets ::CAsteroid :



Graphe de collaboration de ayoaron ::objets ::CAsteroid :



Fonctions membres publiques

- void **addAbsoluteEffort** (CEffort F)
ajoute un effort exprimé dans le repère global
- void **addEnfant** (std ::string nom, CSolideRigide *enfant)
ajoute un nouvel enfant (s'il n'existe pas déjà)
- void **addLocalEffort** (CEffort F)
ajoute un effort exprimé dans le repère local
- void **addRelativeEffort** (CEffort F)
ajoute un effort exprimé dans le repère du parent
- CAsteroid (data : CDataNode *d, physic : CSecteur *p, physic : CSolideRigide *po=NULL, irr : s32 ID=-1)
Constructeur.
- void **dropEnfant** (std ::string nom)
supprime un enfant de la liste
- void **frame** (double dt)
effectue les diverses opération de mise à jour
- irr : core : vector3df **getAbsoluteForceResultante** ()
renvoie la force résultante exprimée dans le repère global

- irr : :core ::vector3df **getAbsoluteMomentResultante** ()

renvoie le moment résultant exprimé dans le repère global
- irr : :core ::vector3df **getAbsolutePosition** ()

renvoie la position de l'objet dans le monde
- irr : :core ::vector3df **getAbsoluteRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- irr : :core ::matrix4 **getAbsoluteRotation** ()

renvoie la matrice de rotation entre le repère global et le repère local
- irr : :core ::vector3df **getAbsoluteVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- **data** : :CDataNode * **getDonn** ()

acces aux données
- CSolideRigide * **getEnfant** (std ::string nom)

renvoie l'enfant donné
- irr : :core ::matrix4 **getGlobalInertie** ()

renvoie la matrice d'inertie
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantBegin** ()

renvoie un itérator sur les enfant pointant au début de la map
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantEnd** ()

renvoie un itérator sur les enfant pointant à la fin de la map
- irr : :core ::vector3df **getLocalForceResultante** ()

renvoie la force résultante exprimée dans le repère local
- irr : :core ::matrix4 **getLocalInertie** ()

renvoie la matrice d'inertie
- irr : :core ::vector3df **getLocalMomentResultante** ()

renvoie le moment résultant exprimé dans le repère local
- irr : :core ::vector3df **getLocalRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère local
- irr : :core ::vector3df **getLocalVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère local
- irr : :f64 **getMasse** ()

fonction qui renvoie la masse du solide
- irr : :u32 **getNbEnfant** ()

renvoie le nombre d'enfants de cet objet
- irr : :scene ::ISceneNode * **getNode** ()

renvoie le scene node et le maillage associé
- irr : :core ::vector3df **getRelativeForceResultante** ()

renvoie la force résultante exprimée dans le repère du parent
- irr : :core ::matrix4 **getRelativeInertie** ()

renvoie la matrice d'inertie
- irr : :core ::vector3df **getRelativeMomentResultante** ()

renvoie le moment résultant exprimé dans le repère du parent
- irr : :core ::vector3df **getRelativePosition** ()

renvoie la position de l'objet par rapport à l'objet parent
- irr : :core ::vector3df **getRelativeRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère du parent
- irr : :core ::matrix4 **getRelativeRotation** ()

renvoie la matrice de rotation entre le repère relatif et le repère local
- irr : :core ::vector3df **getRelativeVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère du parent
- void **loadObjetData** (data : :CDataNode *d=NULL)

charge l'objet depuis un fichier
- virtual void **postPhysique** (double dt)

effectue les diverses opération de mise à jour après le calcul de la physique
- virtual void **prePhysique** (double dt)

effectue les diverses opération de mise à jour avant de calculer la physique
- void **resetEffort** ()

- `void setAbsoluteRitesse (irr ::core ::vector3df vit)`
 - remet les efforts à zéro*
 - définit la vitesse de rotation du solide (tient compte de la vitesse d'entraînement du repère parent)*
- `void setAbsoluteVitesse (irr ::core ::vector3df vit)`
 - définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entraînement du repère parent)*
- `void setLocalRitesse (irr ::core ::vector3df vit)`
 - définit la vitesse de rotation du solide*
- `void setLocalVitesse (irr ::core ::vector3df vit)`
 - définit la vitesse du solide*
- `void setParent (CSolideRigide *par)`
 - change le parent de cet objet*
- `void setRelativePosition (irr ::core ::vector3df pos)`
 - définit la position de l'objet par rapport à son parent*
- `void setRelativeRitesse (irr ::core ::vector3df vit)`
 - définit la vitesse de rotation du solide*
- `void setRelativeRotation (irr ::core ::vector3df rot)`
 - à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.*
- `void setRelativeVitesse (irr ::core ::vector3df vit)`
 - définit la vitesse du solide*
- `virtual ~CAsteroid ()`
 - destructeur*

Fonctions membres protégées

- `irr ::core ::matrix4 getMatrix3 (std ::string nom, REPÈRE ou, REPÈRE reperestockage)`
 - renvoie le vecteur donné dans la base voulue*
- `irr ::core ::vector3df getVector (std ::string nom, REPÈRE ou, REPÈRE reperestockage)`
 - renvoie le vecteur donné dans la base voulue*
- `void setMatrix3 (std ::string nom, REPÈRE source, irr ::core ::matrix4 mat, REPÈRE reperestockage)`
 - fonction permettant le changement de repère des matrices*
- `void setVector (std ::string nom, REPÈRE source, irr ::core ::vector3df vec, REPÈRE reperestockage)`
 - fonction permettant le changement de repère des vecteurs*

Attributs protégés

- `irr ::scene ::ICameraSceneNode * cam`
 - pointeur vers le node de camera*
- `irr ::core ::vector3df caml`
 - direction du regard de la caméra*
- `irr ::core ::vector3df camp`
 - position de la caméra dans le repère local*
- `irr ::core ::vector3df camup`
 - vecteur de la verticale de la caméra*
- `data ::CDataNode * donn`
 - pointeur vers la structure de donnée*
- `irr ::core ::vector3df forceResultante`
 - le vecteur de la résultante des efforts, il est exprimé dans le repère local*
- `bool hascamera`
 - interrupteur de caméra*
- `std ::map< std ::string, CSolideRigide * > l_enfant`

- liste des objets enfants*
- irr : core ::vector3df **momentResultant**
le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)
 - irr : scene ::ISceneNode * **node**
pointeur vers le maillage et le scene node
 - CSolideRigide * **parent**
pointeur vers l'objet parent
 - double **srho**
masse volumique
 - physic ::CSecteur * **world**
pointeur vers le secteur parent

7.40.1 Description détaillée

Une classe de description d'un astéroïde.

Un astéroïde est un objet très simple composé d'un maillage fixe. On assimile la forme de l'astéroïde à l'ellipsoïde dans le moteur physique

7.40.2 Documentation des constructeurs et destructeur

7.40.2.1 ayoaron ::objets ::CAsteroide ::CAsteroide (data ::CDataNode * *d*,
physic ::CSecteur * *p*, physic ::CSolideRigide * *po* = NULL, irr ::s32 *ID* =
-1)

Constructeur.

7.40.2.2 ayoaron ::objets ::CAsteroide ::~CAsteroide () [virtual]

destructeur

7.40.3 Documentation des fonctions membres

7.40.3.1 void ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort (CEffort *F*)
[inherited]

ajoute un effort exprimé dans le repère global

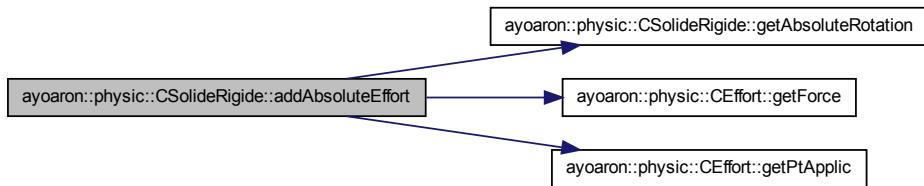
Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère global, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.40.3.2 void ayoaron ::physic ::CSolideRigide ::addEnfant (std ::string *nom*,
CSolideRigide * *enfant*) [inherited]**

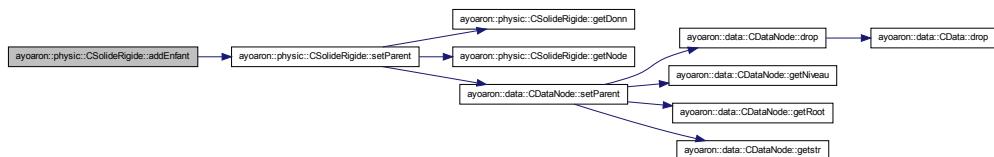
ajoute un nouvel enfant (s'il n'existe pas déjà)

Paramètres

<i>nom</i>	le nom (unique) de l'enfant
<i>enfant</i>	pointeur vers l'enfant en question

Références ayoaron ::physic ::CSolideRigide ::l_enfant, et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.40.3.3 void ayoaron ::physic ::CSolideRigide ::addLocalEffort (CEffort *F*)
 [inherited]

ajoute un effort exprimé dans le repère local

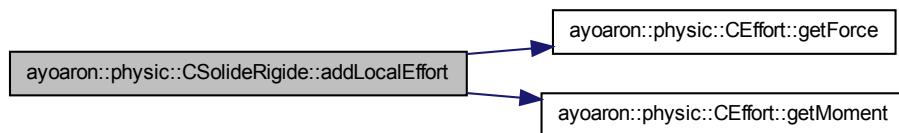
Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère local, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getMoment(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par ayoaron ::objets ::CMoteur ::prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.4 void ayoaron ::physic ::CSolideRigide ::addRelativeEffort (CEffort *F*)
 [inherited]

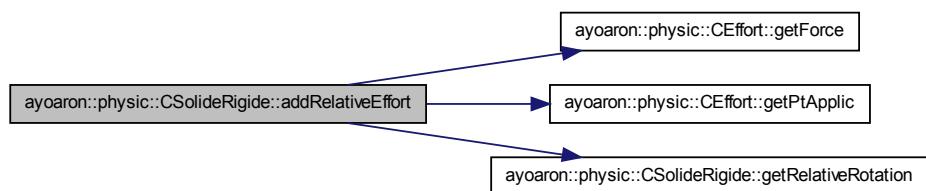
ajoute un effort exprimé dans le repère du parent

Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère du parent, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron : :physic : :CSolideRigide : :forceResultante, ayoaron : :physic : :CEffort : :getForce(), ayoaron : :physic : :CEffort : :getPtApplic(), ayoaron : :physic : :CSolideRigide : :getRelativeRotation(), et ayoaron : :physic : :CSolideRigide : :momentResultant.

Voici le graphe d'appel pour cette fonction :



7.40.3.5 `void ayoaron ::physic ::CSolideRigide ::dropEnfant (std ::string nom)`
[inherited]

supprime un enfant de la liste

Paramètres

nom nom de l'enfant à supprimer

7.40.3.6 void ayoaron ::physic ::CSolideBigide ::frame(double *dt*) [inherited]

effectue les diverses opérations de mise à jour

Paramètres

paramètres

la frame :

tout d'abord l'effort global

on pré-frame tous les objets enfants

la translation calculée dans le repère du parent

la rotation calculée dans le repère parent

on post frame tous les objets enfants

Béférances avoaron : physic : CSolideRigide

sic ::CSolideRigide ::cam, ayoaron ::physic ::

sic : :C:SolidRigide : :camp, aycaron : :physic : :

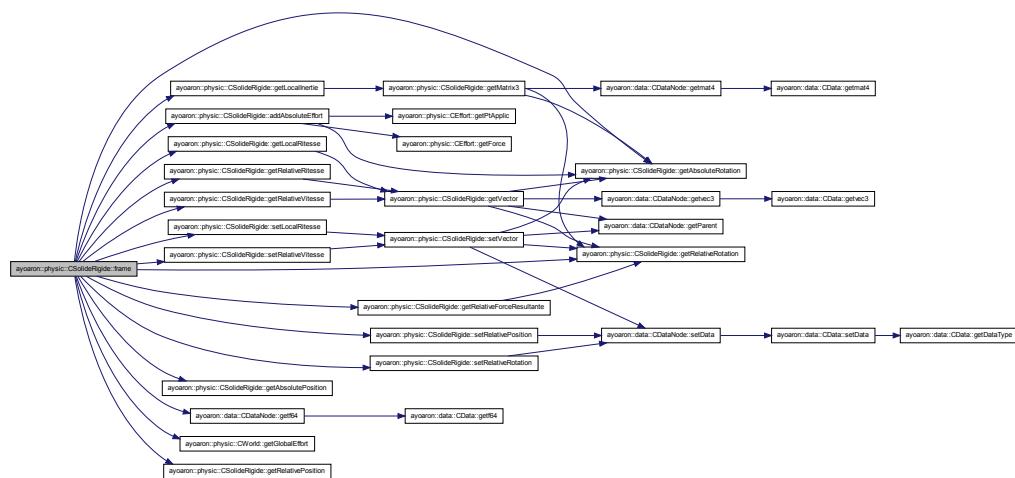
ron : :physic : :CSolideRigide : :getAbsolutePosition(), ayoaron : :physic : :CSolideRigide : :getAbsoluteRotation(), ayoaron : :data : :CDataNode : :getf64(), ayoaron : :phys-

2016 RELEASE UNDER E.O. 14176

Généré le Wed Nov 30 2011 14:30:11 pour Ayoaron par Doxygen

sic :: CWorld ::getGlobalEffort(), ayoaron :: physic :: CSolideRigide ::getLocalInertie(), ayoaron :: physic :: CSolideRigide ::getLocalRitesse(), ayoaron :: physic :: CSolideRigide ::getRelativeForceResultante(), ayoaron :: physic :: CSolideRigide ::getRelativePosition(), ayoaron :: physic :: CSolideRigide ::getRelativeRitesse(), ayoaron :: physic :: CSolideRigide ::getRelativeRotation(), ayoaron :: physic :: CSolideRigide ::getRelativeVitesse(), ayoaron :: physic :: CSolideRigide ::hascamera, ayoaron :: physic :: CSolideRigide ::l_enfant, ayoaron :: physic :: CSolideRigide ::momentResultant, ayoaron :: physic :: CSolideRigide ::node, ayoaron :: physic :: CSolideRigide ::setLocalRitesse(), ayoaron :: physic :: CSolideRigide ::setRelativePosition(), ayoaron :: physic :: CSolideRigide ::setRelativeRotation(), ayoaron :: physic :: CSolideRigide ::setRelativeVitesse(), et ayoaron :: physic :: CSolideRigide ::world.

Voici le graphe d'appel pour cette fonction :



7.40.3.7 irr :: core :: vector3df ayoaron :: physic :: CSolideRigide ::getAbsoluteForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère global

Renvoie

la résultante des efforts exprimée dans le référentiel global

Références ayoaron :: physic :: CSolideRigide ::forceResultante, et ayoaron :: physic :: CSolideRigide ::getAbsoluteRotation().

Voici le graphe d'appel pour cette fonction :



7.40.3.8 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère global

Renvoie

le moment résultant exprimé dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.40.3.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsolutePosition () [inherited]

renvoie la position de l'objet dans le monde

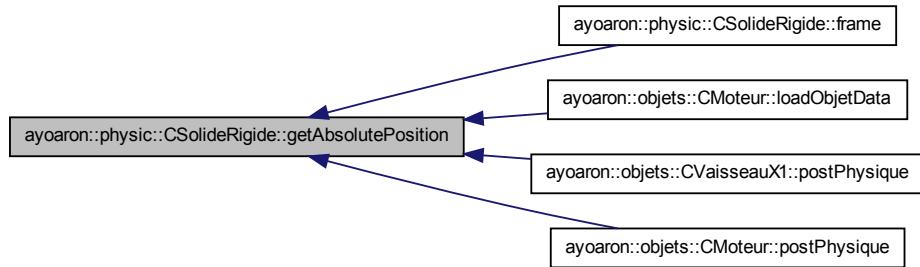
Renvoie

le vecteur position absolue de l'objet

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::objets ::CMoteur ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.40.3.10 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse ()
 [inherited]

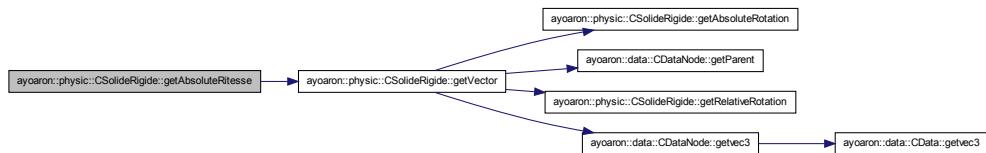
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse de rotation dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.40.3.11 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation ()
 [inherited]

renvoie la matrice de rotation entre le repère global et le repère local

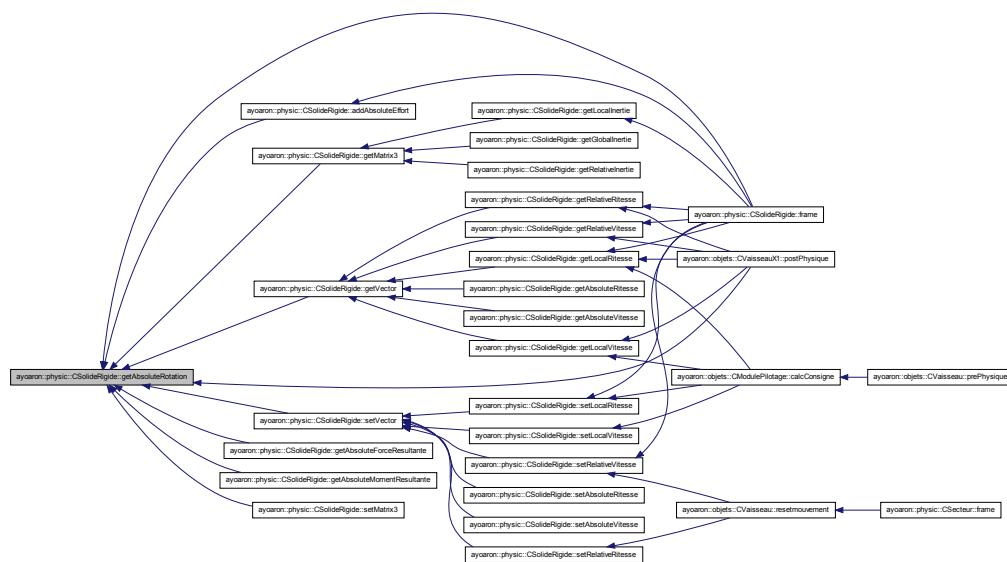
Renvoie

la matrice de rotation du repère global au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.40.3.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse () [inherited]

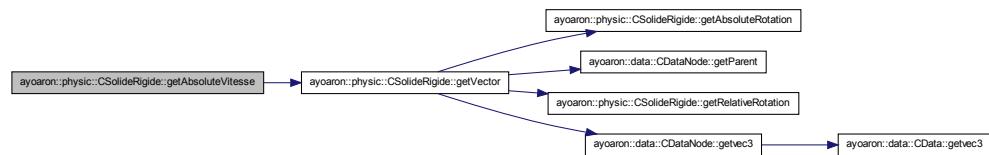
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.40.3.13 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::getDonn ()
 [inline, inherited]

acces aux données

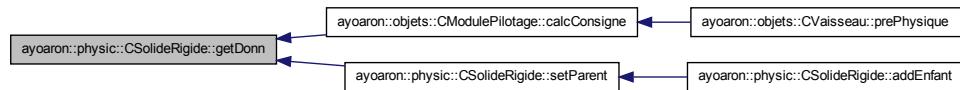
Renvoie

pointeur vers la structure de donnée

Références ayoaron ::physic ::CSolideRigide ::donn.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.40.3.14 CSolideRigide * ayoaron ::physic ::CSolideRigide ::getEnfant (std ::string nom)
 [inherited]

renvoie l'enfant donné

Paramètres

nom	le nom de l'enfant voulu
-----	--------------------------

Renvoie

pointeur vers l'enfant désiré, renvoie NULL si l'enfant n'existe pas

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.40.3.15 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getGlobalInertie ()
[inline, inherited]

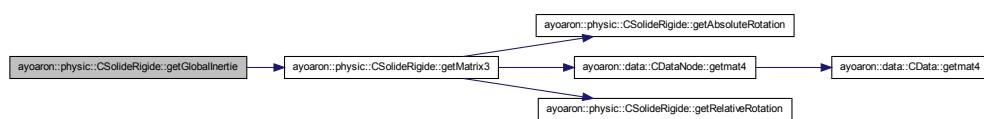
renvoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Voici le graphe d'appel pour cette fonction :



7.40.3.16 std ::map< std ::string, CSolideRigide * > ::iterator
ayooron ::physic ::CSolideRigide ::getIteratorEnfantBegin () [inherited]

renvoie un itérатор sur les enfant pointant au début de la map

Renvoie

iterator pointant au début de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.40.3.17 std ::map< std ::string, CSolideRigide * > ::iterator
ayooron ::physic ::CSolideRigide ::getIteratorEnfantEnd () [inherited]

renvoie un itérатор sur les enfant pointant à la fin de la map

Renvoie

iterator pointant à la fin de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.40.3.18 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère local

Renvoie

la résultante des efforts exprimée dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::forceResultante.

7.40.3.19 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getLocalInertie ()
[inline, inherited]

revoie la matrice d'inertie

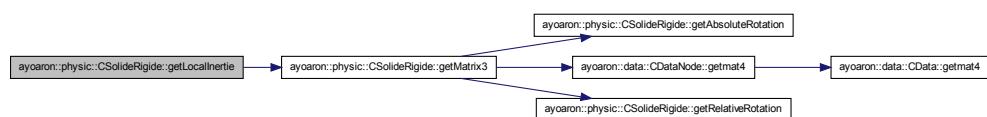
Renvoie

la matrice d'inertie dans le repère local

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.20 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère local

Renvoie

le moment résultant exprimé dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::momentResultant.

7.40.3.21 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalRitesse () [inherited]

renvoie l'expression de la vitesse de rotation dans le repère local

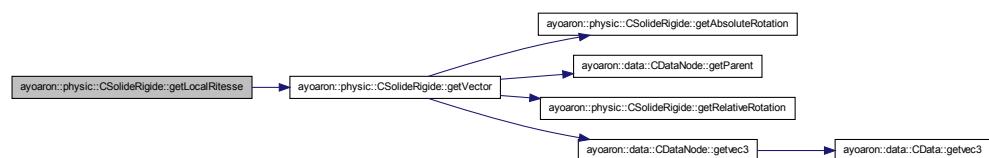
Renvoie

le vecteur vitesse de rotation dans le repère local

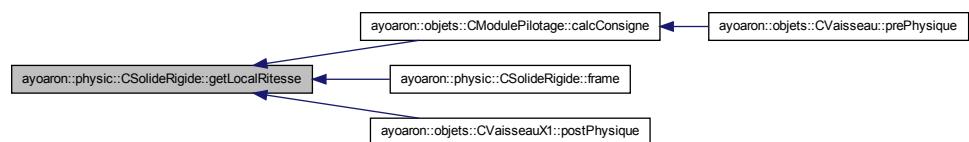
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.22 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalVitesse () [inherited]

renvoie l'expression de la vitesse de translation dans le repère local

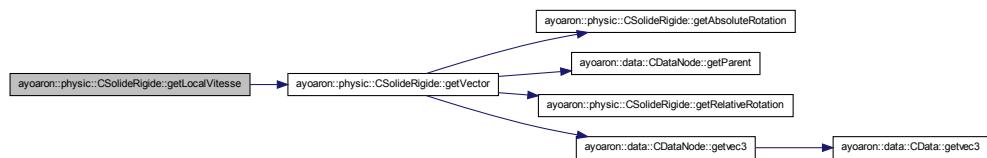
Renvoie

le vecteur vitesse dans le repère local

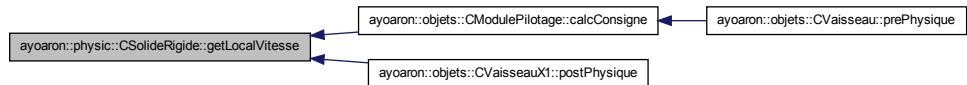
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.23 irr : :f64 ayoaron ::physic ::CSolideRigide ::getMasse () [inline, inherited]

fonction qui renvoie la masse du solide

Renvoie

la masse du solide

Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.40.3.24 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getMatrix3 (std ::string *nom*,
REPERE *ou*, REPERE *reperestockage*) [protected, inherited]**

renvoie le vecteur donné dans la base voulue

Paramètres

<i>nom</i>	nom de la clé dans la base de données
<i>ou</i>	le repère dans lequel on veut le vecteur
<i>reperestockage</i>	le repère dans lequel est stocker le vecteur

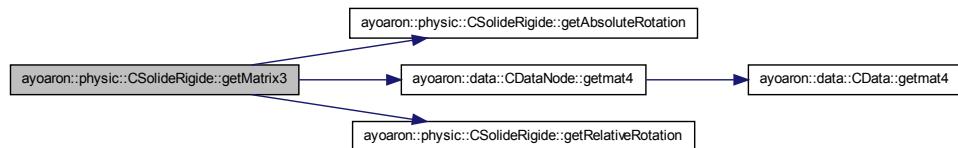
Renvoie

le vecteur voulu dans la base voulue

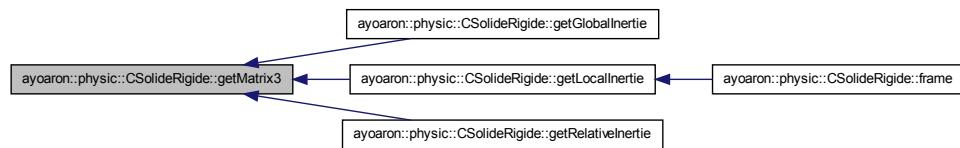
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getmat4(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getGlobalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), et ayoaron ::physic ::CSolideRigide ::getRelativeInertie().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.25 `irr ::u32 ayoaron ::physic ::CSolideRigide ::getNbEnfant() [inline, inherited]`

renvoie le nombre d'enfants de cet objet

Renvoie

le nombre d'enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.40.3.26 `irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::getNode() [inline, inherited]`

renvoie le scene node et le maillage associé

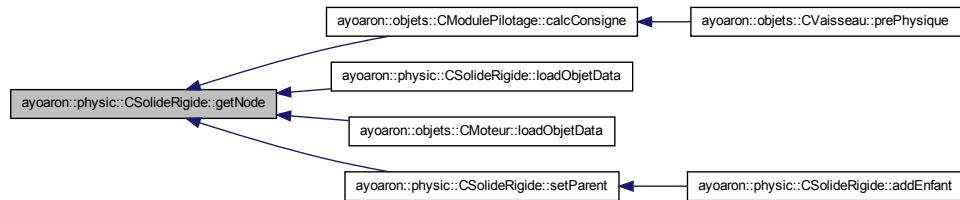
Renvoie

pointeur vers le IAnimatedMeshSceneNode

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.40.3.27 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante()
[inherited]

renvoie la force résultante exprimée dans le repère du parent

Renvoie

la résultante des efforts exprimée dans le référentiel du parent

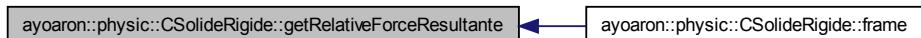
Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getRelativeRotation().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.28 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeInertie()
[inline, inherited]

renvoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère du parent

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.40.3.29 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère du parent

Renvoie

le moment résultant exprimé dans le référentiel du parent

Références ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.40.3.30 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativePosition () [inherited]

renvoie la position de l'objet par rapport à l'objet parent

Renvoie

le vecteur position relative au parent

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.40.3.31 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeRitesse ()
 [inherited]

renvoie l'expression de la vitesse de rotation dans le repère du parent

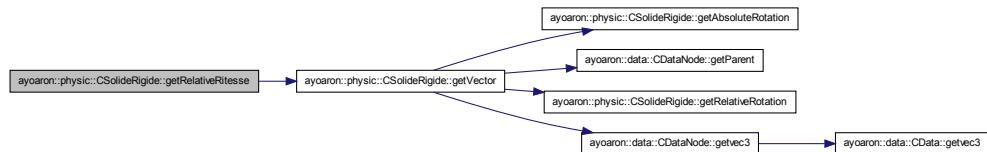
Renvoie

le vecteur vitesse de rotation dans le repère du parent

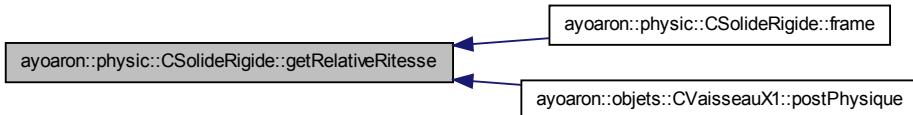
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.32 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeRotation ()
[inherited]

renvoie la matrice de rotation entre le repère relatif et le repère local

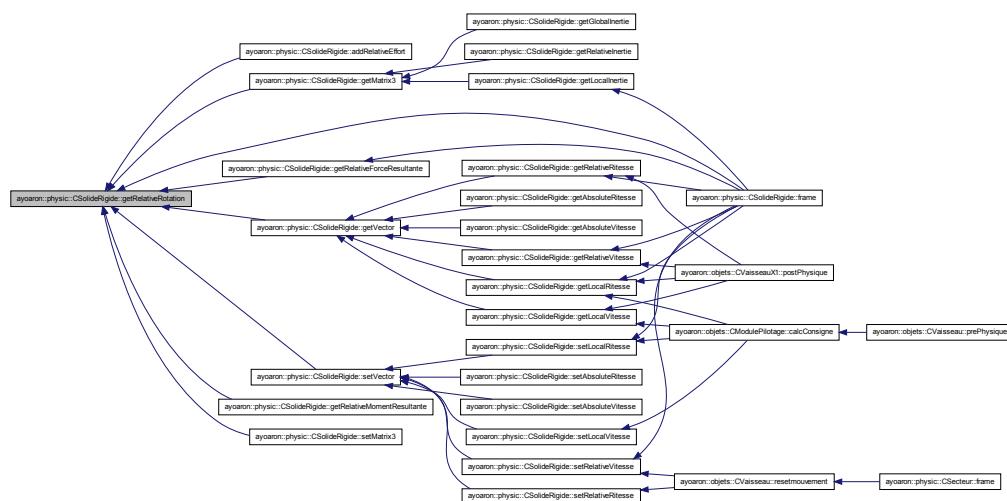
Renvoie

la matrice de rotation du repère relatif au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.40.3.33 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeVitesse ()
[inherited]

renvoie l'expression de la vitesse de translation dans le repère du parent

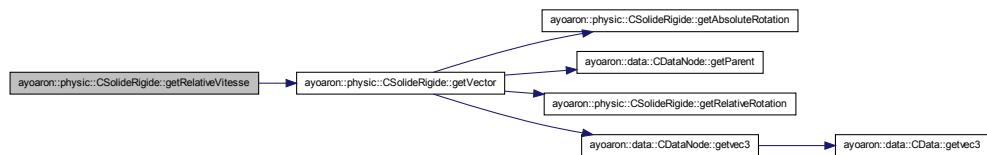
Renvoie

le vecteur vitesse dans le repère du parent

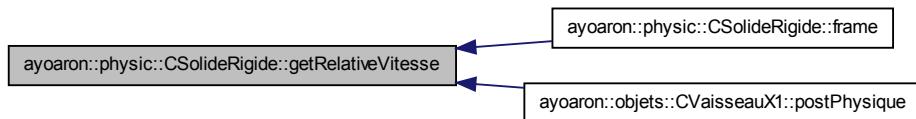
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.34 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getVector (std ::string nom, REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

<i>nom</i>	nom de la clé dans la base de données
<i>ou</i>	le repère dans lequel on veut le vecteur
<i>reperestockage</i>	le repère dans lequel est stocker le vecteur

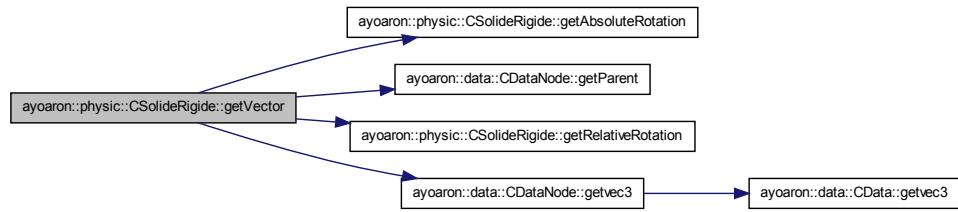
Renvoie

le vecteur voulu dans la base voulue

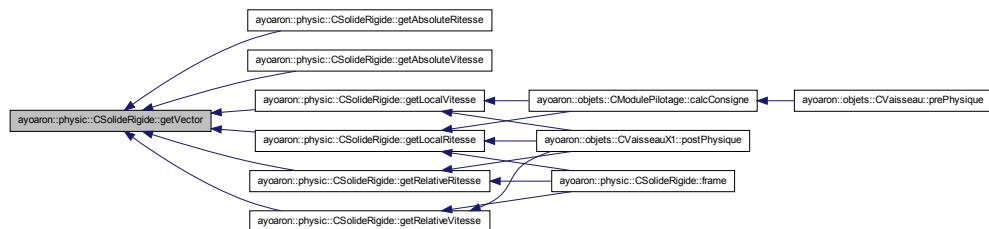
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getLocalVitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::getRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



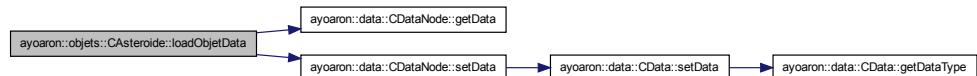
7.40.3.35 void ayoaron ::objets ::CAsteroide ::loadObjetData (data ::CDataNode * d = NULL) [virtual]

charge l'objet depuis un fichier

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Références [ayoaron ::physic ::CSolideRigide ::donn](#), [ayoaron ::data ::CDataNode ::getData\(\)](#), et [ayoaron ::data ::CDataNode ::setData\(\)](#).

Voici le graphe d'appel pour cette fonction :



7.40.3.36 `virtual void ayoaron ::physic ::CSolideRigide ::postPhysique (double dt)`
 [inline, virtual, inherited]

effectue les diverses opération de mise à jour après le calcul de la physique

Paramètres

in	dt	le pas de temps
----	----	-----------------

Réimplémentée dans `ayoaron ::objets ::CMoteur`, et `ayoaron ::objets ::CVaisseauX1`.

7.40.3.37 `virtual void ayoaron ::physic ::CSolideRigide ::prePhysique (double dt)`
 [inline, virtual, inherited]

effectue les diverses opération de mise à jour avant de calculer la physique

Paramètres

in	dt	le pas de temps
----	----	-----------------

Réimplémentée dans `ayoaron ::objets ::CMoteur`, et `ayoaron ::objets ::CVaisseau`.

Références `ayoaron ::physic ::CSolideRigide ::forceResultante`, et `ayoaron ::physic ::CSolideRigide ::momentResultant`.

7.40.3.38 `void ayoaron ::physic ::CSolideRigide ::resetEffort (void)` [inherited]

remet les efforts à zéro

Références `ayoaron ::physic ::CSolideRigide ::forceResultante`, et `ayoaron ::physic ::CSolideRigide ::momentResultant`.

7.40.3.39 `void ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse (irr ::core ::vector3df vit)` [inherited]

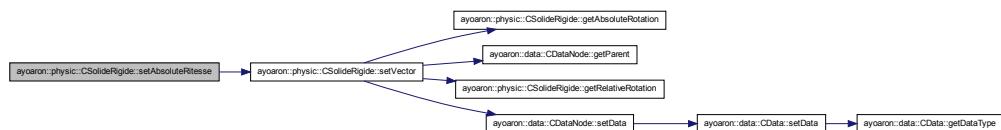
définit la vitesse de rotation du solide (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère global
----	-----	--

Références `ayoaron ::physic ::GENERAL`, `ayoaron ::physic ::PARENT`, et `ayoaron ::physic ::CSolideRigide ::setVector()`.

Voici le graphe d'appel pour cette fonction :



7.40.3.40 void ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse (irr ::core ::vector3df vit) [inherited]

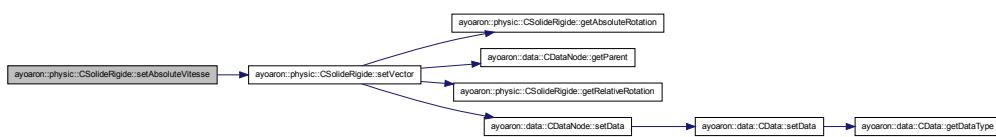
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.40.3.41 void ayoaron ::physic ::CSolideRigide ::setLocalRitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

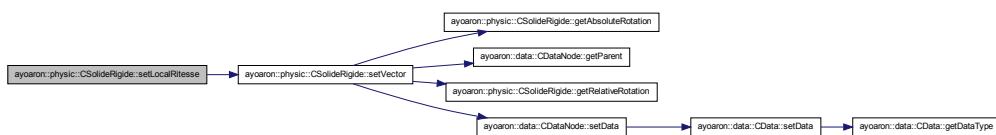
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère local
----	-----	---

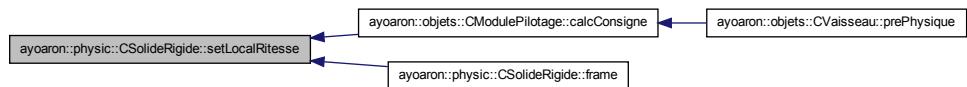
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.42 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

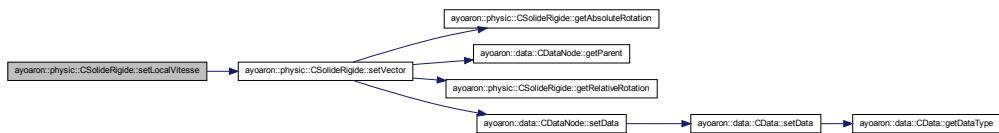
Paramètres

in	vit	le vecteur vitesse exprimé dans le repère local
----	-----	---

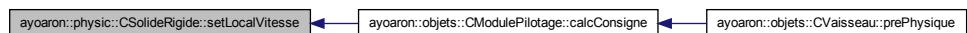
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.43 void ayoaron ::physic ::CSolideRigide ::setMatrix3 (std ::string nom, REPERE source, irr ::core ::matrix4 mat, REPERE reperstockage) [protected, inherited]

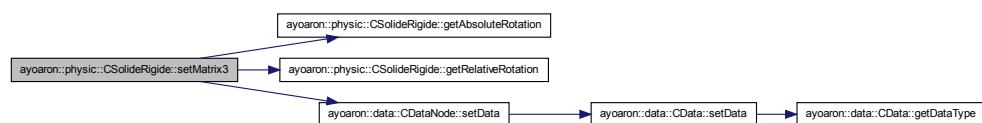
fonction permettant le changement de repère des matrices

Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>mat</i>	la matrice à stocker
<i>reperto- ckage</i>	le repère de stockage du vecteur

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.40.3.44 void ayoaron ::physic ::CSolideRigide ::setParent (CSolideRigide * *par*) [inherited]

change le parent de cet objet

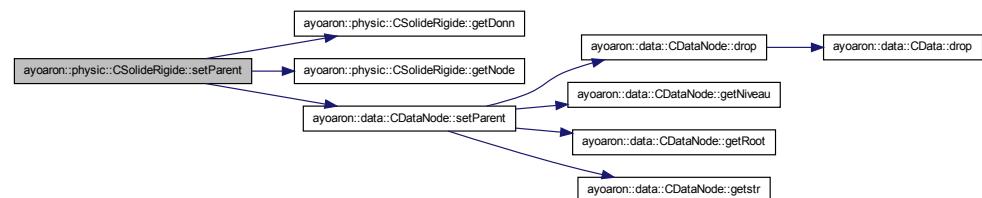
Paramètres

in	<i>par</i>	le nouveau parent
----	------------	-------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::getDonn(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::node, ayoaron ::physic ::CSolideRigide ::parent, et ayoaron ::data ::CDataNode ::setParent().

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.45 void ayoaron ::physic ::CSolideRigide ::setRelativePosition (irr ::core ::vector3df *pos*) [inherited]

définit la position de l'objet par rapport à son parent

Paramètres

<i>pos</i>	la position relative de l'objet
------------	---------------------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.46 void ayoaron ::physic ::CSolideRigide ::setRelativeRitesse (irr ::core ::vector3df *vit*) [inherited]

définit la vitesse de rotation du solide

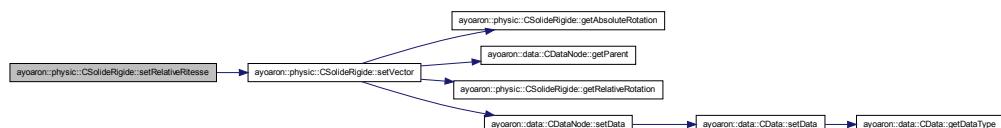
Paramètres

in	<i>vit</i>	le vecteur vitesse de rotation exprimé dans le repère du parent
----	------------	---

Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.47 void ayoaron ::physic ::CSolideRigide ::setRelativeRotation (irr ::core ::vector3df rot) [inherited]

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.

Paramètres

`rot` le vecteur rotation

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.48 void ayoaron ::physic ::CSolideRigide ::setRelativeVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

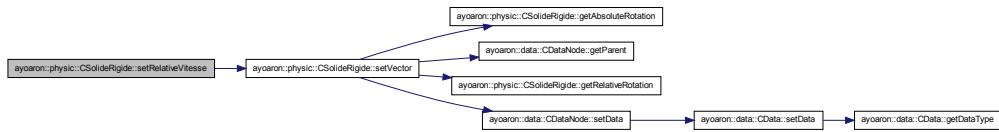
Paramètres

in	vit	le vecteur vitesse exprimé dans le repère du parent
----	-----	---

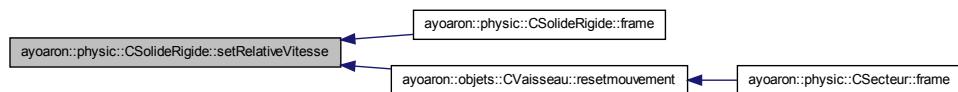
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.3.49 void ayoaron ::physic ::CSolideRigide ::setVector (std ::string nom, REPERE source, irr ::core ::vector3df vec, REPERE reperestockage) [protected, inherited]

fonction permettant le changement de repère des vecteurs

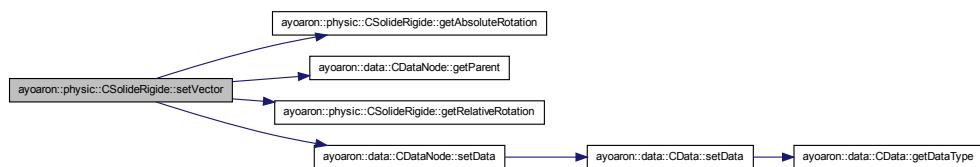
Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>vec</i>	le vecteur à stocker
<i>reperstockage</i>	le repère de stockage du vecteur

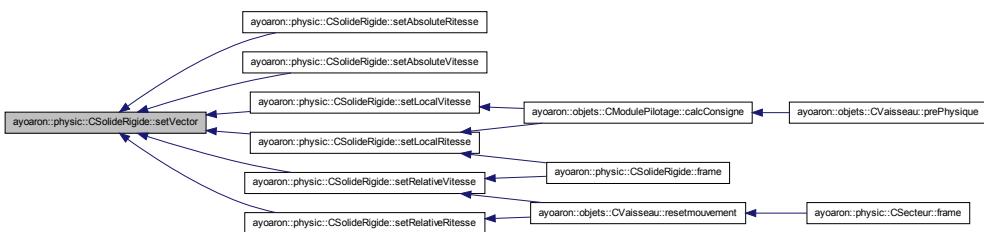
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setLocalVitesse(), ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.40.4 Documentation des données membres

7.40.4.1 **irr** : :scene : :ICameraSceneNode* ayoaron : :physic : :CSolideRigide : :cam
[protected, inherited]

pointeur vers le node de camera

Référencé par ayoaron : :physic : :CSolideRigide : :CSolideRigide(), ayoaron : :physic : :CSolideRigide : :frame(), et ayoaron : :objets : :CVaisseauX1 : :loadObjetData().

7.40.4.2 **irr** : :core : :vector3df ayoaron : :physic : :CSolideRigide : :caml
[protected, inherited]

direction du regard de la caméra

Référencé par ayoaron : :physic : :CSolideRigide : :CSolideRigide(), ayoaron : :physic : :CSolideRigide : :frame(), et ayoaron : :objets : :CVaisseauX1 : :loadObjetData().

7.40.4.3 **irr** : :core : :vector3df ayoaron : :physic : :CSolideRigide : :camp
[protected, inherited]

position de la caméra dans le repère local

Référencé par ayoaron : :physic : :CSolideRigide : :CSolideRigide(), ayoaron : :physic : :CSolideRigide : :frame(), et ayoaron : :objets : :CVaisseauX1 : :loadObjetData().

7.40.4.4 **irr** : :core : :vector3df ayoaron : :physic : :CSolideRigide : :camup
[protected, inherited]

vecteur de la verticale de la caméra

Référencé par ayoaron : :physic : :CSolideRigide : :CSolideRigide(), ayoaron : :physic : :CSolideRigide : :frame(), et ayoaron : :objets : :CVaisseauX1 : :loadObjetData().

7.40.4.5 **data** : :CDataNode* ayoaron : :physic : :CSolideRigide : :donn
[protected, inherited]

pointeur vers la structure de donnée

Référencé par ayoaron : :physic : :CSolideRigide : :CSolideRigide(), ayoaron : :physic : :CSolideRigide : :frame(), ayoaron : :physic : :CSolideRigide : :getDonn(), ayoaron : :objets : :CMoteur : :getForce(), ayoaron : :physic : :CSolideRigide : :getMasse(), ayoaron : :physic : :CSolideRigide : :getMatrix3(), ayoaron : :objets : :CMoteur : :getPoussee(), ayoaron : :objets : :CMoteur : :getPousseeMax(), ayoaron : :objets : :CMoteur : :getRegime(), ayoaron : :physic : :CSolideRigide : :getVector(), ayoaron : :objets : :CMoteur : :isReversible(), ayoaron : :objets : :CVaisseauX1 : :loadObjetData(), ayoaron : :physic : :CSolideRigide : :loadObjetData(), ayoaron : :objets : :CMoteur : :loadObjetData(), loadObjetData(), ayoaron : :objets : :CVaisseauX1 : :postPhysique(), ayoaron : :objets : :CMoteur : :postPhysique(), ayoaron : :physic : :CSolideRigide : :setMatrix3(), ayoaron : :physic : :CSolideRigide : :setParent(), ayoaron : :objets : :CMoteur : :setRegime(), ayoaron : :physic : :CSolideRigide : :setRelativePosition(), ayoaron : :physic : :CSolideRigide : :setRelativeRotation(), et ayoaron : :physic : :CSolideRigide : :setVector().

7.40.4.6 **irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::forceResultante** [protected, inherited]

le vecteur de la résultante des efforts, il est exprimé dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getLocalForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.40.4.7 **bool ayoaron ::physic ::CSolideRigide ::hascamera** [protected, inherited]

interrupteur de caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.40.4.8 **std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::CSolideRigide ::l_enfant** [protected, inherited]

liste des objets enfants

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getEnfant(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd(), ayoaron ::physic ::CSolideRigide ::getNbEnfant(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

7.40.4.9 **irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::momentResultant** [protected, inherited]

le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.40.4.10 **irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::node** [protected, inherited]

pointeur vers le maillage et le scene node

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::getRelativePosition(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setParent(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), et ayoaron ::physic ::CSolideRigide ::setRelativeRotation().

7.40.4.11 CSolideRigide* ayoaron ::physic ::CSolideRigide ::parent [protected, inherited]

pointeur vers l'objet parent

Réimplémentée dans ayoaron ::objets ::CMoteur.

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::objets ::CVaisseau ::CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

7.40.4.12 double ayoaron ::objets ::CAsteroid : :srho [protected]

masse volumique

7.40.4.13 physic ::CSecteur* ayoaron ::objets ::CAsteroid ::world [protected]

pointeur vers le secteur parent

Réimplémentée à partir de ayoaron ::physic ::CSolideRigide.

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

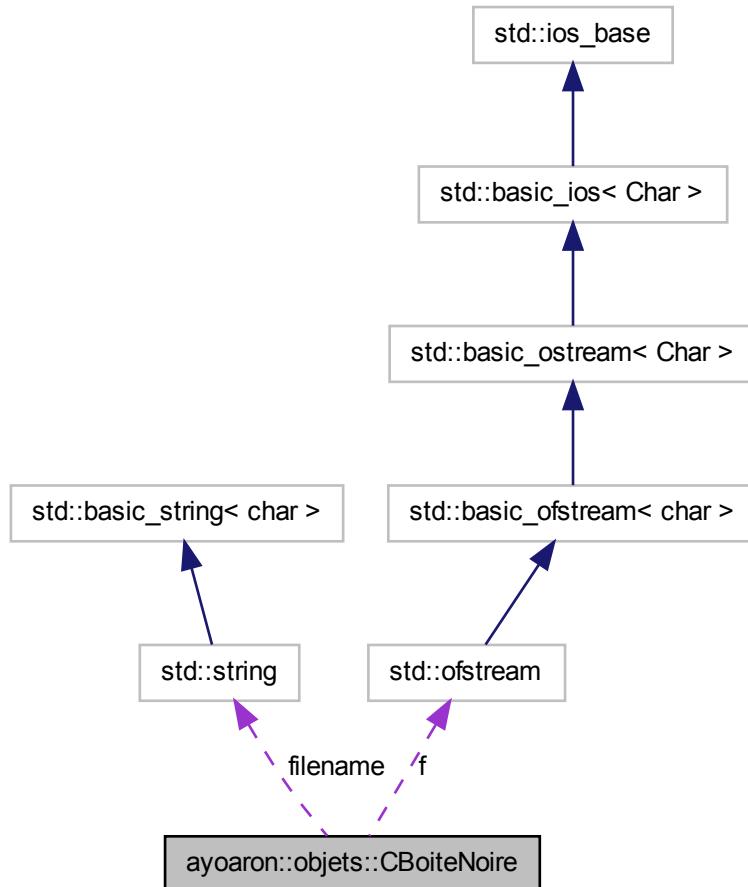
- CAsteroid.h
- CAsteroid.cpp

7.41 Référence de la classe ayoaron ::objets ::CBoiteNoire

l'élément de boite-noire

```
#include <CBoiteNoire.h>
```

Graphe de collaboration de ayoaron ::objets ::CBoiteNoire :



Fonctions membres publiques

- `CBoiteNoire (std ::string file)`
Constructeur.
- `void frame (irr ::f64 dt)`
mise à jour
- `void logstr (std ::string s)`
log la chaîne de caractère
- `void logvect (std ::string nom, irr ::core ::vector3df vec)`
log d'un vecteur
- `virtual ~CBoiteNoire ()`
Destructeur.

Attributs protégés

- std ::ofstream **f**
flux de sortie
- std ::string **filename**
nom de fichier
- irr ::f64 **temps**
mesure du temps

7.41.1 Description détaillée

l'élément de boite-noire

7.41.2 Documentation des constructeurs et destructeur**7.41.2.1 ayoaron ::objets ::CBoiteNoire ::CBoiteNoire (std ::string *file*)**

Constructeur.

Références f, filename, et temps.

7.41.2.2 ayoaron ::objets ::CBoiteNoire ::~CBoiteNoire () [virtual]

Destructeur.

Références f.

7.41.3 Documentation des fonctions membres**7.41.3.1 void ayoaron ::objets ::CBoiteNoire ::frame (irr ::f64 *dt*)**

mise à jour

Références f, et temps.

Référencé par ayoaron ::objets ::CVaisseau ::prePhysique().

Voici le graphe d'appel pour cette fonction :

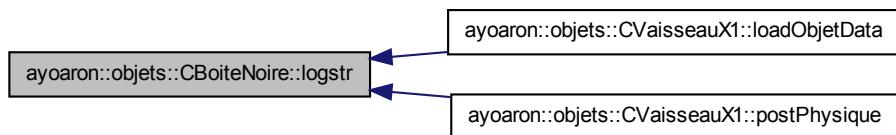
**7.41.3.2 void ayoaron ::objets ::CBoiteNoire ::logstr (std ::string *s*)**

log la chaîne de caractère

Références f.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.41.3.3 void ayoaron ::objets ::CBoiteNoire ::logvect (std ::string nom, irr ::core ::vector3df vec)

log d'un vecteur

Références f.

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.41.4 Documentation des données membres

7.41.4.1 std ::ofstream ayoaron ::objets ::CBoiteNoire ::f [protected]

flux de sortie

Référencé par CBoiteNoire(), frame(), logstr(), logvect(), et ~CBoiteNoire().

7.41.4.2 std ::string ayoaron ::objets ::CBoiteNoire ::filename [protected]

nom de fichier

Référencé par CBoiteNoire().

7.41.4.3 irr ::f64 ayoaron ::objets ::CBoiteNoire ::temps [protected]

mesure du temps

Référencé par CBoiteNoire(), et frame().

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

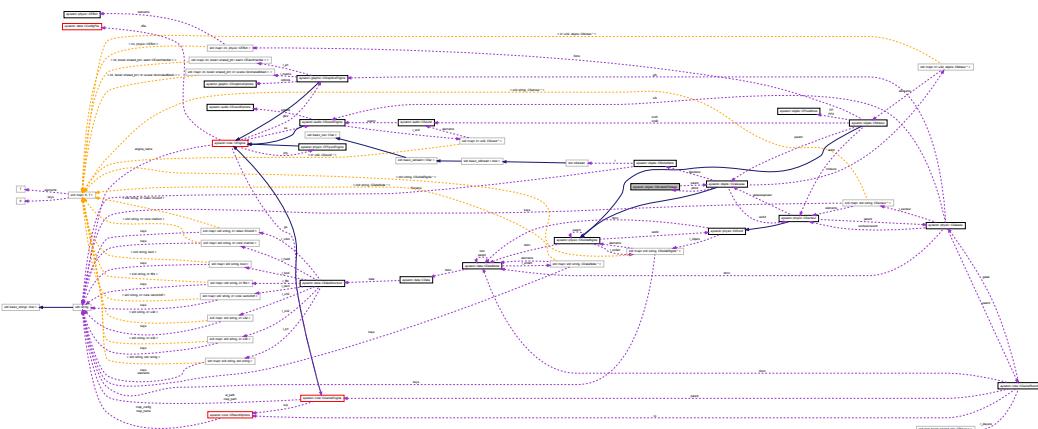
- [CBoiteNoire.h](#)
- [CBoiteNoire.cpp](#)

7.42 Référence de la classe ayoaron ::objets ::CModulePilotage

une classe de module de pilotage

```
#include <CModulePilotage.h>
```

Graphe de collaboration de ayoaron ::objets ::CModulePilotage :



Fonctions membres publiques

- virtual void **calcConsigne** (irr ::f64 dt)
calcule de la consigne
- **CModulePilotage** (CVaisseau *p, irr ::core ::vector3df accm, irr ::core ::vector3df arom, irr ::f64 ddt1)
Constructeur attention !!! doit être appelé APRES la création de tous les moteurs.
- virtual void **setConsigne** (irr ::core ::vector3df cvit, irr ::core ::vector3df cvitrot, irr ::f64 dt)
setconsigne
- virtual **~CModulePilotage** ()
Destructeur.

Attributs protégés

- irr ::core ::vector3df **accmax**
acceleration max
- irr ::core ::vector3df **aromax**

- *acceleration en rotation max*
- `irr ::core ::vector3df consignevit`
la consigne de vitesse
- `irr ::core ::vector3df consignevitrot`
la consigne de vitesse de rotation
- `irr ::core ::vector3df deltavit`
le delta de vitesse
- `irr ::core ::vector3df deltavitrot`
le delta de vitesse de rotation
- `irr ::f64 dt1`
delai de réponse
- `irr ::core ::vector3df DTV`
vecteur du grand pas en translation
- `irr ::core ::vector3df DTVR`
vecteur du grand pas en rotation
- `CVaisseau * parent`
pointeur vers le parent
- `irr ::core ::vector3df SRV`
signe du deltarot V
- `irr ::core ::vector3df SV`
signe du delta V
- `irr ::f32 temps`
temps depuis la dernière consigne

7.42.1 Description détaillée

une classe de module de pilotage

7.42.2 Documentation des constructeurs et destructeur

7.42.2.1 ayoaron ::objets ::CModulePilotage (`CVaisseau * p,` `irr ::core ::vector3df accm, irr ::core ::vector3df arom, irr ::f64 ddt1`)

Constructeur attention !!! dois être appelé APRES la création de tous les moteurs.

Références accmax, aromax, consignevit, consignevitrot, dt1, parent, et temps.

7.42.2.2 ayoaron ::objets ::CModulePilotage ::~CModulePilotage () [virtual]

Destructeur.

7.42.3 Documentation des fonctions membres

7.42.3.1 `void ayoaron ::objets ::CModulePilotage ::calcConsigne (irr ::f64 dt)` `[virtual]`

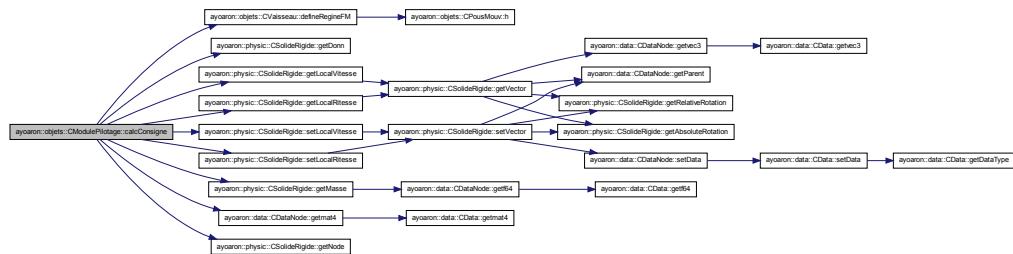
calcule de la consigne

Références accmax, aromax, consignevit, consignevitrot, ayoaron ::objets ::CVaisseau ::defineRegimeFM(), ayoaron ::physic ::CSolideRigide ::getDonn(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getLocalVitesse(), ayoaron ::physic ::CSolideRigide ::getMasse(), ayoaron ::data ::CDa-

taNode ::getmat4(), ayoaron ::physic ::CSolideRigide ::getNode(), parent, ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), et ayoaron ::physic ::CSolideRigide ::setLocalVitesse().

Référencé par ayoaron ::objets ::CVaisseau ::prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.42.3.2 `void ayoaron ::objets ::CModulePilotage ::setConsigne (irr ::core ::vector3df cvit, irr ::core ::vector3df cvitrot, irr ::f64 dt) [virtual]`

setconsigne

Références consignevit, et consignevitrot.

Référencé par ayoaron ::objets ::CVaisseau ::prePhysique().

Voici le graphe d'appel pour cette fonction :



7.42.4 Documentation des données membres

7.42.4.1 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::accmax**
[protected]

acceleration max

Référencé par calcConsigne(), et CModulePilotage().

7.42.4.2 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::aromax**
[protected]

acceleration en rotation max

Référencé par calcConsigne(), et CModulePilotage().

7.42.4.3 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::consignevit**
[protected]

la consigne de vitesse

Référencé par calcConsigne(), CModulePilotage(), et setConsigne().

7.42.4.4 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::consignevitrot**
[protected]

la consigne de vitesse de rotation

Référencé par calcConsigne(), CModulePilotage(), et setConsigne().

7.42.4.5 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::deltavit**
[protected]

le delta de vitesse

7.42.4.6 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::deltavitrot**
[protected]

le delta de vitesse de rotation

7.42.4.7 **irr ::f64 ayoaron ::objets ::CModulePilotage ::dt1** [protected]

delai de réponse

Référencé par CModulePilotage().

7.42.4.8 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::DTV**
[protected]

vecteur du grand pas en translation

7.42.4.9 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::DTVR**
[protected]

vecteur du grand pas en rotation

7.42.4.10 **CVaisseau* ayoaron ::objets ::CModulePilotage ::parent**
[protected]

pointeur vers le parent

Référencé par calcConsigne(), et CModulePilotage().

7.42.4.11 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::SRV**
[protected]

signe du deltarot V

7.42.4.12 **irr ::core ::vector3df ayoaron ::objets ::CModulePilotage ::SV**
[protected]

signe du delta V

7.42.4.13 **irr ::f32 ayoaron ::objets ::CModulePilotage ::temps** [protected]

temps depuis la dernière consigne

Référencé par CModulePilotage().

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

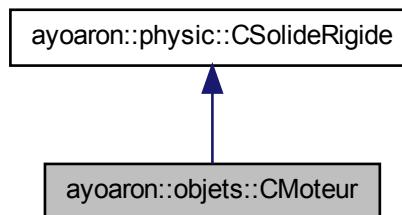
- [CModulePilotage.h](#)
- [CModulePilotage.cpp](#)

7.43 Référence de la classe ayoaron ::objets ::CMoteur

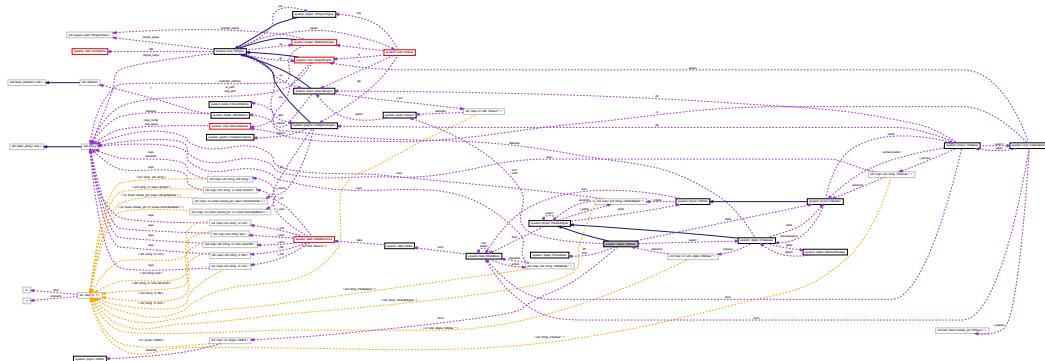
classe de description des moteurs

```
#include <CMoteur.h>
```

Graphe d'héritage de ayoaron ::objets ::CMoteur :



Graphe de collaboration de ayoaron ::objets ::CMoteur :



Fonctions membres publiques

- void **addAbsoluteEffort** (CEffort F)
ajoute un effort exprimé dans le repère global
- void **addEnfant** (std ::string nom, CSolideRigide *enfant)
ajoute un nouvel enfant (s'il n'existe pas déjà)
- void **addLocalEffort** (CEffort F)
ajoute un effort exprimé dans le repère local
- void **addRelativeEffort** (CEffort F)
ajoute un effort exprimé dans le repère du parent
- **CMoteur** (data ::CDataNode *d, physic ::CSecteur *p, objets ::CVaisseau *po)
Constructeur.
- void **dropEnfant** (std ::string nom)
supprime un enfant de la liste
- void **frame** (double dt)
effectue les diverses opération de mise à jour
- irr ::core ::vector3df **getAbsoluteForceResultante** ()
renvoie la force résultante exprimée dans le repère global
- irr ::core ::vector3df **getAbsoluteMomentResultante** ()
renvoie le moment résultant exprimé dans le repère global
- irr ::core ::vector3df **getAbsolutePosition** ()
renvoie la position de l'objet dans le monde
- irr ::core ::vector3df **getAbsoluteRitesse** ()
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- irr ::core ::matrix4 **getAbsoluteRotation** ()
renvoie la matrice de rotation entre le repère global et le repère local
- irr ::core ::vector3df **getAbsoluteVitesse** ()
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- **data ::CDataNode * getDonn ()**
acces aux données
- CSolideRigide * **getEnfant** (std ::string nom)
renvoie l'enfant donné
- physic ::CEffort **getForce** (void)
renvoie la force de réaction à la poussée
- irr ::core ::matrix4 **getGlobalInertie** ()
renvoie la matrice d'inertie
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantBegin** ()

- *renvoie un itérator sur les enfant pointant au début de la map*
 - `std ::map< std ::string, CSolideRigide * > ::iterator getIteratorEnfantEnd ()`
renvoie un itérator sur les enfant pointant à la fin de la map
- `irr ::core ::vector3df getLocalForceResultante ()`
renvoie la force résultante exprimée dans le repère local
- `irr ::core ::matrix4 getLocalInertie ()`
revoie la matrice d'inertie
- `irr ::core ::vector3df getLocalMomentResultante ()`
renvoie le moment résultant exprimé dans le repère local
- `irr ::core ::vector3df getLocalRitesse ()`
renvoie l'expression de la vitesse de rotation dans le repère local
- `irr ::core ::vector3df getLocalVitesse ()`
renvoie l'expression de la vitesse de translation dans le repère local
- `irr ::f64 getMasse ()`
fonction qui renvoie la masse du solide
- `irr ::u32 getNbEnfant ()`
renvoie le nombre d'enfants de cet objet
- `irr ::scene ::ISceneNode * getNode ()`
renvoie le scene node et le maillage associé
- `physic ::CEffort getPoussee (void)`
renvoie la poussée du moteur
- `irr ::f64 getPousseeMax ()`
renvoie la poussée max
- `CPousMouv getPP ()`
renvoie la capacité du moteur dans la direction 1
- `CPousMouv getPP2 ()`
renvoie la capacité du moteur dans la direction 2
- `irr ::f64 getRegime ()`
renvoie le régime
- `irr ::core ::vector3df getRelativeForceResultante ()`
renvoie la force résultante exprimée dans le repère du parent
- `irr ::core ::matrix4 getRelativelInertie ()`
revoie la matrice d'inertie
- `irr ::core ::vector3df getRelativeMomentResultante ()`
renvoie le moment résultant exprimé dans le repère du parent
- `irr ::core ::vector3df getRelativePosition ()`
renvoie la position de l'objet par rapport à l'objet parent
- `irr ::core ::vector3df getRelativeRitesse ()`
renvoie l'expression de la vitesse de rotation dans le repère du parent
- `irr ::core ::matrix4 getRelativeRotation ()`
renvoie la matrice de rotation entre le repère relatif et le repère local
- `irr ::core ::vector3df getRelativeVitesse ()`
renvoie l'expression de la vitesse de translation dans le repère du parent
- `bool isReversible ()`
- `void loadObjetData (data ::CDataNode *d=NULL)`
charge les données
- `virtual void postPhysique (double dt)`
fonction de frame du moteur
- `virtual void prePhysique (double dt)`
fonction de frame du moteur
- `void resetEffort ()`
remet les efforts à zéro
- `void setAbsoluteRitesse (irr ::core ::vector3df vit)`
définit la vitesse de rotation du solide (tient compte de la vitesse d'entraînement du repère parent)
- `void setAbsoluteVitesse (irr ::core ::vector3df vit)`
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
- `void setLocalRitesse (irr ::core ::vector3df vit)`

- void **setLocalVitesse** (irr ::core ::vector3df vit)
 - définit la vitesse de rotation du solide*
- void **setParent** (CSolideRigide *par)
 - définit la vitesse du solide*
 - change le parent de cet objet*
- void **setRegime** (irr ::f64 r)
 - définit le régime moteur [0..1]*
- void **setRelativePosition** (irr ::core ::vector3df pos)
 - définit la position de l'objet par rapport à son parent*
- void **setRelativeRitesse** (irr ::core ::vector3df vit)
 - définit la vitesse de rotation du solide*
- void **setRelativeRotation** (irr ::core ::vector3df rot)
 - à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.*
- void **setRelativeVitesse** (irr ::core ::vector3df vit)
 - définit la vitesse du solide*
- virtual **~CMoteur** ()
 - Destructeur.*

Fonctions membres protégées

- irr ::core ::matrix4 **getMatrix3** (std ::string nom, REPERE ou, REPERE reperestockage)
 - renvoie le vecteur donné dans la base voulue*
- irr ::core ::vector3df **getVector** (std ::string nom, REPERE ou, REPERE reperestockage)
 - renvoie le vecteur donné dans la base voulue*
- void **setMatrix3** (std ::string nom, REPERE source, irr ::core ::matrix4 mat, REPERE reperestockage)
 - fonction permettant le changement de repère des matrices*
- void **setVector** (std ::string nom, REPERE source, irr ::core ::vector3df vec, REPERE reperestockage)
 - fonction permettant le changement de repère des vecteurs*

Attributs protégés

- irr ::scene ::ICameraSceneNode * **cam**
 - pointeur vers le node de camera*
- irr ::core ::vector3df **caml**
 - direction du regard de la caméra*
- irr ::core ::vector3df **camp**
 - position de la caméra dans le repère local*
- irr ::core ::vector3df **camup**
 - vecteur de la verticale de la caméra*
- **data** ::CDataNode * **donn**
 - pointeur vers la structure de donnée*
- irr ::scene ::IVolumeLightSceneNode * **flux**
 - le flux de particule*
- irr ::scene ::IVolumeLightSceneNode * **flux2**
 - liste des forces (1 seule ou 2 si réversible)*
- std ::map< int, **physic** ::CEffort > **force**
 - le vecteur de la résultante des efforts, il est exprimé dans le repère local*
- irr ::core ::vector3df **forceResultante**
 - le vecteur de la résultante des efforts, il est exprimé dans le repère local*
- bool **hascamera**
 - interrupteur de caméra*
- std ::map< std ::string, CSolideRigide * > **I_enfant**
 - liste des objets enfants*

- irr : ::core ::vector3df **momentResultant**
le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)
- irr : ::scene ::ISceneNode * **node**
pointeur vers le maillage et le scene node
- **objets** : ::CVaisseau * **parent**
pointeur vers le vaisseau parent
- **CPousMouv PP**
tableau de capacité du moteur
- **CPousMouv PP2**
- **audio** : ::CSound * **snd1**
pointeur vers le son du moteur
- **audio** : ::CSound * **snd2**
- **double temps**
- **physic** : ::CSecteur * **world**
pointeur vers le secteur parent

7.43.1 Description détaillée

classe de description des moteurs

7.43.2 Documentation des constructeurs et destructeur

7.43.2.1 ayoaron ::objets ::CMoteur (**data** : ::CDataNode * *d*, **physic** : ::CSecteur * *p*, **objets** : ::CVaisseau * *po*)

Constructeur.

Références flux, flux2, parent, snd1, snd2, et world.

7.43.2.2 ayoaron ::objets ::CMoteur ::~CMoteur () [virtual]

Destructeur.

7.43.3 Documentation des fonctions membres

7.43.3.1 void ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort (**CEffort** *F*)
[inherited]

ajoute un effort exprimé dans le repère global

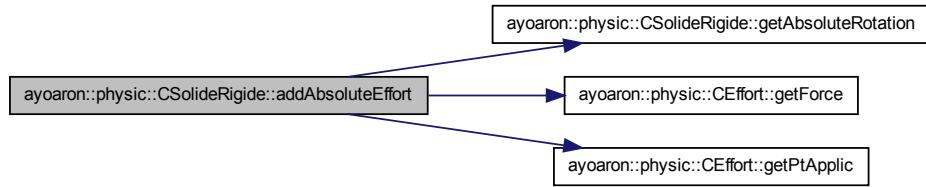
Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère global, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.43.3.2 void ayoaron ::physic ::CSolideRigide ::addEnfant (std ::string *nom*,
CSolideRigide * *enfant*) [inherited]**

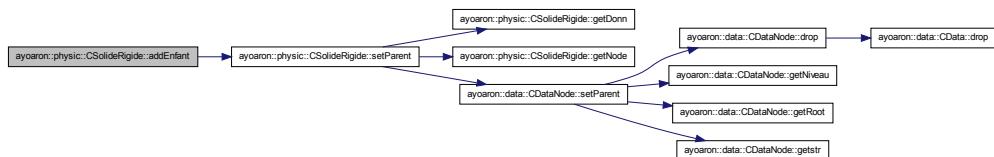
ajoute un nouvel enfant (s'il n'existe pas déjà)

Paramètres

<i>nom</i>	le nom (unique) de l'enfant
<i>enfant</i>	pointeur vers l'enfant en question

Références ayoaron ::physic ::CSolideRigide ::l_enfant, et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.43.3.3 void ayoaron ::physic ::CSolideRigide ::addLocalEffort (CEffort F)
 [inherited]

ajoute un effort exprimé dans le repère local

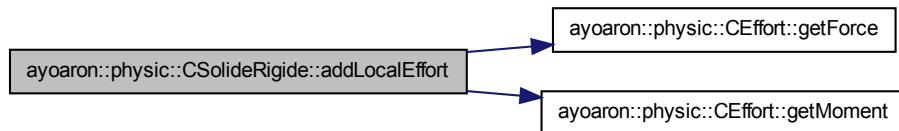
Paramètres

in	F	l'effort à ajouterexprimé dans le repère local, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	-----	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getMoment(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.4 void ayoaron ::physic ::CSolideRigide ::addRelativeEffort (CEffort F)
 [inherited]

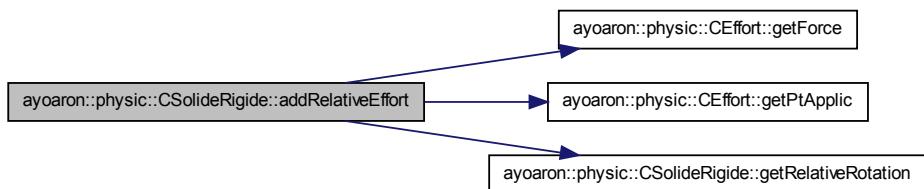
ajoute un effort exprimé dans le repère du parent

Paramètres

in	F	l'effort à ajouterexprimé dans le repère du parent, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	-----	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.43.3.5 void ayoaron ::physic ::CSolideRigide ::dropEnfant (std ::string nom) [inherited]

supprime un enfant de la liste

Paramètres

<i>nom</i>	nom de l'enfant à supprimer
------------	-----------------------------

7.43.3.6 void ayoaron ::physic ::CSolideRigide ::frame (double dt) [inherited]

effectue les diverses opération de mise à jour

Paramètres

in	<i>dt</i>	le pas de temps
----	-----------	-----------------

la frame :

tout d'abord, l'effort global

on pre frame tous les objets enfants

la translation, calculée dans le repère du parent

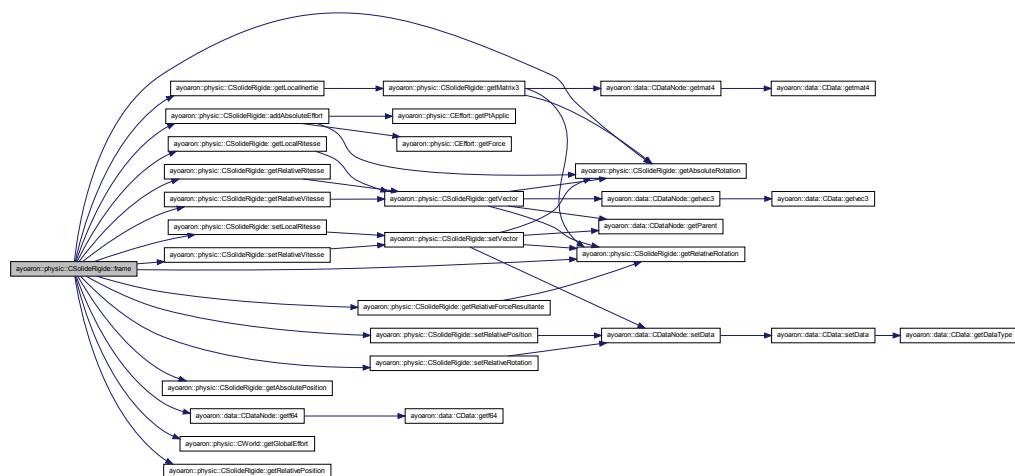
la rotation, calculée dans le repère parent

on post frame tous les objets enfants

Références ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::cam, ayoaron ::physic ::CSolideRigide ::caml, ayoaron ::physic ::CSolideRigide ::camp, ayoaron ::physic ::CSolideRigide ::camup, ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getf64(), ayoaron ::phys-

```
sic : :CWorld : :getGlobalEffort(), ayoaron : :physic : :CSolideRigide : :getLocalInertie(),  
ayoaron : :physic : :CSolideRigide : :getLocalRitesse(), ayoaron : :physic : :CSolide-  
Rigide : :getRelativeForceResultante(), ayoaron : :physic : :CSolideRigide : :getRelati-  
vePosition(), ayoaron : :physic : :CSolideRigide : :getRelativeRitesse(), ayoaron : :phys-  
ic : :CSolideRigide : :getRelativeRotation(), ayoaron : :physic : :CSolideRigide : :ge-  
tRelativeVitesse(), ayoaron : :physic : :CSolideRigide : :hascamera, ayoaron : :phys-  
ic : :CSolideRigide : :l_enfant, ayoaron : :physic : :CSolideRigide : :momentResultant,  
ayoaron : :physic : :CSolideRigide : :node, ayoaron : :physic : :CSolideRigide : :setLo-  
calRitesse(), ayoaron : :physic : :CSolideRigide : :setRelativePosition(), ayoaron : :phys-  
ic : :CSolideRigide : :setRelativeRotation(), ayoaron : :physic : :CSolideRigide : :setRe-  
lativeVitesse(), et ayoaron : :physic : :CSolideRigide : :world.
```

Voici le graphe d'appel pour cette fonction :



7.43.3.7 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère global

Renvoie

la résultante des efforts exprimée dans le référentiel global

Références ayoaron : :physic : :CSolideRigide : :forceResultante, et ayoaron : :physic : :CSolideRigide : :getAbsoluteRotation().

Voici le graphe d'appel pour cette fonction :



7.43.3.8 **irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante ()** [inherited]

renvoie le moment résultant exprimé dans le repère global

Renvoie

le moment résultant exprimé dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.43.3.9 **irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsolutePosition ()** [inherited]

renvoie la position de l'objet dans le monde

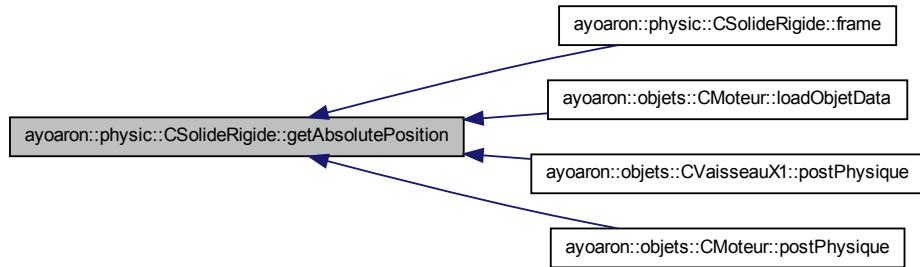
Renvoie

le vecteur position absolue de l'objet

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



7.43.3.10 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse ()
[inherited]

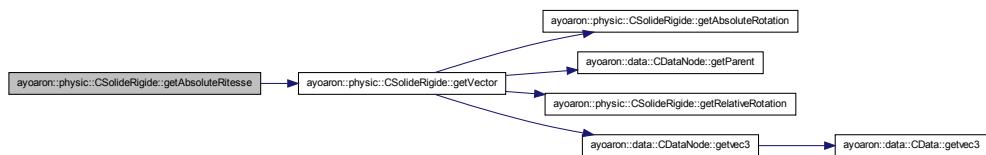
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse de rotation dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.43.3.11 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation ()
[inherited]

renvoie la matrice de rotation entre le repère global et le repère local

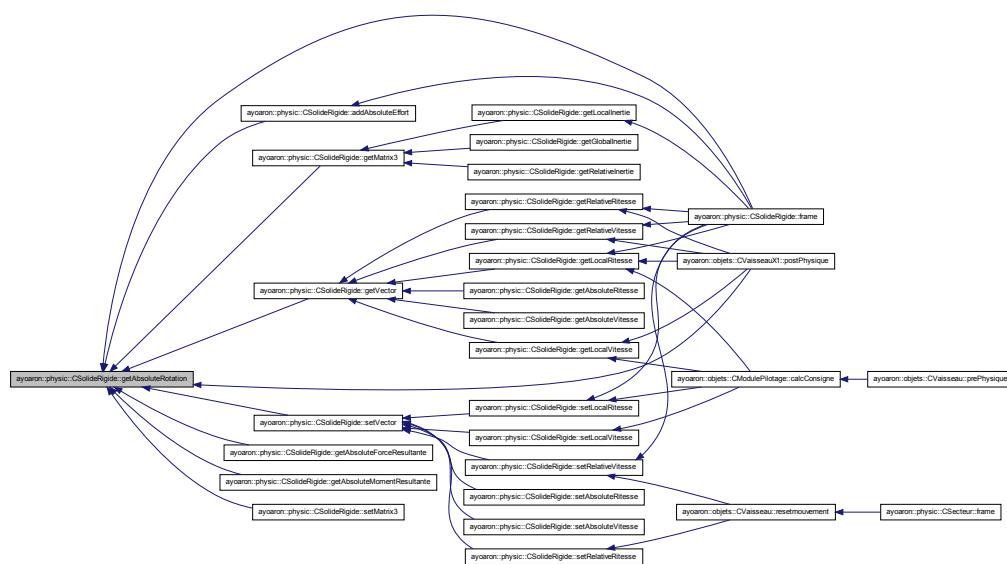
Renvoie

la matrice de rotation du repère global au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.43.3.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse ()
[inherited]

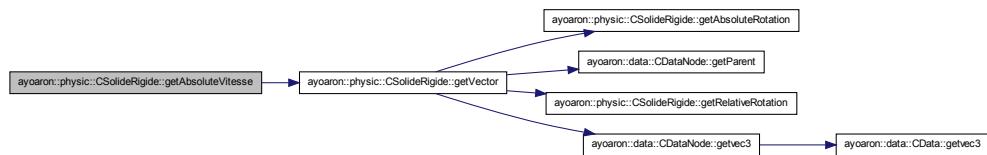
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.43.3.13 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::getDonn ()
 [inline, inherited]

acces aux données

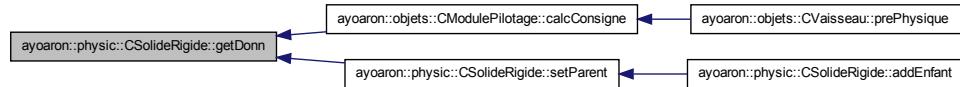
Renvoie

pointeur vers la structure de donnée

Références ayoaron ::physic ::CSolideRigide ::donn.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.43.3.14 CSolideRigide * ayoaron ::physic ::CSolideRigide ::getEnfant (std ::string nom)
 [inherited]

renvoie l'enfant donné

Paramètres

nom	le nom de l'enfant voulu
-----	--------------------------

Renvoie

pointeur vers l'enfant désiré, renvoie NULL si l'enfant n'existe pas

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

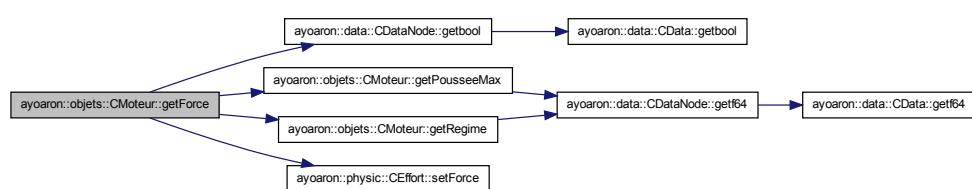
7.43.3.15 physic ::CEffort ayoaron ::objets ::CMoteur ::getForce (void)

renvoie la force de réaction à la poussée

Références ayoaron ::physic ::CSolideRigide ::donn, force, ayoaron ::data ::CDataNode ::getbool(), getPousseeMax(), getRegime(), et ayoaron ::physic ::CEffort ::setForce().

Référencé par prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

7.43.3.16 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getGlobalInertie ()
[inline, inherited]

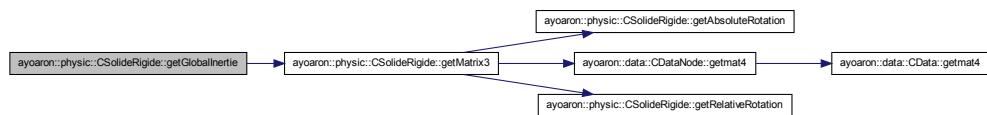
revoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Voici le graphe d'appel pour cette fonction :



7.43.3.17 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin () [inherited]`

renvoie un itérator sur les enfant pointant au début de la map

Renvoie

iterator pointant au début de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.43.3.18 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd () [inherited]`

renvoie un itérator sur les enfant pointant à la fin de la map

Renvoie

iterator pointant à la fin de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.43.3.19 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalForceResultante () [inherited]`

renvoie la force résultante exprimée dans le repère local

Renvoie

la résultante des efforts exprimée dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::forceResultante.

7.43.3.20 `irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getLocallnertie () [inline, inherited]`

revoie la matrice d'inertie

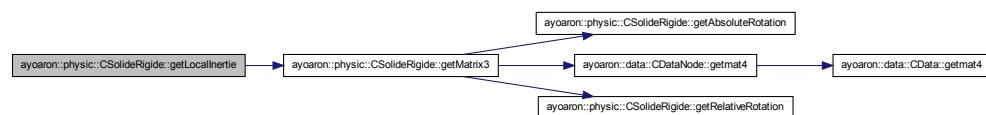
Renvoie

la matrice d'inertie dans le repère local

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LO-CAL.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.21 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante ()` [inherited]

renvoie le moment résultant exprimé dans le repère local

Renvoie

le moment résultant exprimé dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::momentResultant.

7.43.3.22 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalRitesse ()` [inherited]

renvoie l'expression de la vitesse de rotation dans le repère local

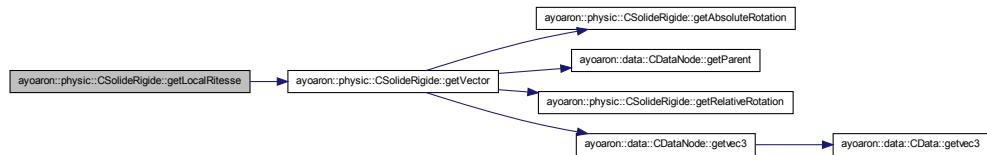
Renvoie

le vecteur vitesse de rotation dans le repère local

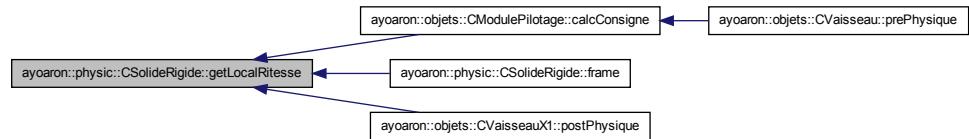
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.23 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalVitesse () [inherited]

renvoie l'expression de la vitesse de translation dans le repère local

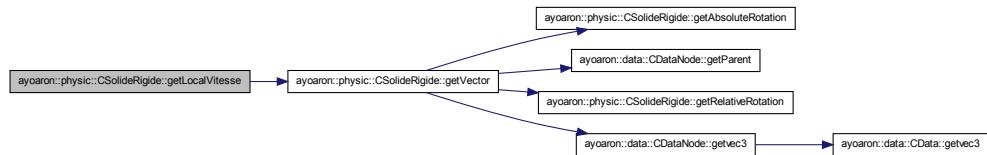
Renvoie

le vecteur vitesse dans le repère local

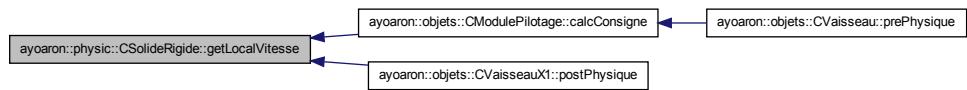
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.24 irr : :f64 ayoaron ::physic ::CSolideRigide ::getMasse () [inline, inherited]

fonction qui renvoie la masse du solide

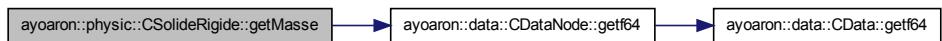
Renvoie

la masse du solide

Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CData-Node ::getf64().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.25 irr : :core ::matrix4 ayoaron ::physic ::CSolideRigide ::getMatrix3 (std ::string nom, REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

nom	nom de la clé dans la base de données
ou	le repère dans lequel on veut le vecteur

<i>reperestocage</i>	le repère dans lequel est stocker le vecteur
----------------------	--

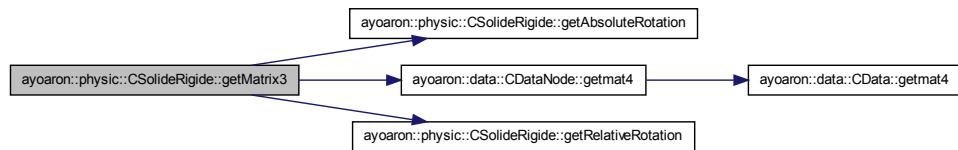
Renvoie

le vecteur voulu dans la base voulue

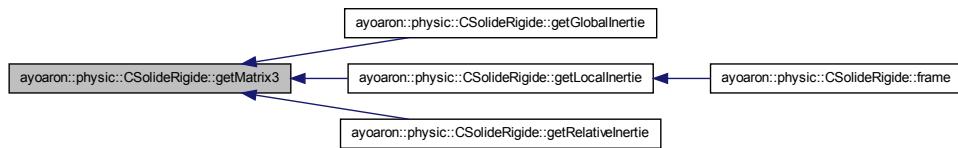
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getmat4(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getGlobalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), et ayoaron ::physic ::CSolideRigide ::getRelativeInertie().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.26 **irr : :u32 ayoaron ::physic ::CSolideRigide ::getNbEnfant () [inline, inherited]**

renvoie le nombre d'enfants de cet objet

Renvoie

le nombre d'enfant

Références ayoaron ::physic ::CSolideRigide :::l_enfant.

7.43.3.27 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::getNode ()
[inline, inherited]

renvoie le scene node et le maillage associé

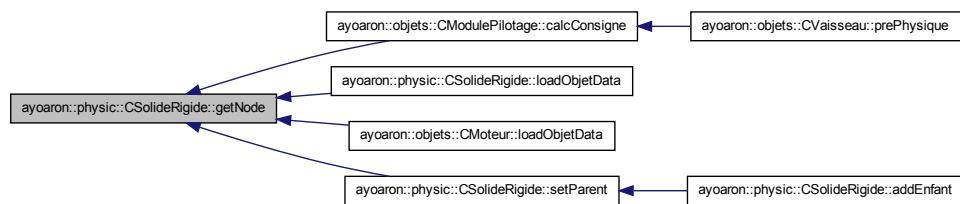
Renvoie

pointeur vers le IAnimatedMeshSceneNode

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :

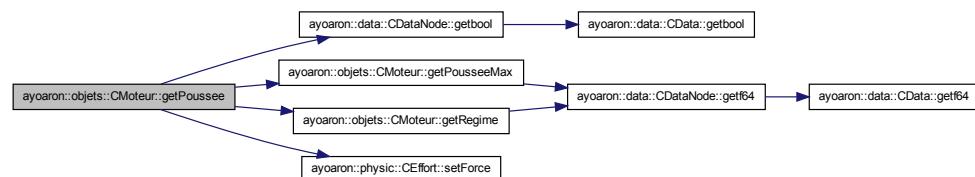


7.43.3.28 physic ::CEffort ayoaron ::objets ::CMoteur ::getPoussee (void)

renvoie la poussée du moteur

Références ayoaron ::physic ::CSolideRigide ::donn, force, ayoaron ::data ::CDataNode ::getbool(), getPousseeMax(), getRegime(), et ayoaron ::physic ::CEffort ::setForce().

Voici le graphe d'appel pour cette fonction :



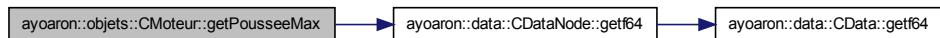
7.43.3.29 irr ::f64 ayoaron ::objets ::CMoteur ::getPousseeMax ()

renvoie la poussée max

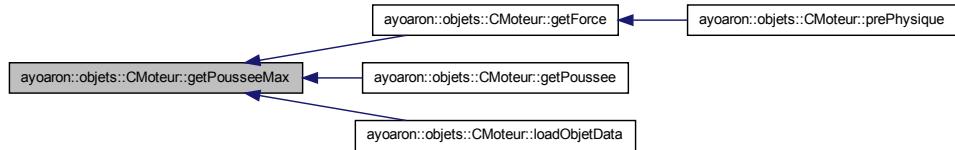
Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par getForce(), getPoussee(), et loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.30 CPousMouv ayoaron ::objets ::CMoteur ::getPP() [inline]

renvoie la capacité du moteur dans la direction 1

Références PP.

7.43.3.31 CPousMouv ayoaron ::objets ::CMoteur ::getPP2() [inline]

renvoie la capacité du moteur dans la direction 2

Références PP2.

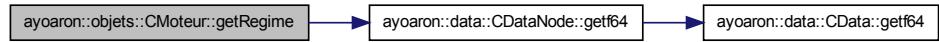
7.43.3.32 irr ::f64 ayoaron ::objets ::CMoteur ::getRegime()

renvoie le régime

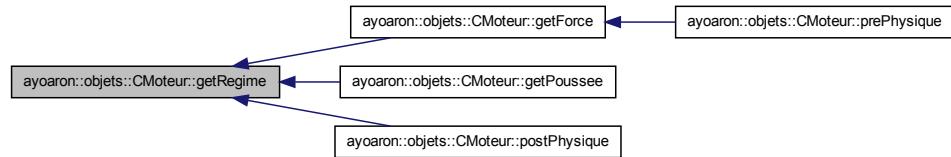
Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par getForce(), getPoussee(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.43.3.33 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante
() [inherited]**

renvoie la force résultante exprimée dans le repère du parent

Renvoie

la résultante des efforts exprimée dans le référentiel du parent

Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getRelativeRotation().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.34 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativelnertie ()
[inline, inherited]

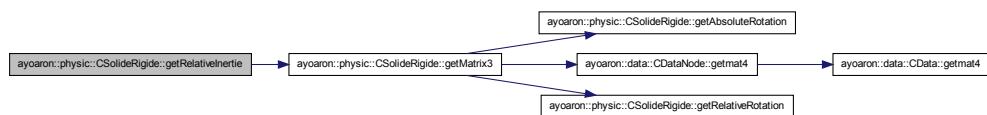
revoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère du parent

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::LO-CAL, et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.43.3.35 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère du parent

Renvoie

le moment résultant exprimé dans le référentiel du parent

Références ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.43.3.36 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativePosition ()
[inherited]

renvoie la position de l'objet par rapport à l'objet parent

Renvoie

le vecteur position relative au parent

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.43.3.37 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeRitesse ()
[inherited]

renvoie l'expression de la vitesse de rotation dans le repère du parent

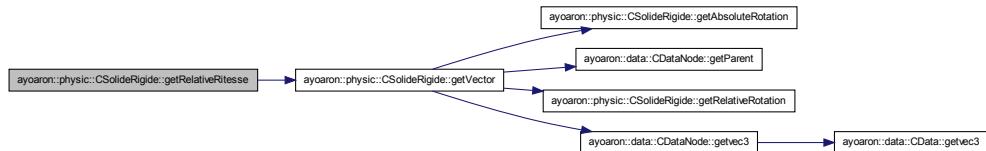
Renvoie

le vecteur vitesse de rotation dans le repère du parent

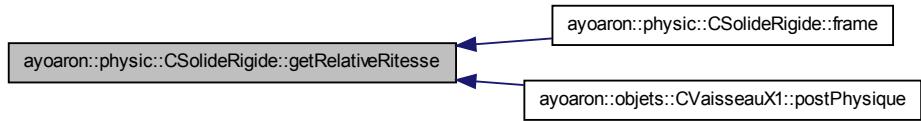
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.38 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeRotation ()
 [inherited]

renvoie la matrice de rotation entre le repère relatif et le repère local

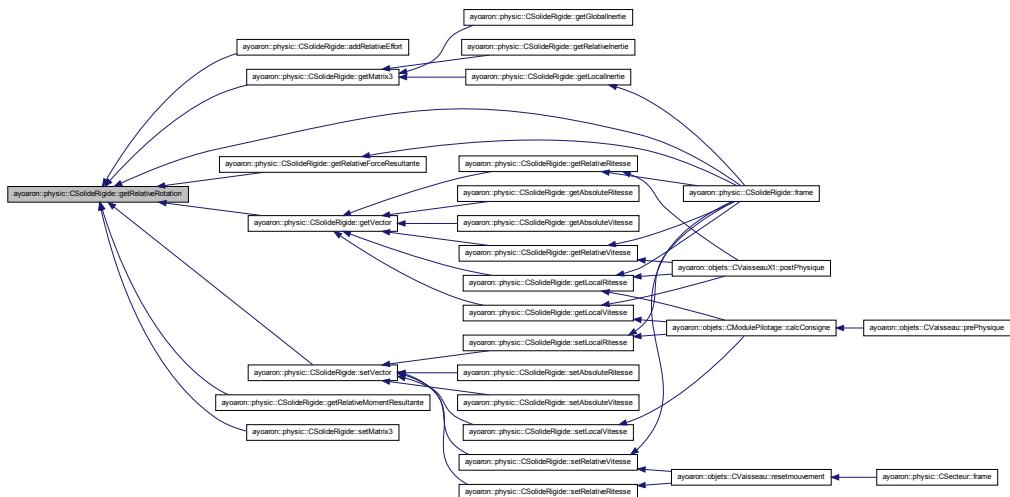
Renvoie

la matrice de rotation du repère relatif au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.43.3.39 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeVitesse ()
[inherited]

renvoie l'expression de la vitesse de translation dans le repère du parent

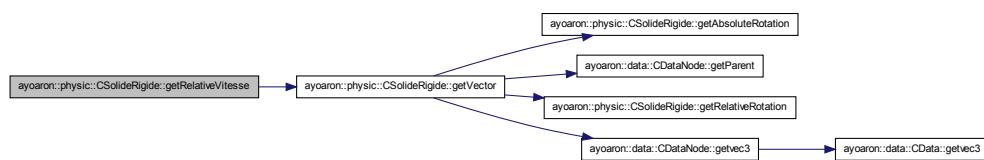
Renvoie

le vecteur vitesse dans le repère du parent

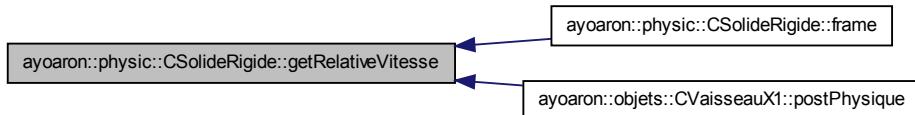
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PA-RENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.40 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getVector (std ::string nom,
REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

nom	nom de la clé dans la base de données
ou	le repère dans lequel on veut le vecteur
reperestockage	le repère dans lequel est stocker le vecteur

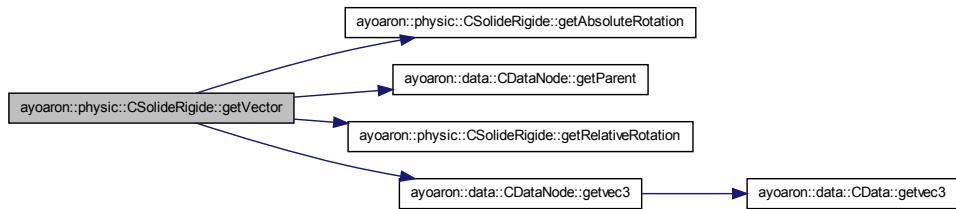
Renvoie

le vecteur voulu dans la base voulue

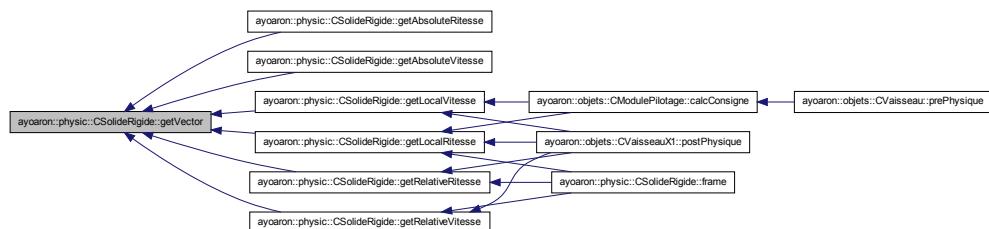
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getLocalVitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::getRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.41 bool ayoaron ::objets ::CMoteur ::isReversible() [inline]

\ brief renvoie true si le moteur est reversible

Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::getbool().

Voici le graphe d'appel pour cette fonction :



7.43.3.42 void ayoaron ::objets ::CMoteur ::loadObjetData (data : CDataNode * d = NULL) [virtual]

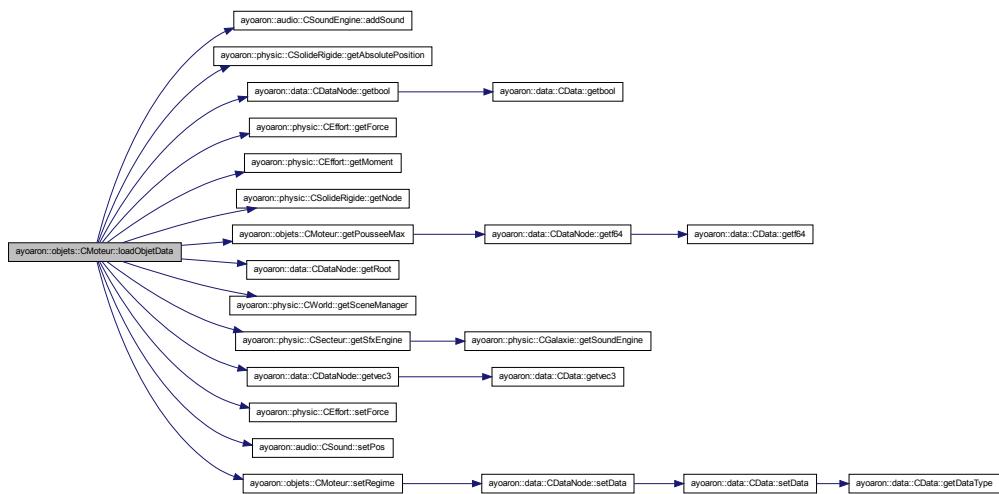
charge les données

chargement très particulier pour les moteurs

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Références ayoaron ::audio ::CSoundEngine ::addSound(), ayoaron ::physic ::CSolideRigide :::donn, flux, flux2, force, ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::data ::CDataNode ::getbool(), ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getMoment(), ayoaron ::physic ::CSolideRigide ::getNode(), getPousseeMax(), ayoaron ::data ::CDataNode ::getRoot(), ayoaron ::physic ::CWorld ::getSceneManager(), ayoaron ::physic ::CSecteur ::getSfxEngine(), ayoaron ::data ::CDataNode ::getvec3(), parent, PP, PP2, ayoaron ::physic ::CEffort ::setForce(), ayoaron ::audio ::CSound ::setPos(), setRegime(), snd1, snd2, ayoaron ::audio ::SND_3D_CONTINUOUS, et world.

Voici le graphe d'appel pour cette fonction :



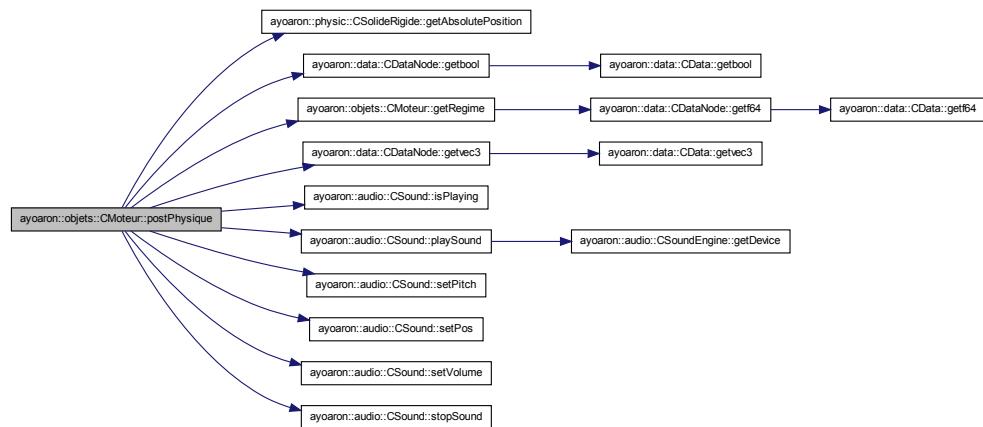
7.43.3.43 void ayoaron ::objets ::CMoteur ::postPhysique (double *dt*) [virtual]

fonction de frame du moteur

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Références ayoaron ::physic ::CSolideRigide ::donn, flux, flux2, force, ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::data ::CDataNode ::getbool(), getRegime(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::audio ::CSound ::isPlaying(), parent, ayoaron ::audio ::CSound ::playSound(), ayoaron ::audio ::CSound ::setPitch(), ayoaron ::audio ::CSound ::setPos(), ayoaron ::audio ::CSound ::setVolume(), snd1, snd2, et ayoaron ::audio ::CSound ::stopSound().

Voici le graphe d'appel pour cette fonction :

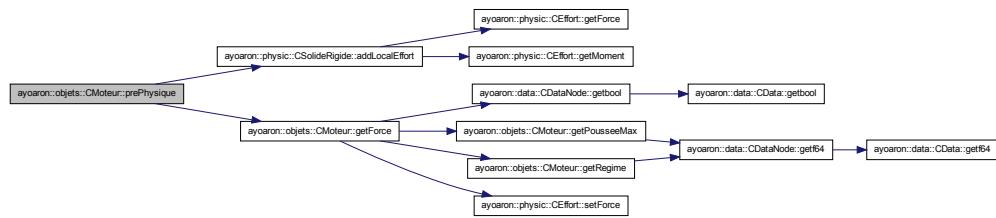
7.43.3.44 void ayoaron ::objets ::CMoteur ::prePhysique (double *dt*) [virtual]

fonction de frame du moteur

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Références ayoaron ::physic ::CSolideRigide ::addLocalEffort(), getForce(), et parent.

Voici le graphe d'appel pour cette fonction :



7.43.3.45 void ayoaron ::physic ::CSolideRigide ::resetEffort (void) [inherited]

remet les efforts à zéro

Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::momentResultant.

7.43.3.46 void ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse (irr ::core ::vector3df vit) [inherited]

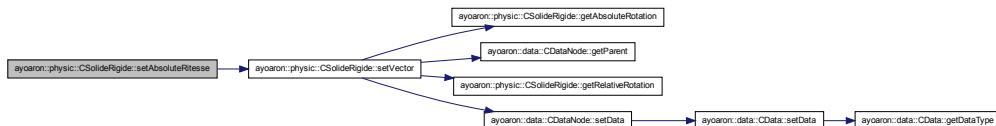
définit la vitesse de rotation du solide (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.43.3.47 void ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse (irr ::core ::vector3df vit) [inherited]

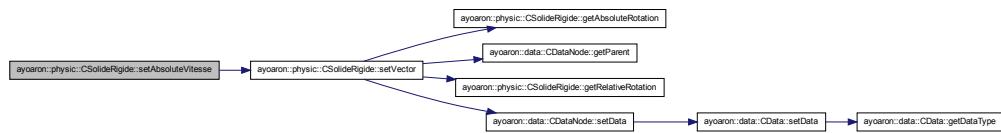
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.43.3.48 void ayoaron ::physic ::CSolideRigide ::setLocalRitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

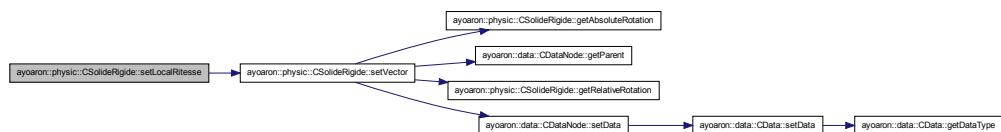
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère local
----	-----	---

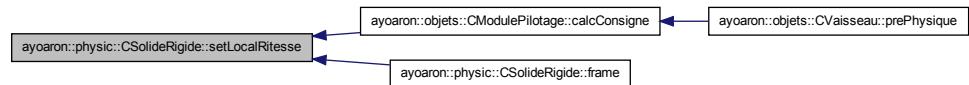
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.49 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df *vit*)
 [inherited]

définit la vitesse du solide

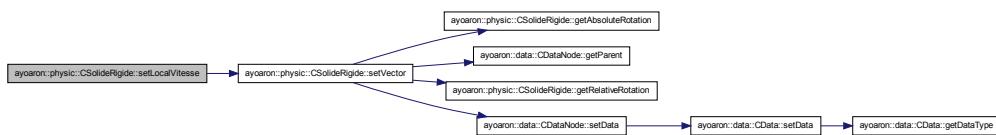
Paramètres

in	<i>vit</i>	le vecteur vitesse exprimé dans le repère local
----	------------	---

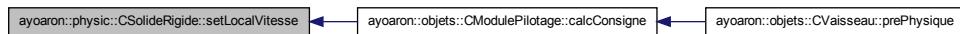
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.50 void ayoaron ::physic ::CSolideRigide ::setMatrix3 (std ::string *nom*, REPÈRE *source*, irr ::core ::matrix4 *mat*, REPÈRE *reperstockage*) [protected, inherited]

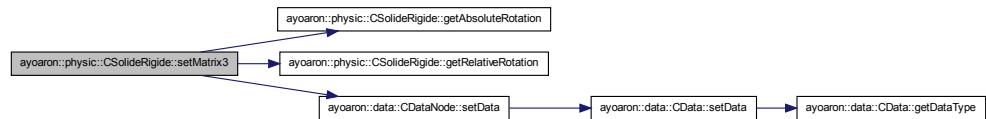
fonction permettant le changement de repère des matrices

Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>mat</i>	la matrice à stocker
<i>reperstockage</i>	le repère de stockage du vecteur

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.43.3.51 void ayoaron ::physic ::CSolideRigide ::setParent (CSolideRigide * par)
 [inherited]

change le parent de cet objet

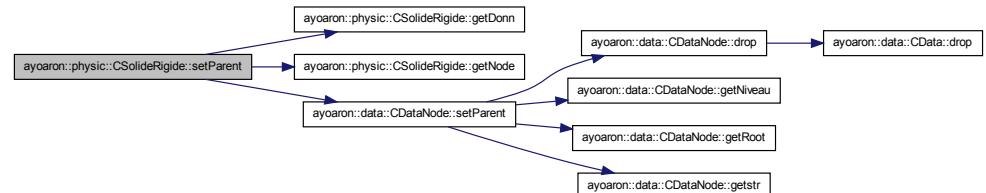
Paramètres

in	par	le nouveau parent
----	-----	-------------------

Références ayoaron ::physic ::CSolideRigide :::donn, ayoaron ::physic ::CSolideRigide :::getDonn(), ayoaron ::physic ::CSolideRigide :::getNode(), ayoaron ::physic ::CSolideRigide :::node, ayoaron ::physic ::CSolideRigide :::parent, et ayoaron ::data ::CDataNode :::setParent().

Référencé par ayoaron ::physic ::CSolideRigide :::addEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



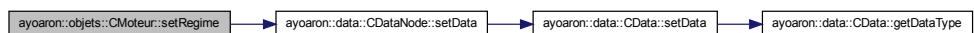
7.43.3.52 void ayoaron ::objets ::CMoteur ::setRegime (irr ::f64 *r*)

définit le régime moteur [0..1]

Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::setData().

Référencé par loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

7.43.3.53 void ayoaron ::physic ::CSolideRigide ::setRelativePosition (irr ::core ::vector3df *pos*) [inherited]

définit la position de l'objet par rapport à son parent

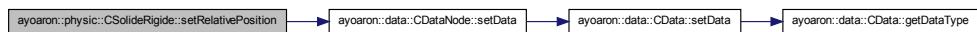
Paramètres

<i>pos</i>	la position relative de l'objet
------------	---------------------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.54 void ayoaron ::physic ::CSolideRigide ::setRelativeRitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

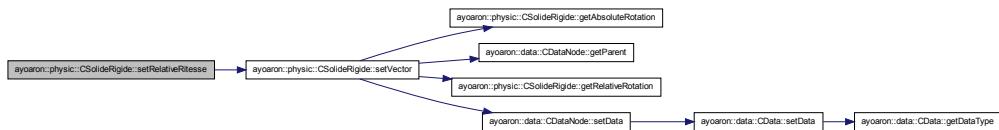
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère du parent
----	-----	---

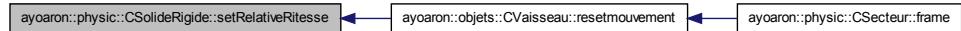
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.55 void ayoaron ::physic ::CSolideRigide ::setRelativeRotation (irr ::core ::vector3df rot) [inherited]

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.

Paramètres

<i>rot</i>	le vecteur rotation
------------	---------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.56 void ayoaron ::physic ::CSolideRigide ::setRelativeVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

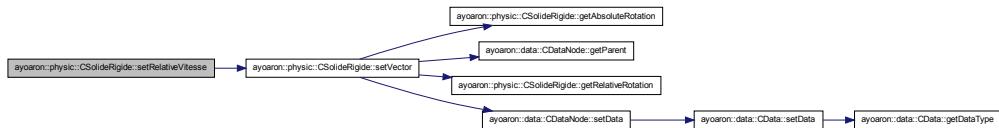
Paramètres

<i>in</i>	<i>vit</i>	le vecteur vitesse exprimé dans le repère du parent
-----------	------------	---

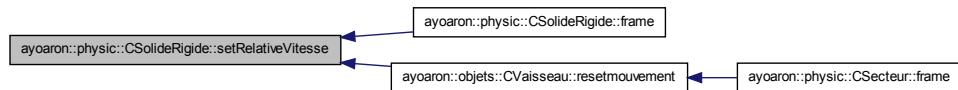
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.3.57 void ayoaron ::physic ::CSolideRigide ::setVector (std ::string *nom*, REPÈRE *source*, irr ::core ::vector3df *vec*, REPÈRE *reperestockage*) [protected, inherited]

fonction permettant le changement de repère des vecteurs

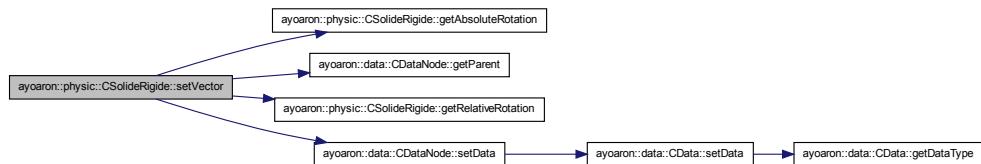
Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>vec</i>	le vecteur à stocker
<i>reperestockage</i>	le repère de stockage du vecteur

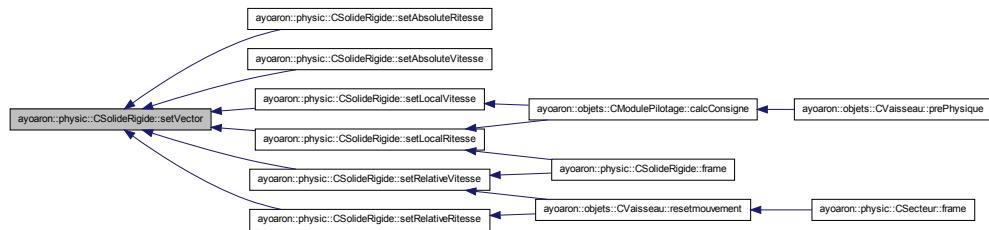
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setLocalVitesse(), ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.43.4 Documentation des données membres

7.43.4.1 irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CSolideRigide ::cam
 [protected, inherited]

pointeur vers le node de camera

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.43.4.2 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::caml
 [protected, inherited]

direction du regard de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.43.4.3 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camp
 [protected, inherited]

position de la caméra dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.43.4.4 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camup
 [protected, inherited]

vecteur de la verticale de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.43.4.5 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::donn
 [protected, inherited]

pointeur vers la structure de donnée

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getDonn(), getForce(), ayoaron ::physic ::CSolideRigide ::getMasse(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), getPoussee(), getPousseeMax(), getRegime(), ayoaron ::physic ::CSolideRigide ::getVector(), isReversible(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), loadObjetData(), ayoaron ::objets ::CAsteroide ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), ayoaron ::physic ::CSolideRigide ::setParent(), setRegime(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::setVector().

7.43.4.6 irr ::scene ::IVolumeLightSceneNode* ayoaron ::objets ::CMoteur ::flux
[protected]

le flux de particule

Référencé par CMoteur(), loadObjetData(), et postPhysique().

7.43.4.7 irr ::scene ::IVolumeLightSceneNode * ayoaron ::objets ::CMoteur ::flux2
[protected]

Référencé par CMoteur(), loadObjetData(), et postPhysique().

7.43.4.8 std ::map<int,physic ::CEffort> ayoaron ::objets ::CMoteur ::force
[protected]

liste des forces (1 seule ou 2 si revesible)

Référencé par getForce(), getPoussee(), loadObjetData(), et postPhysique().

7.43.4.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::forceResultante
[protected, inherited]

le vecteur de la résultante des efforts, il est exprimé dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getLocalForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.43.4.10 bool ayoaron ::physic ::CSolideRigide ::hascamera [protected,
inherited]

interupteur de caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.43.4.11 std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::sic ::CSolideRigide ::l_enfant [protected, inherited]

liste des objets enfants

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getEnfant(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd(), ayoaron ::physic ::CSolideRigide ::getNbEnfant(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

7.43.4.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::momentResultant [protected, inherited]

le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.43.4.13 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::node [protected, inherited]

pointeur vers le maillage et le scene node

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::getRelativePosition(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setParent(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), et ayoaron ::physic ::CSolideRigide ::setRelativeRotation().

7.43.4.14 objets ::CVaisseau* ayoaron ::objets ::CMoteur ::parent [protected]

pointeur vers le vaisseau parent

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Référencé par CMoteur(), loadObjetData(), postPhysique(), et prePhysique().

7.43.4.15 CPousMouv ayoaron ::objets ::CMoteur ::PP [protected]

tableau de capacité du moteur

Référencé par getPP(), et loadObjetData().

7.43.4.16 CPousMouv ayoaron ::objets ::CMoteur ::PP2 [protected]

Référencé par getPP2(), et loadObjetData().

7.43.4.17 audio ::CSound* ayoaron ::objets ::CMoteur ::snd1 [protected]

pointeur vers le son du moteur

Référencé par CMoteur(), loadObjetData(), et postPhysique().

7.43.4.18 audio ::CSound * ayoaron ::objets ::CMoteur ::snd2 [protected]

Référencé par CMoteur(), loadObjetData(), et postPhysique().

7.43.4.19 double ayoaron ::objets ::CMoteur ::temps [protected]

7.43.4.20 physic ::CSecteur* ayoaron ::objets ::CMoteur ::world [protected]

pointeur vers le secteur parent

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Référencé par CMoteur(), et loadObjetData().

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

- [CMoteur.h](#)
- [CMoteur.cpp](#)

7.44 Référence de la classe ayoaron ::objets ::CPousMouv

une classe de vecteur un peu particulière spécifique au pilotage des moteurs

```
#include <CMoteur.h>
```

Fonctions membres publiques

- **CPousMouv abs ()**
renvoie la valeur absolue de chaque terme
- **CPousMouv ()**
constructeur
- **CPousMouv h ()**
revoie 1 pour chaque terme non nul
- **double max ()**
renvoie la plus grande valeur
- **CPousMouv operator* (double b)**
opérateur de multiplication par un scalaire
- **CPousMouv operator* (CPousMouv b)**
opérateur de multiplication terme à terme (ni produit scalaire ni produit vectoriel)
- **CPousMouv operator+ (CPousMouv b)**
opérateur d'addition terme à terme
- **CPousMouv operator- (CPousMouv b)**
opérateur de soustraction terme à terme

- **CPousMouv operator/ (CPousMouv b)**
opérateur de division terme à terme renvoie 0 si divise par 0
- double & **operator[]** (int i)
opérateur d'accès à une valeur
- double **sum** ()
renvoie la somme de tous les termes
- virtual **~CPousMouv** ()
destructeur

Attributs privés

- double **P** [12]
les valeurs

7.44.1 Description détaillée

une classe de vecteur un peu particulière spécifique au pilotage des moteurs

7.44.2 Documentation des constructeurs et destructeur

7.44.2.1 ayoaron ::objets ::CPousMouv ::CPousMouv ()

constructeur

Références P.

7.44.2.2 virtual ayoaron ::objets ::CPousMouv ::~CPousMouv () [inline, virtual]

destructeur

7.44.3 Documentation des fonctions membres

7.44.3.1 CPousMouv ayoaron ::objets ::CPousMouv ::abs ()

renvoie la valeur absolue de chaque terme

Références P.

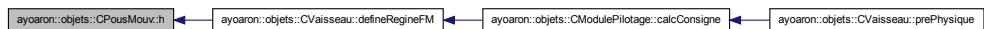
7.44.3.2 CPousMouv ayoaron ::objets ::CPousMouv ::h ()

revoie 1 pour chaque terme non nul

Références P.

Référencé par ayoaron ::objets ::CVaisseau ::defineRegineFM().

Voici le graphe d'appel pour cette fonction :



7.44.3.3 double ayoaron ::objets ::CPousMouv ::max ()

renvoie la plus grande valeur

Références P.

7.44.3.4 CPousMouv ayoaron ::objets ::CPousMouv ::operator* (CPousMouv b)

opérateur de multiplication terme à terme (ni produit scalaire ni produit vectoriel)

Références P.

7.44.3.5 CPousMouv ayoaron ::objets ::CPousMouv ::operator* (double b)

opérateur de multiplication par un scalaire

Références P.

7.44.3.6 CPousMouv ayoaron ::objets ::CPousMouv ::operator+ (CPousMouv b)

opérateur d'addition terme à terme

Références P.

7.44.3.7 CPousMouv ayoaron ::objets ::CPousMouv ::operator- (CPousMouv b)

opérateur de soustraction terme à terme

Références P.

7.44.3.8 CPousMouv ayoaron ::objets ::CPousMouv ::operator/ (CPousMouv b)

opérateur de division terme à terme renvoie 0 si divise par 0

Références P.

7.44.3.9 double & ayoaron ::objets ::CPousMouv ::operator[] (int i)

opérateur d'accès à une valeur

Références P.

7.44.3.10 double ayoaron ::objets ::CPousMouv ::sum ()

renvoie la somme de tous les termes

Références P.

7.44.4 Documentation des données membres

7.44.4.1 double ayoaron ::objets ::CPousMouv ::P[12] [private]

les valeurs

Référencé par abs(), CPousMouv(), h(), max(), operator*(), operator+(), operator-(), operator/(), operator[](), et sum().

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

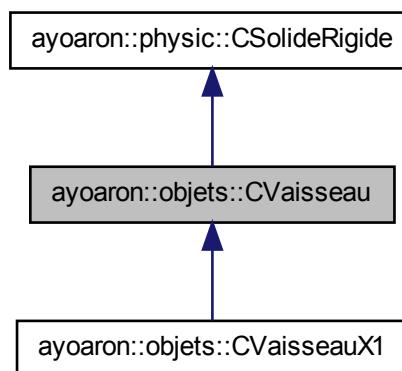
- [CMoteur.h](#)
- [CMoteur.cpp](#)

7.45 Référence de la classe ayoaron ::objets ::CVaisseau

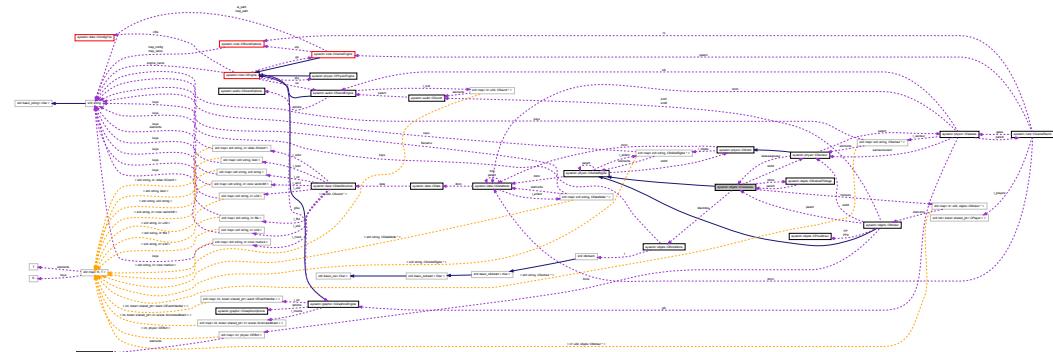
Classe de base de description d'un vaisseau quelconque.

```
#include <CVaisseau.h>
```

Graphe d'héritage de ayoaron ::objets ::CVaisseau :



Graphe de collaboration de ayoaron ::objets ::CVaisseau ::



Fonctions membres publiques

- void **addAbsoluteEffort** (CEffort F)
 - ajoute un effort exprimé dans le repère global*
 - void **addEnfant** (std : :string nom, CSolideRigide *enfant)
 - ajoute un nouvel enfant (s'il n'existe pas déjà)*
 - void **addLocalEffort** (CEffort F)
 - ajoute un effort exprimé dans le repère local*
 - void **addpoussee** (irr : :u32 moteurId, irr : :f64 regime)
 - ajoute une composante de poussée*
 - void **addRelativeEffort** (CEffort F)
 - ajoute un effort exprimé dans le repère du parent*
 - **CVaisseau** (data : :CDataNode *d, physic : :CSecteur *p, physic : :CSolideRigide *po=NULL, irr : :s32 ID=-1)
 - Constructeur.*
 - virtual void **defineRegineFM** (irr : :core : :vector3df f, irr : :core : :vector3df m)
 - defini les régimes moteur en fonction de la consigne de force et vitesse*
 - void **dropEnfant** (std : :string nom)
 - supprime un enfant de la liste*
 - void **frame** (double dt)
 - effectue les diverses opération de mise à jour*
 - irr : :core : :vector3df **getAbsoluteForceResultante** ()
 - renvoie la force résultante exprimée dans le repère global*
 - irr : :core : :vector3df **getAbsoluteMomentResultante** ()
 - renvoie le moment résultant exprimé dans le repère global*
 - irr : :core : :vector3df **getAbsolutePosition** ()
 - renvoie la position de l'objet dans le monde*
 - irr : :core : :vector3df **getAbsoluteRitesse** ()
 - renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entraînement du repère parent)*
 - irr : :core : :matrix4 **getAbsoluteRotation** ()
 - renvoie la matrice de rotation entre le repère global et le repère local*
 - irr : :core : :vector3df **getAbsoluteVitesse** ()
 - renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entraînement du repère parent)*
 - data : :CDataNode * **getDonn** ()
 - acces aux données*
 - CSolideRigide* **getEnfant** (std : :string nom)
 - renvoie l'enfant donné*

- irr ::core ::matrix4 **getGlobalInertie** ()

renvoie la matrice d'inertie
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantBegin** ()

renvoie un itérateur sur les enfant pointant au début de la map
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantEnd** ()

renvoie un itérateur sur les enfant pointant à la fin de la map
- irr ::core ::vector3df **getLocalForceResultante** ()

renvoie la force résultante exprimée dans le repère local
- irr ::core ::matrix4 **getLocalInertie** ()

renvoie la matrice d'inertie
- irr ::core ::vector3df **getLocalMomentResultante** ()

renvoie le moment résultant exprimé dans le repère local
- irr ::core ::vector3df **getLocalRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère local
- irr ::core ::vector3df **getLocalVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère local
- irr ::f64 **getMasse** ()

fonction qui renvoie la masse du solide
- irr ::u32 **getNbEnfant** ()

renvoie le nombre d'enfants de cet objet
- irr ::scene ::ISceneNode * **getNode** ()

renvoie le scene node et le maillage associé
- irr ::core ::vector3df **getRelativeForceResultante** ()

renvoie la force résultante exprimée dans le repère du parent
- irr ::core ::matrix4 **getRelativeInertie** ()

renvoie la matrice d'inertie
- irr ::core ::vector3df **getRelativeMomentResultante** ()

renvoie le moment résultant exprimé dans le repère du parent
- irr ::core ::vector3df **getRelativePosition** ()

renvoie la position de l'objet par rapport à l'objet parent
- irr ::core ::vector3df **getRelativeRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère du parent
- irr ::core ::matrix4 **getRelativeRotation** ()

renvoie la matrice de rotation entre le repère relatif et le repère local
- irr ::core ::vector3df **getRelativeVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère du parent
- virtual void **loadObjetData** (data ::CDataNode *d=NULL)

charge l'objet
- virtual void **postPhysique** (double dt)

effectue les diverses opération de mise à jour après le calcul de la physique
- void **prePhysique** (double dt)

fonction de mise à jour
- virtual void **resetconsigne** ()

reset les consignes
- void **resetEffort** ()

remet les efforts à zéro
- void **resetmouvement** (void)

reset moteur
- void **resetpousse** (void)

reset moteur
- void **setAbsoluteRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide (tient compte de la vitesse d'entraînement du repère parent)
- void **setAbsoluteVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
- void **setLocalRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide
- void **setLocalVitesse** (irr ::core ::vector3df vit)

- définit la vitesse du solide
- void **setParent** (CSolideRigide *par)
change le parent de cet objet
- void **setRelativePosition** (irr ::core ::vector3df pos)
définit la position de l'objet par rapport à son parent
- void **setRelativeRitesse** (irr ::core ::vector3df vit)
définit la vitesse de rotation du solide
- void **setRelativeRotation** (irr ::core ::vector3df rot)
à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.
- void **setRelativeVitesse** (irr ::core ::vector3df vit)
définit la vitesse du solide
- void **toggleconsigne** (void)
interrupteur de consigne
- virtual void **vitavar** (irr ::f32 cs)
ajoute à la consigne de vitesse de translation avant-arrière
- virtual void **vitadr** (irr ::f32 cs)
ajoute à la consigne de vitesse de translation gauche droite
- virtual void **vithaba** (irr ::f32 cs)
ajoute à la consigne de vitesse de translation haut bas
- virtual void **vitlace** (irr ::f32 cs)
ajoute à la consigne de vitesse de rotation lacet
- virtual void **vitroul** (irr ::f32 cs)
ajoute à la consigne de vitesse de rotation roulis
- virtual void **vittang** (irr ::f32 cs)
ajoute à la consigne de vitesse de rotation tangage
- virtual ~**CVaisseau** ()
Destructeur.

Attributs publics

- **CBoiteNoire** * **blackbox**
pointeur vers une boite noire
- irr ::core ::vector3df **DTV**
- irr ::core ::vector3df **DTVR**

Fonctions membres protégées

- irr ::core ::matrix4 **getMatrix3** (std ::string nom, REPERE ou, REPERE reperestockage)
renvoie le vecteur donné dans la base voulue
- irr ::core ::vector3df **getVector** (std ::string nom, REPERE ou, REPERE reperestockage)
renvoie le vecteur donné dans la base voulue
- void **setMatrix3** (std ::string nom, REPERE source, irr ::core ::matrix4 mat, REPERE reperestockage)
fonction permettant le changement de repère des matrices
- void **setVector** (std ::string nom, REPERE source, irr ::core ::vector3df vec, REPERE reperestockage)
fonction permettant le changement de repère des vecteurs

Attributs protégés

- irr ::scene ::ICameraSceneNode * **cam**
pointeur vers le node de camera
- irr ::core ::vector3df **caml**

direction du regard de la caméra
 – irr ::core ::vector3df **camp**
position de la caméra dans le repère local
 – irr ::core ::vector3df **camup**
vecteur de la verticale de la caméra
 – bool **consigne**
interrupteur de la commande de consigne
 – irr ::core ::vector3df **consignerotvitesse**
consigne de vitesse de rotation
 – irr ::core ::vector3df **consignevitesse**
consigne de vitesse de translation
 – **data** ::CDataNode * **donn**
pointeur vers la structure de donnée
 – irr ::core ::vector3df **forceResultante**
le vecteur de la résultante des efforts, il est exprimé dans le repère local
 – bool **hascamera**
interrupteur de caméra
 – std ::map< std ::string, CSolideRigide * > **l_enfant**
liste des objets enfants
 – irr ::core ::vector3df **momentResultant**
le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel
G=(0,0,0)
 – std ::map< irr ::u32, objets ::CMoteur * > **moteurs**
liste des moteurs
 – irr ::u32 **Nmoteur**
nombre de moteurs
 – irr ::scene ::ISceneNode * **node**
pointeur vers le maillage et le scene node
 – CSolideRigide * **parent**
pointeur vers l'objet parent
 – **CModulePilotage** * **pilote**
module de pilotage
 – irr ::f32 **vitmax**
vitesse max
 – **physic** ::CSecteur * **world**
sur déclaration de world en CSecteur (dérivant d'un CWorld)

7.45.1 Description détaillée

Classe de base de description d'un vaisseau quelconque.

7.45.2 Documentation des constructeurs et destructeur

7.45.2.1 ayoaron ::objets ::CVaisseau ::CVaisseau (**data** ::CDataNode * *d*,
physic ::CSecteur * *p*, **physic** ::CSolideRigide * *po* = NULL, irr ::s32 *ID* =
-1)

Constructeur.

Références blackbox, ayoaron ::physic ::CSolideRigide ::parent, pilote, et world.

7.45.2.2 ayoaron ::objets ::CVaisseau ::~CVaisseau () [virtual]

Destructeur.

7.45.3 Documentation des fonctions membres

7.45.3.1 `void ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort (CEffort F)`
 [inherited]

ajoute un effort exprimé dans le repère global

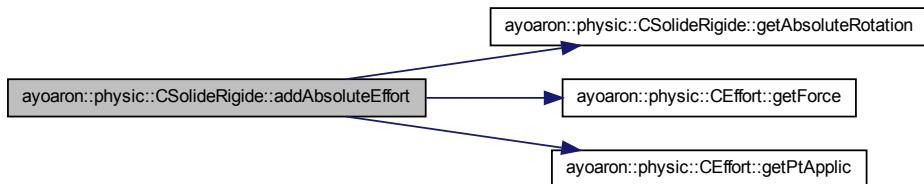
Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère global, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	----------	---

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.2 `void ayoaron ::physic ::CSolideRigide ::addEnfant (std ::string nom,
 CSolideRigide * enfant)` [inherited]

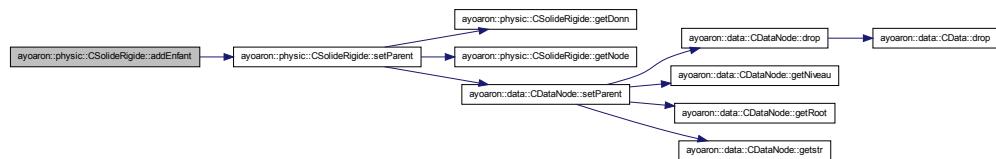
ajoute un nouvel enfant (s'il n'existe pas déjà)

Paramètres

<i>nom</i>	le nom (unique) de l'enfant
<i>enfant</i>	pointeur vers l'enfant en question

Références ayoaron ::physic ::CSolideRigide ::l_enfant, et ayoaron ::physic ::CSolideRigide :::setParent().

Voici le graphe d'appel pour cette fonction :



7.45.3.3 void ayoaron ::physic ::CSolideRigide :::addLocalEffort (CEffort F) [inherited]

ajoute un effort exprimé dans le repère local

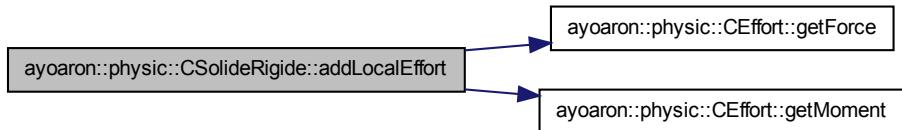
Paramètres

in	<i>F</i>	l'effort à ajouter exprimé dans le repère local, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron ::physic ::CSolideRigide :::forceResultante, ayoaron ::physic ::CEffort :::getForce(), ayoaron ::physic ::CEffort :::getMoment(), et ayoaron ::physic ::CSolideRigide :::momentResultant.

Référencé par ayoaron ::objets ::CMoteur :::prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.4 void ayoaron ::objets ::CVaisseau ::addpoussee (irr ::u32 moteurId, irr ::f64 regime)

ajoute une composante de poussée

Références moteurs, et Nmoteur.

7.45.3.5 void ayoaron ::physic ::CSolideRigide ::addRelativeEffort (CEffort F) [inherited]

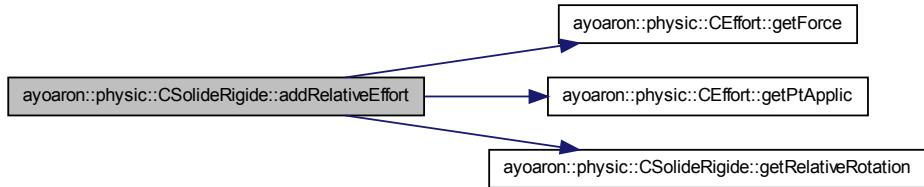
ajoute un effort exprimé dans le repère du parent

Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère du parent, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	----------	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.45.3.6 void ayoaron ::objets ::CVaisseau ::defineRegimeFM (irr ::core ::vector3df f, irr ::core ::vector3df m) [virtual]

defini les régimes moteur en fonction de la consigne de force et vitesse

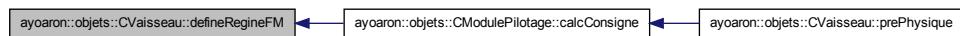
Références ayoaron ::objets ::CPousMouv ::h(), et moteurs.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.7 void ayoaron ::physic ::CSolideRigide ::dropEnfant (std ::string *nom*)
 [inherited]

supprime un enfant de la liste

Paramètres

<i>nom</i>	nom de l'enfant à supprimer
------------	-----------------------------

7.45.3.8 void ayoaron ::physic ::CSolideRigide ::frame (double *dt*) [inherited]

effectue les diverses opération de mise à jour

Paramètres

in	<i>dt</i>	le pas de temps
----	-----------	-----------------

la frame :

tout d'abord, l'effort global

on pre frame tous les objets enfants

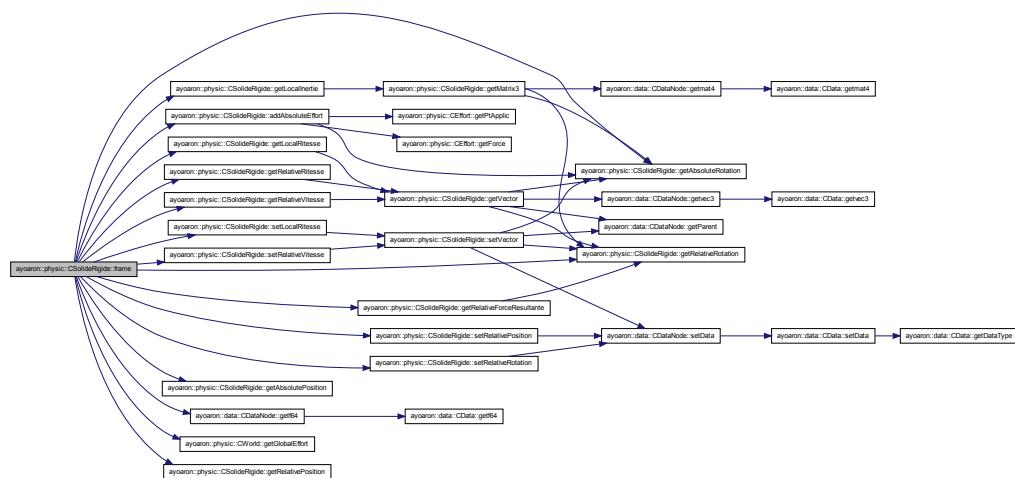
la translation, calculée dans le repère du parent

la rotation, calculée dans le repère parent

on post frame tous les objets enfants

Références ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::cam, ayoaron ::physic ::CSolideRigide ::caml, ayoaron ::physic ::CSolideRigide ::camp, ayoaron ::physic ::CSolideRigide ::camup, ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getf64(), ayoaron ::physic ::CWorld ::getGlobalEffort(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativePosition(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeVitesse(), ayoaron ::physic ::CSolideRigide ::hascamera, ayoaron ::physic ::CSolideRigide ::l_enfant, ayoaron ::physic ::CSolideRigide ::momentResultant, ayoaron ::physic ::CSolideRigide ::node, ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), ayoaron ::physic ::CSolideRigide ::setRelativeVitesse(), et ayoaron ::physic ::CSolideRigide ::world.

Voici le graphe d'appel pour cette fonction :



7.45.3.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère global

Renvoie

la résultante des efforts exprimée dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation().

Voici le graphe d'appel pour cette fonction :



7.45.3.10 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante()` [inherited]

renvoie le moment résultant exprimé dans le repère global

Renvoie

le moment résultant exprimé dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.45.3.11 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsolutePosition()` [inherited]

renvoie la position de l'objet dans le monde

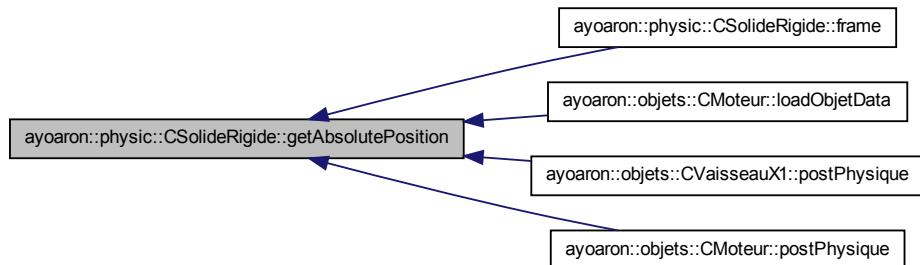
Renvoie

le vecteur position absolue de l'objet

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::objets ::CMoteur ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.45.3.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse ()
 [inherited]

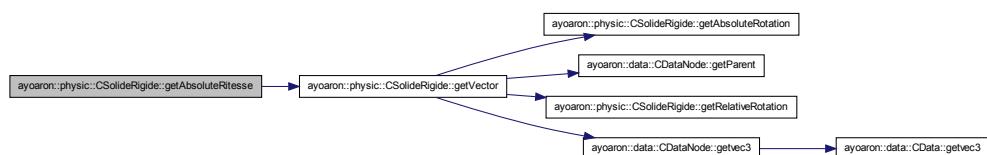
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse de rotation dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.45.3.13 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation ()
 [inherited]

renvoie la matrice de rotation entre le repère global et le repère local

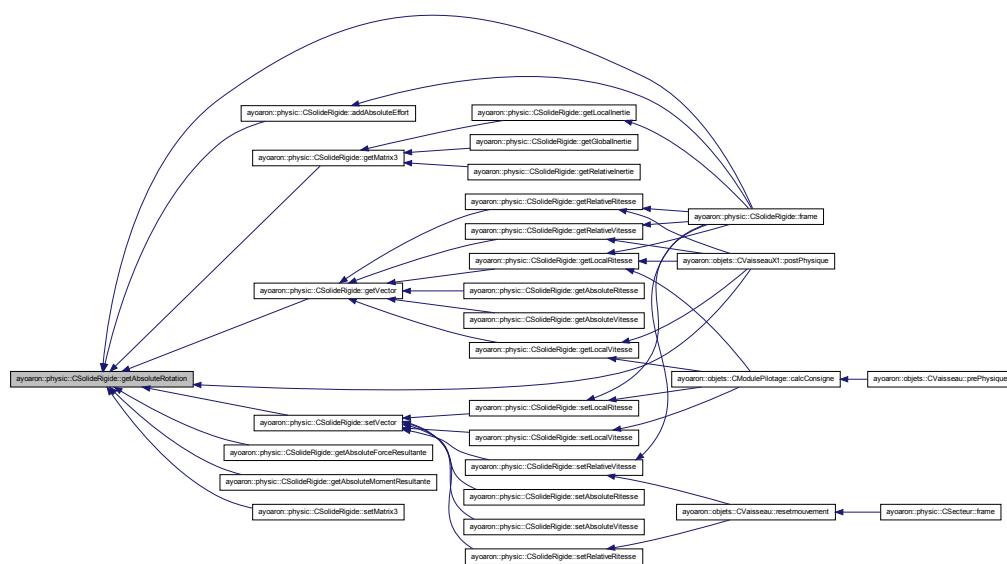
Renvoie

la matrice de rotation du repère global au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.45.3.14 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse ()
[inherited]

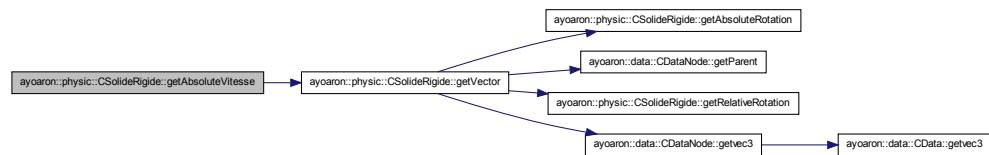
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.45.3.15 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::getDonn ()
 [inline, inherited]

acces aux données

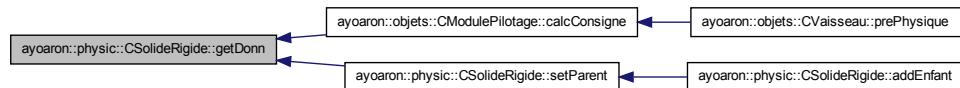
Renvoie

pointeur vers la structure de donnée

Références ayoaron ::physic ::CSolideRigide ::donn.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.45.3.16 CSolideRigide * ayoaron ::physic ::CSolideRigide ::getEnfant (std ::string nom)
 [inherited]

renvoie l'enfant donné

Paramètres

nom	le nom de l'enfant voulu
-----	--------------------------

Renvoie

pointeur vers l'enfant désiré, renvoie NULL si l'enfant n'existe pas

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.45.3.17 `irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getGlobalInertie()`
 [inline, inherited]

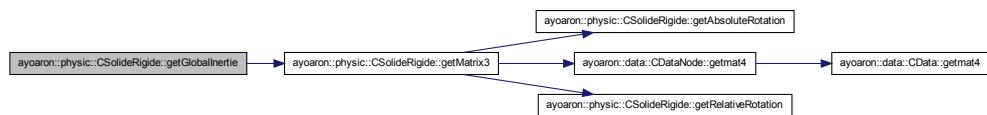
renvoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Voici le graphe d'appel pour cette fonction :



7.45.3.18 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin()` [inherited]

renvoie un itérатор sur les enfant pointant au début de la map

Renvoie

iterator pointant au début de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.45.3.19 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd()` [inherited]

renvoie un itérатор sur les enfant pointant à la fin de la map

Renvoie

iterator pointant à la fin de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.45.3.20 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalForceResultante()` [inherited]

renvoie la force résultante exprimée dans le repère local

Renvoie

la résultante des efforts exprimée dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::forceResultante.

7.45.3.21 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getLocalInertie ()
[inline, inherited]

renvoie la matrice d'inertie

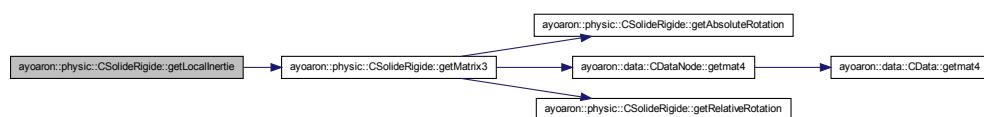
Renvoie

la matrice d'inertie dans le repère local

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.22 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère local

Renvoie

le moment résultant exprimé dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::momentResultant.

7.45.3.23 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalRitesse () [inherited]

renvoie l'expression de la vitesse de rotation dans le repère local

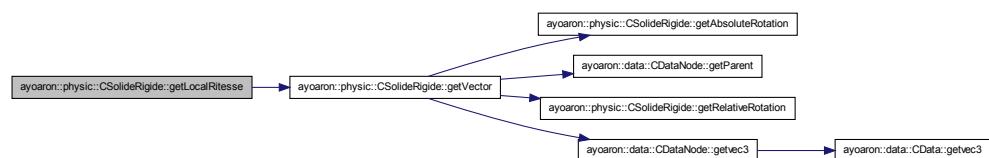
Renvoie

le vecteur vitesse de rotation dans le repère local

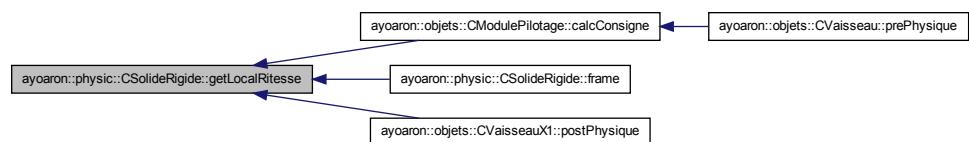
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.24 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalVitesse () [inherited]

renvoie l'expression de la vitesse de translation dans le repère local

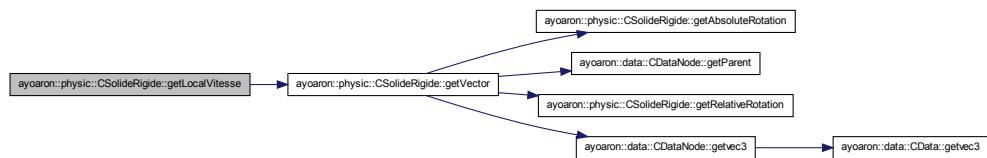
Renvoie

le vecteur vitesse dans le repère local

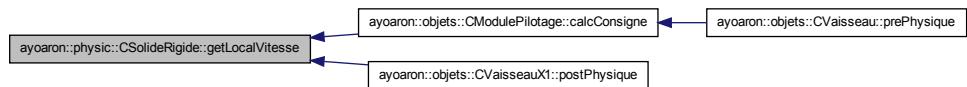
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.25 irr : :f64 ayoaron ::physic ::CSolideRigide ::getMasse () [inline, inherited]

fonction qui renvoie la masse du solide

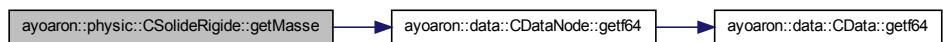
Renvoie

la masse du solide

Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.26 `irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getMatrix3 (std ::string nom,
REPERE ou, REPERE reperestockage)` [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

<i>nom</i>	nom de la clé dans la base de données
<i>ou</i>	le repère dans lequel on veut le vecteur
<i>reperestockage</i>	le repère dans lequel est stocker le vecteur

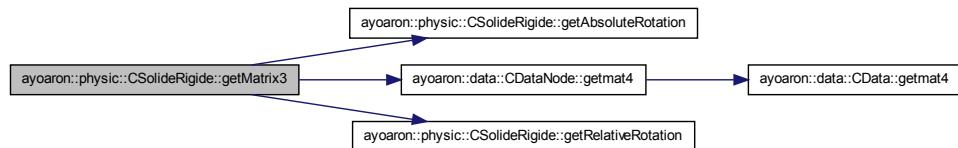
Renvoie

le vecteur voulu dans la base voulue

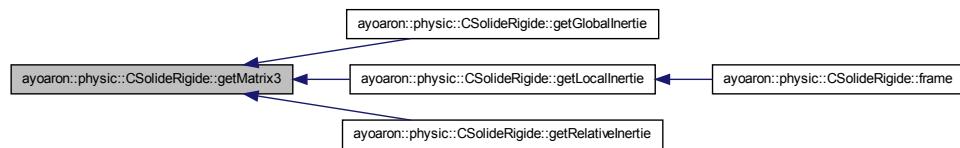
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getmat4(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getGlobalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), et ayoaron ::physic ::CSolideRigide ::getRelativeInertie().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.27 `irr ::u32 ayoaron ::physic ::CSolideRigide ::getNbEnfant ()` [inline, inherited]

renvoie le nombre d'enfants de cet objet

Renvoie

le nombre d'enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.45.3.28 `irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::getNode ()` [inline, inherited]

renvoie le scene node et le maillage associé

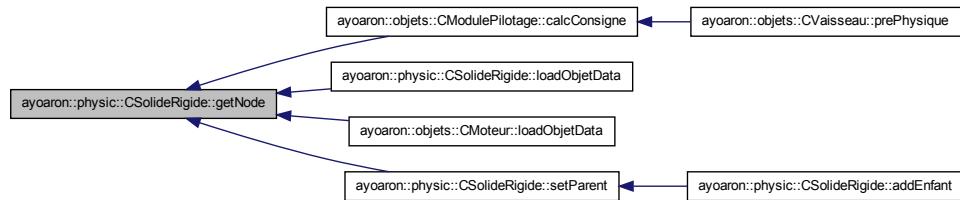
Renvoie

pointeur vers le IAnimatedMeshSceneNode

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.45.3.29 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante()
[inherited]

renvoie la force résultante exprimée dans le repère du parent

Renvoie

la résultante des efforts exprimée dans le référentiel du parent

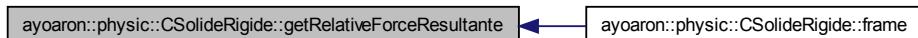
Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getRelativeRotation().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.30 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeInertie()
[inline, inherited]

renvoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère du parent

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.45.3.31 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère du parent

Renvoie

le moment résultant exprimé dans le référentiel du parent

Références ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.45.3.32 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativePosition () [inherited]

renvoie la position de l'objet par rapport à l'objet parent

Renvoie

le vecteur position relative au parent

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.33 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeRitesse ()
 [inherited]

renvoie l'expression de la vitesse de rotation dans le repère du parent

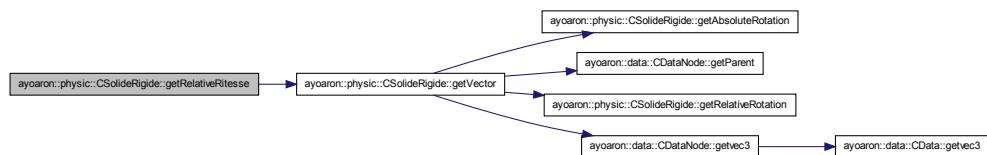
Renvoie

le vecteur vitesse de rotation dans le repère du parent

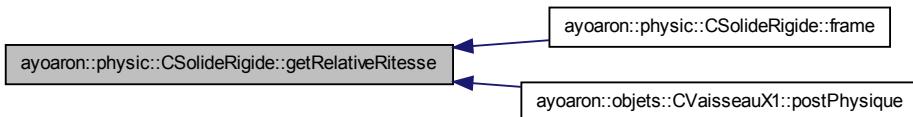
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.34 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeRotation ()
[inherited]

renvoie la matrice de rotation entre le repère relatif et le repère local

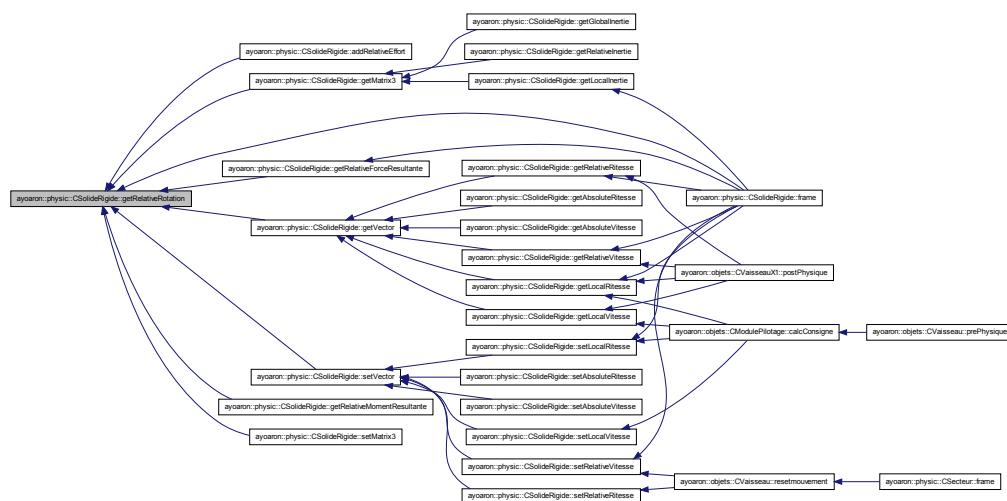
Renvoie

la matrice de rotation du repère relatif au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.45.3.35 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeVitesse ()
[inherited]

renvoie l'expression de la vitesse de translation dans le repère du parent

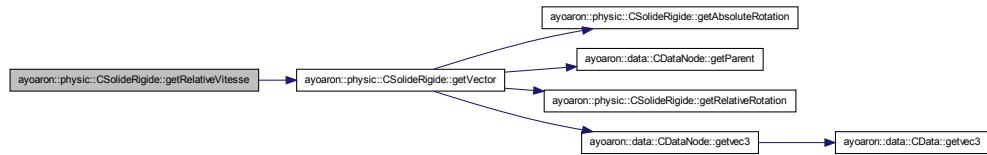
Renvoie

le vecteur vitesse dans le repère du parent

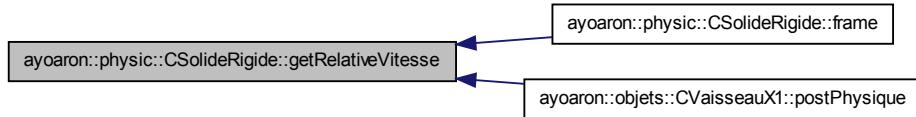
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.36 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getVector (std ::string nom, REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

<i>nom</i>	nom de la clé dans la base de données
<i>ou</i>	le repère dans lequel on veut le vecteur
<i>reperestockage</i>	le repère dans lequel est stocker le vecteur

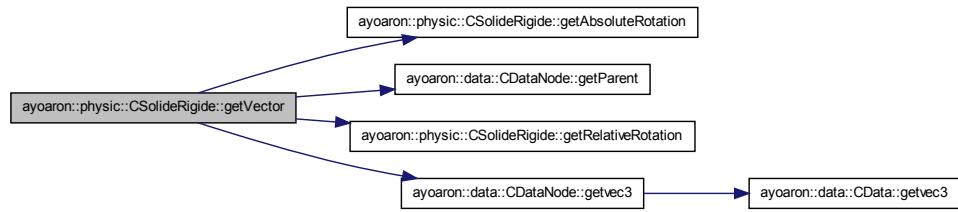
Renvoie

le vecteur voulu dans la base voulue

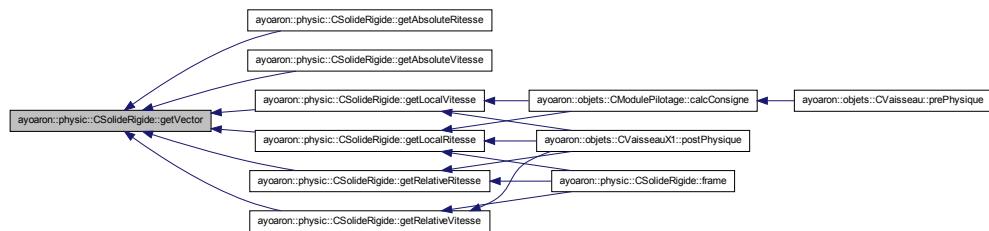
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getLocalVitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::getRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.37 `virtual void ayoaron ::objets ::CVaisseau ::loadObjetData (data :: :CDataNode * d = NULL) [inline, virtual]`

charge l'objet

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Réimplémentée dans [ayoaron ::objets ::CVaisseauX1](#).

7.45.3.38 `virtual void ayoaron ::physic ::CSolideRigide ::postPhysique (double dt) [inline, virtual, inherited]`

effectue les diverses opération de mise à jour après le calcul de la physique

Paramètres

in	dt	le pas de temps
----	----	-----------------

Réimplémentée dans [ayoaron ::objets ::CMoteur](#), et [ayoaron ::objets ::CVaisseauX1](#).

7.45.3.39 `void ayoaron ::objets ::CVaisseau ::prePhysique (double dt) [virtual]`

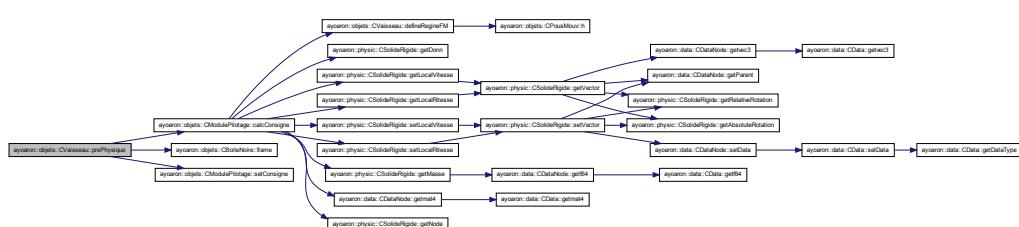
fonction de mise à jour

calcul de la consigne moteur

Réimplémentée à partir de [ayoaron](#) : `physic` : `CSolideRigide`.

Références blackbox, ayoaron : :objets : :CModulePilotage : :calcConsigne(), consigne, consignerotvitesse, consignevitesse, ayoaron : :objets : :CBoiteNoire : :frame(), pilote, et ayoaron : :objets : :CModulePilotage : :setConsigne().

Voici le graphe d'appel pour cette fonction :



7.45.3.40 void ayoaron ::objets ::CVaisseau ::resetconsigne() [virtual]

reset les consignes

Références consignerotvitesse, et consigne vitesse.

Référencé par ayoaron : physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.41 void ayoaron ::physic ::CSolideRigide ::resetEffort (void) [inherited]

remet les efforts à zéro

Références ayoaron : :physic : :CSolideRigide : :forceResultante, et ayoaron : :physic : :CSolideRigide : :momentResultant.

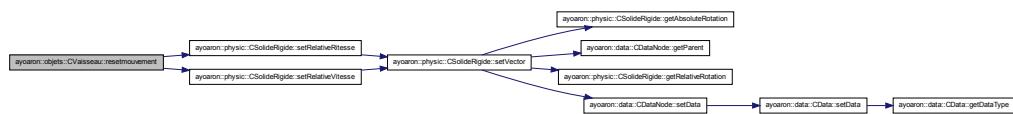
7.45.3.42 void ayoaron ::objets ::CVaisseau ::resetmouvement (void) [inline]

reset moteur

Références ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.43 void ayoaron ::objets ::CVaisseau ::resetpousse (void)

reset moteur

Références moteurs, et Nmoteur.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.44 void ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse (irr ::core ::vector3df vit) [inherited]

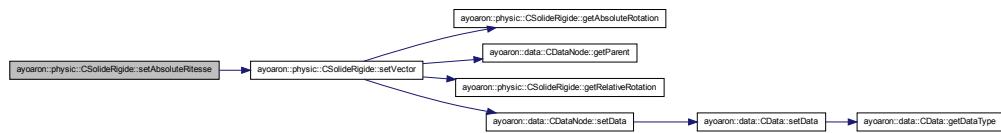
définit la vitesse de rotation du solide (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.45.3.45 void ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse (irr ::core ::vector3df vit) [inherited]

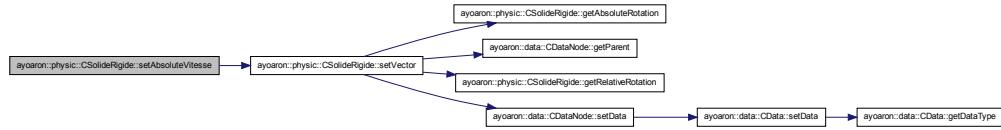
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.45.3.46 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

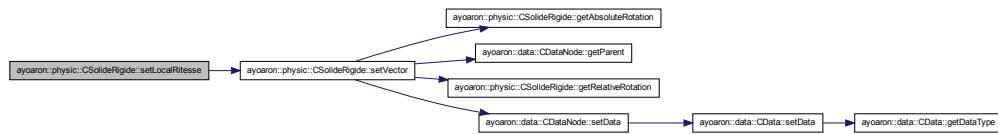
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère local
----	-----	---

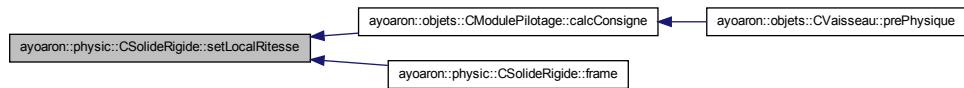
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.45.3.47 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df vit)
[inherited]**

définit la vitesse du solide

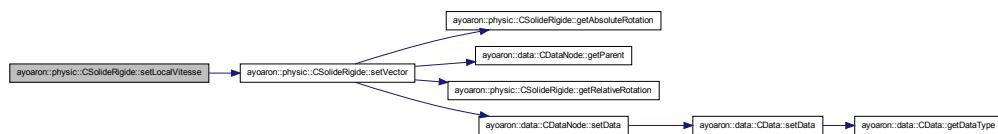
Paramètres

in	vit	le vecteur vitesse exprimé dans le repère local
----	-----	---

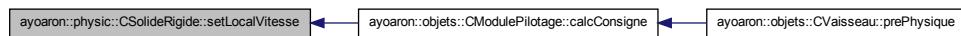
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.48 void ayoaron ::physic ::CSolideRigide ::setMatrix3 (std ::string *nom*, REPERE *source*, irr ::core ::matrix4 *mat*, REPERE *reperstockage*) [protected, inherited]

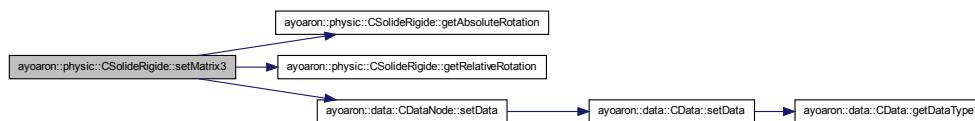
fonction permettant le changement de repère des matrices

Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>mat</i>	la matrice à stocker
<i>reperstockage</i>	le repère de stockage du vecteur

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.45.3.49 void ayoaron ::physic ::CSolideRigide ::setParent (CSolideRigide * *par*) [inherited]

change le parent de cet objet

Paramètres

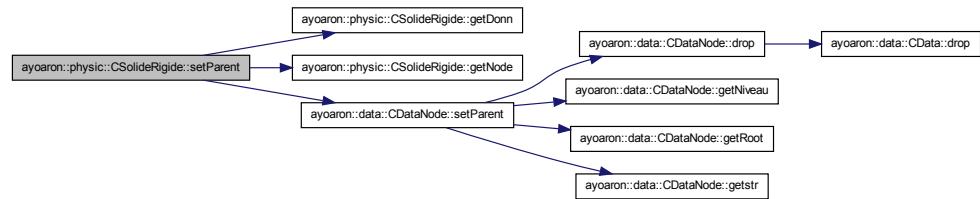
<i>in</i>	<i>par</i>	le nouveau parent
-----------	------------	-------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::getDonn(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::node, ayoaron ::physic ::CSolideRigide ::parent, et ayoaron ::data ::CData

taNode ::setParent().

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.50 void ayoaron ::physic ::CSolideRigide ::setRelativePosition (irr ::core ::vector3df pos) [inherited]

définit la position de l'objet par rapport à son parent

Paramètres

<i>pos</i>	la position relative de l'objet
------------	---------------------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.51 void ayoaron ::physic ::CSolideRigide ::setRelativeRitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

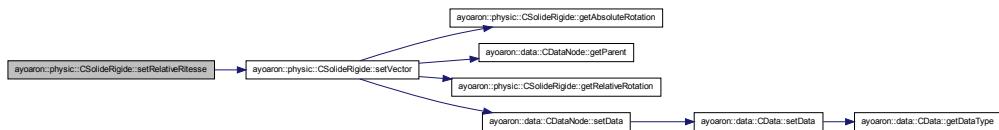
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère du parent
----	-----	---

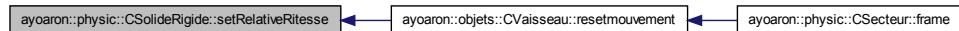
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.52 void ayoaron ::physic ::CSolideRigide ::setRelativeRotation (irr ::core ::vector3df rot) [inherited]

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.

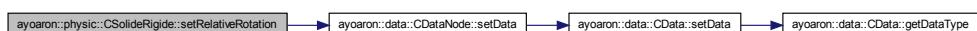
Paramètres

<i>rot</i>	le vecteur rotation
------------	---------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.53 void ayoaron ::physic ::CSolideRigide ::setRelativeVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

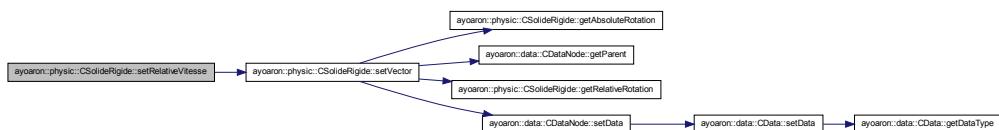
Paramètres

<i>in</i>	<i>vit</i>	le vecteur vitesse exprimé dans le repère du parent
-----------	------------	---

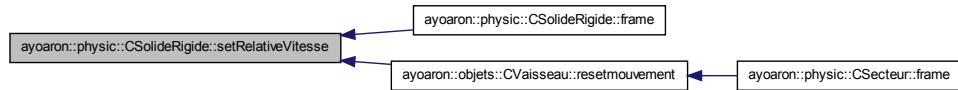
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.54 void ayoaron ::physic ::CSolideRigide ::setVector (std ::string *nom*, REPERE *source*, irr ::core ::vector3df *vec*, REPERE *reperestockage*) [protected, inherited]

fonction permettant le changement de repère des vecteurs

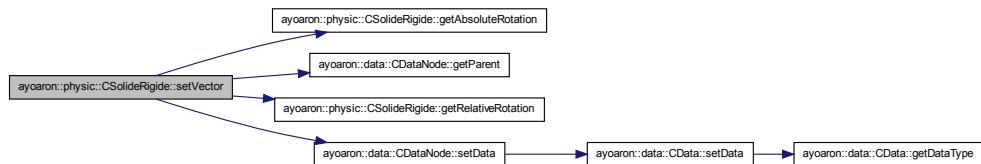
Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>vec</i>	le vecteur à stacker
<i>reperestockage</i>	le repère de stockage du vecteur

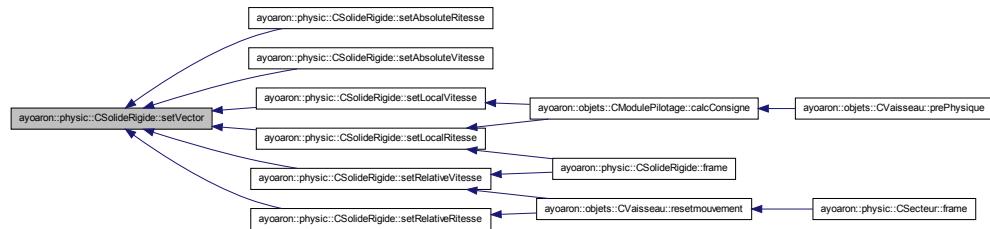
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setLocalVitesse(), ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.45.3.55 void ayoaron ::objets ::CVaisseau ::toggleconsigne (void) [inline]

interrupteur de consigne

Références consigne.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.56 void ayoaron ::objets ::CVaisseau ::vitavar (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de translation avant-arrière

Références consignevitesse, et vitmax.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.57 void ayoaron ::objets ::CVaisseau ::vitgadr (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de translation gauche droite

Références consigne vitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.58 void ayoaron ::objets ::CVaisseau ::vithaba (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de translation haut bas

Références consigne vitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.59 void ayoaron ::objets ::CVaisseau ::vitlace (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de rotation lacet

Références consigne rot vitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.60 void ayoaron ::objets ::CVaisseau ::vitroul (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de rotation roulis

Références consignerotvitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.3.61 void ayoaron ::objets ::CVaisseau ::vittang (irr ::f32 cs) [virtual]

ajoute à la consigne de vitesse de rotation tangage

Références consignerotvitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.45.4 Documentation des données membres

7.45.4.1 CBoiteNoire* ayoaron ::objets ::CVaisseau ::blackbox

pointeur vers une boite noire

Référencé par CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et prePhysique().

7.45.4.2 irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CSolideRigide ::cam [protected, inherited]

pointeur vers le node de camera

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.45.4.3 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::cam1 [protected, inherited]

direction du regard de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.45.4.4 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camp [protected, inherited]

position de la caméra dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.45.4.5 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camup [protected, inherited]

vecteur de la verticale de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.45.4.6 bool ayoaron ::objets ::CVaisseau ::consigne [protected]

interrupteur de la commande de consigne

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), prePhysique(), et toggleconsigne().

7.45.4.7 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::consignerotvitesse [protected]

consigne de vitesse de rotation

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique(), prePhysique(), reset-

consigne(), vitlace(), vitroul(), et vittang().

7.45.4.8 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::consignevitesse
[protected]

consigne de vitesse de translation

Référencé par ayoaron ::objets ::CVaisseauX1 ::postPhysique(), prePhysique(), resetConsigne(), vitavar(), vitgadr(), et vithaba().

7.45.4.9 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::donn
[protected, inherited]

pointeur vers la structure de donnée

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getDonn(), ayoaron ::objets ::CMoteur ::getForce(), ayoaron ::physic ::CSolideRigide ::getMasse(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::objets ::CMoteur ::getPoussee(), ayoaron ::objets ::CMoteur ::getPousseeMax(), ayoaron ::objets ::CMoteur ::getRegime(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::objets ::CMoteur ::isReversible(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CAsteriode ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::objets ::CMoteur ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), ayoaron ::physic ::CSolideRigide ::setParent(), ayoaron ::objets ::CMoteur ::setRegime(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::setVector().

7.45.4.10 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::DTV

7.45.4.11 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::DTVR

7.45.4.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::forceResultante
[protected, inherited]

le vecteur de la résultante des efforts, il est exprimé dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getLocalForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.45.4.13 bool ayoaron ::physic ::CSolideRigide ::hascamera [protected,
inherited]

interrupteur de caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::phys-

sic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.45.4.14 std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::sic ::CSolideRigide ::l_enfant [protected, inherited]

liste des objets enfants

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getEnfant(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd(), ayoaron ::physic ::CSolideRigide ::getNbEnfant(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

7.45.4.15 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::momentResultant [protected, inherited]

le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.45.4.16 std ::map<irr ::u32,objets ::CMoteur*> ayoaron ::objets ::CVaisseau ::moteurs [protected]

liste des moteurs

Référencé par addpoussee(), defineRegineFM(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et resetpousse().

7.45.4.17 irr ::u32 ayoaron ::objets ::CVaisseau ::Nmoteur [protected]

nombre de moteurs

Référencé par addpoussee(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et resetpousse().

7.45.4.18 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::node [protected, inherited]

pointeur vers le maillage et le scene node

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::getRelativePosition(), ayoaron ::physic ::CSolideRigide ::getRelativeRota-

tion(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::physic ::CSolideRigide ::setParent(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), et ayoaron ::physic ::CSolideRigide ::setRelativeRotation().

7.45.4.19 CSolideRigide* ayoaron ::physic ::CSolideRigide ::parent
[protected, inherited]

pointeur vers l'objet parent

Réimplémentée dans [ayoaron ::objets ::CMoteur](#).

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

7.45.4.20 CModulePilotage* ayoaron ::objets ::CVaisseau ::pilote
[protected]

module de pilotage

Référencé par CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et prePhysique().

7.45.4.21 irr ::f32 ayoaron ::objets ::CVaisseau ::vitmax [protected]

vitesse max

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et vitavar().

7.45.4.22 physic ::CSecteur* ayoaron ::objets ::CVaisseau ::world
[protected]

sur déclaration de world en CSecteur (dérivant d'un CWorld)

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Référencé par CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

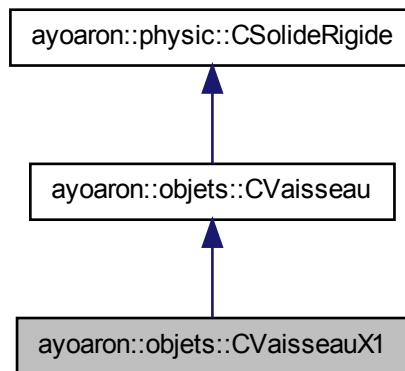
La documentation de cette classe a été générée à partir des fichiers suivants :

- [CVaisseau.h](#)
- [CVaisseau.cpp](#)

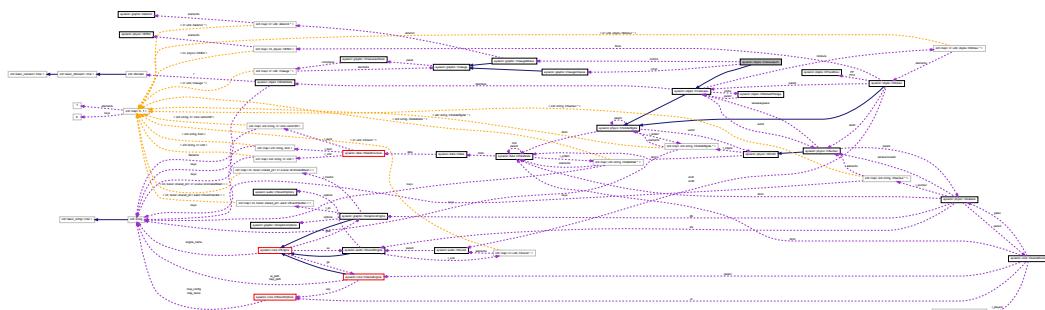
7.46 Référence de la classe ayoaron ::objets ::CVaisseauX1

```
#include <CVaisseauX1.h>
```

Graphe d'héritage de ayoaron ::objets ::CVaisseauX1 :



Graphe de collaboration de ayoaron ::objets ::CVaisseauX1 :



Fonctions membres publiques

- void **addAbsoluteEffort** (CEffort F)
ajoute un effort exprimé dans le repère global
- void **addEnfant** (std ::string nom, CSolideRigide *enfant)
ajoute un nouvel enfant (s'il n'existe pas déjà)
- void **addLocalEffort** (CEffort F)
ajoute un effort exprimé dans le repère local
- void **addpoussee** (irr : u32 moteurId, irr : f64 regime)
ajoute une composante de poussée
- void **addRelativeEffort** (CEffort F)
ajoute un effort exprimé dans le repère du parent
- **CVaisseauX1** (data : CDataNode *d, physic : CSecteur *p, physic : CSolideRigide *po=NULL, irr : s32 ID=-1)

- constructeur*
- virtual void **defineRegimeFM** (irr ::core ::vector3df f, irr ::core ::vector3df m)

defini les régimes moteur en fonction de la consigne de force et vitesse
 - void **dropEnfant** (std ::string nom)

supprime un enfant de la liste
 - void **frame** (double dt)

effectue les diverses opération de mise à jour
 - irr ::core ::vector3df **getAbsoluteForceResultante** ()

renvoie la force résultante exprimée dans le repère global
 - irr ::core ::vector3df **getAbsoluteMomentResultante** ()

renvoie le moment résultant exprimé dans le repère global
 - irr ::core ::vector3df **getAbsolutePosition** ()

renvoie la position de l'objet dans le monde
 - irr ::core ::vector3df **getAbsoluteRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
 - irr ::core ::matrix4 **getAbsoluteRotation** ()

renvoie la matrice de rotation entre le repère global et le repère local
 - irr ::core ::vector3df **getAbsoluteVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
 - **data** ::CDataNode * **getDonn** ()

acces aux données
 - CSolideRigide * **getEnfant** (std ::string nom)

renvoie l'enfant donné
 - irr ::core ::matrix4 **getGlobalInertie** ()

revoie la matrice d'inertie
 - std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantBegin** ()

renvoie un itérator sur les enfant pointant au début de la map
 - std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantEnd** ()

renvoie un itérator sur les enfant pointant à la fin de la map
 - irr ::core ::vector3df **getLocalForceResultante** ()

renvoie la force résultante exprimée dans le repère local
 - irr ::core ::matrix4 **getLocalInertie** ()

revoie la matrice d'inertie
 - irr ::core ::vector3df **getLocalMomentResultante** ()

renvoie le moment résultant exprimé dans le repère local
 - irr ::core ::vector3df **getLocalRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère local
 - irr ::core ::vector3df **getLocalVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère local
 - irr ::f64 **getMasse** ()

fonction qui renvoie la masse du solide
 - irr ::u32 **getNbEnfant** ()

renvoie le nombre d'enfants de cet objet
 - irr ::scene ::ISceneNode * **getNode** ()

renvoie le scene node et le maillage associé
 - irr ::core ::vector3df **getRelativeForceResultante** ()

renvoie la force résultante exprimée dans le repère du parent
 - irr ::core ::matrix4 **getRelativeInertie** ()

revoie la matrice d'inertie
 - irr ::core ::vector3df **getRelativeMomentResultante** ()

renvoie le moment résultant exprimé dans le repère du parent
 - irr ::core ::vector3df **getRelativePosition** ()

renvoie la position de l'objet par rapport à l'objet parent
 - irr ::core ::vector3df **getRelativeRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère du parent
 - irr ::core ::matrix4 **getRelativeRotation** ()

renvoie la matrice de rotation entre le repère relatif et le repère local

- irr ::core ::vector3df **getRelativeVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère du parent
- virtual void **loadObjetData** (data ::CDataNode *d)

charge l'objet depuis un fichier
- void **postPhysique** (double dt)

fonction de mise à jour
- void **prePhysique** (double dt)

fonction de mise à jour
- virtual void **resetConsigne** ()

reset les consignes
- void **resetEffort** ()

remet les efforts à zéro
- void **resetMouvement** (void)

reset moteur
- void **resetPousse** (void)

reset moteur
- void **setAbsoluteRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide (tient compte de la vitesse d'entraînement du repère parent)
- void **setAbsoluteVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
- void **setLocalRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide
- void **setLocalVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide
- void **setParent** (CSolideRigide *par)

change le parent de cet objet
- void **setRelativePosition** (irr ::core ::vector3df pos)

définit la position de l'objet par rapport à son parent
- void **setRelativeRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide
- void **setRelativeRotation** (irr ::core ::vector3df rot)

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.
- void **setRelativeVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide
- void **toggleConsigne** (void)

interrupteur de consigne
- virtual void **vitavar** (irr ::f32 cs)

ajoute à la consigne de vitesse de translation avant-arrière
- virtual void **vitgadr** (irr ::f32 cs)

ajoute à la consigne de vitesse de translation gauche droite
- virtual void **vithaba** (irr ::f32 cs)

ajoute à la consigne de vitesse de translation haut bas
- virtual void **vitlace** (irr ::f32 cs)

ajoute à la consigne de vitesse de rotation lacet
- virtual void **vitroul** (irr ::f32 cs)

ajoute à la consigne de vitesse de rotation roulis
- virtual void **vittang** (irr ::f32 cs)

ajoute à la consigne de vitesse de rotation tangage
- virtual ~CVaisseauX1 ()

Destructeur.

Attributs publics

- CBoiteNoire * **blackbox**

pointeur vers une boite noire
- irr ::core ::vector3df **DTV**

- irr : :core : :vector3df **DTVR**

Fonctions membres protégées

- irr : :core : :matrix4 **getMatrix3** (std : :string nom, REPÈRE ou, REPÈRE reperstockage)
renvoie le vecteur donné dans la base voulue
- irr : :core : :vector3df **getVector** (std : :string nom, REPÈRE ou, REPÈRE reperstockage)
renvoie le vecteur donné dans la base voulue
- void **setMatrix3** (std : :string nom, REPÈRE source, irr : :core : :matrix4 mat, REPÈRE reperstockage)
fonction permettant le changement de repère des matrices
- void **setVector** (std : :string nom, REPÈRE source, irr : :core : :vector3df vec, REPÈRE reperstockage)
fonction permettant le changement de repère des vecteurs

Attributs protégés

- irr : :scene : :ICameraSceneNode * **cam**
pointeur vers le node de camera
- irr : :core : :vector3df **caml**
direction du regard de la caméra
- irr : :core : :vector3df **camp**
position de la caméra dans le repère local
- irr : :core : :vector3df **camup**
vecteur de la verticale de la caméra
- **graphic** : :CGaugeMoteur * **cntmot**
- **graphic** : :CGaugeVitesse * **cntvit**
defini les régimes moteur en fonction de la consigne de force et vitesse
- bool **consigne**
interrupteur de la commande de consigne
- irr : :core : :vector3df **consignerotvitesse**
consigne de vitesse de rotation
- irr : :core : :vector3df **consignevitesse**
consigne de vitesse de translation
- **data** : :CDataNode * **donn**
pointeur vers la structure de donnée
- irr : :core : :vector3df **forceResultante**
le vecteur de la résultante des efforts, il est exprimé dans le repère local
- bool **hascamera**
interrupteur de caméra
- std : :map< std : :string, CSolideRigide * > **l_enfant**
liste des objets enfants
- irr : :core : :vector3df **momentResultant**
le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)
- std : :map< irr : :u32, objets : :CMoteur * > **moteurs**
liste des moteurs
- irr : :u32 **Nmoteur**
nombre de moteurs
- irr : :scene : :ISceneNode * **node**
pointeur vers le maillage et le scene node
- CSolideRigide * **parent**
pointeur vers l'objet parent
- **CModulePilotage** * **pilote**

- module de pilotage*
- `irr` : `:f32 vitmax`
vitesse max
 - `physic` : `:CSecteur * world`
sur déclaration de world en CSecteur (dérivant d'un CWorld)

7.46.1 Documentation des constructeurs et destructeur

7.46.1.1 `ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1 (data ::CDataNode * d, physic ::CSecteur * p, physic ::CSolideRigide * po = NULL, irr ::s32 ID = -1)`

constructeur

Références `ayoaron ::objets ::CVaisseau ::blackbox`, `ayoaron ::physic ::CSolideRigide ::parent`, `ayoaron ::objets ::CVaisseau ::pilote`, et `ayoaron ::objets ::CVaisseau ::world`.

7.46.1.2 `ayoaron ::objets ::CVaisseauX1 ::~CVaisseauX1 () [virtual]`

Destructeur.

7.46.2 Documentation des fonctions membres

7.46.2.1 `void ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort (CEffort F) [inherited]`

ajoute un effort exprimé dans le repère global

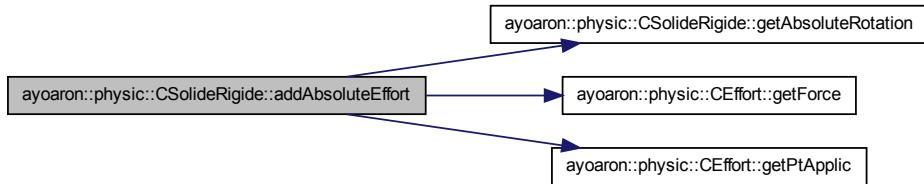
Paramètres

in	F	l'effort à ajouterexprimé dans le repère global, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	-----	--

Références `ayoaron ::physic ::CSolideRigide ::forceResultante`, `ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation()`, `ayoaron ::physic ::CEffort ::getForce()`, `ayoaron ::physic ::CEffort ::getPtApplic()`, et `ayoaron ::physic ::CSolideRigide ::momentResultant`.

Référencé par `ayoaron ::physic ::CSolideRigide ::frame()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



**7.46.2.2 void ayoaron ::physic ::CSolideRigide ::addEnfant (std ::string *nom*,
CSolideRigide * *enfant*) [inherited]**

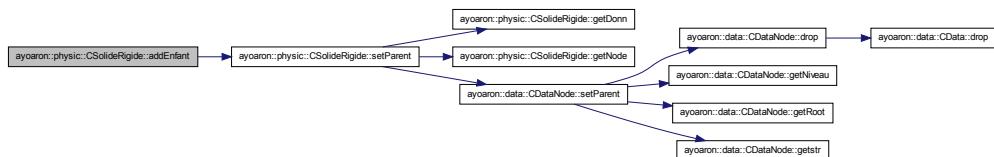
ajoute un nouvel enfant (s'il n'existe pas déjà)

Paramètres

<i>nom</i>	le nom (unique) de l'enfant
<i>enfant</i>	pointeur vers l'enfant en question

Références ayoaron ::physic ::CSolideRigide ::l_enfant, et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.46.2.3 void ayoaron ::physic ::CSolideRigide ::addLocalEffort (**CEffort F**)
 [inherited]

ajoute un effort exprimé dans le repère local

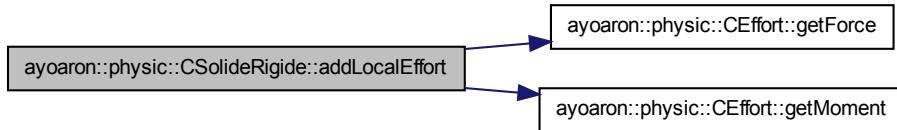
Paramètres

in	F	l'effort à ajouterexprimé dans le repère local, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	----------	---

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getMoment(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Référencé par ayoaron ::objets ::CMoteur ::prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.4 void ayoaron ::objets ::CVaisseau ::addpoussee (*irr : u32 moteurId, irr : f64 regime*) [inherited]

ajoute une composante de poussée

Références ayoaron ::objets ::CVaisseau ::moteurs, et ayoaron ::objets ::CVaisseau ::Nmoteur.

7.46.2.5 void ayoaron ::physic ::CSolideRigide ::addRelativeEffort (**CEffort F**)
 [inherited]

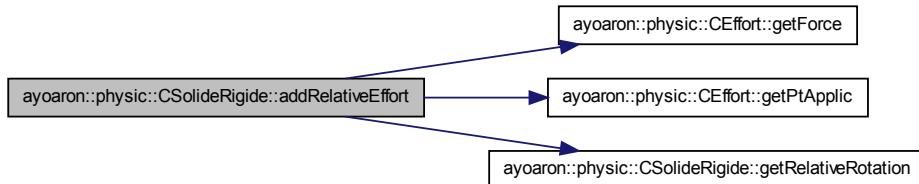
ajoute un effort exprimé dans le repère du parent

Paramètres

in	F	l'effort à ajouter exprimé dans le repère du parent, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	-----	--

Références ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



**7.46.2.6 void ayoaron ::objets ::CVaisseau ::defineRegimeFM (irr ::core ::vector3df f ,
irr ::core ::vector3df m) [virtual, inherited]**

defini les régimes moteur en fonction de la consigne de force et vitesse

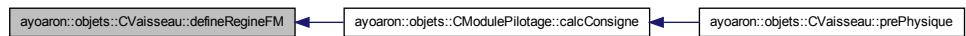
Références ayoaron ::objets ::CPousMouv ::h(), et ayoaron ::objets ::CVaisseau ::moteurs.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.7 void ayoaron ::physic ::CSolideRigide ::dropEnfant (std ::string *nom*)
 [inherited]

supprime un enfant de la liste

Paramètres

<i>nom</i>	nom de l'enfant à supprimer
------------	-----------------------------

7.46.2.8 void ayoaron ::physic ::CSolideRigide ::frame (double *dt*) [inherited]

effectue les diverses opération de mise à jour

Paramètres

in	<i>dt</i>	le pas de temps
----	-----------	-----------------

la frame :

tout d'abord, l'effort global

on pre frame tous les objets enfants

la translation, calculée dans le repère du parent

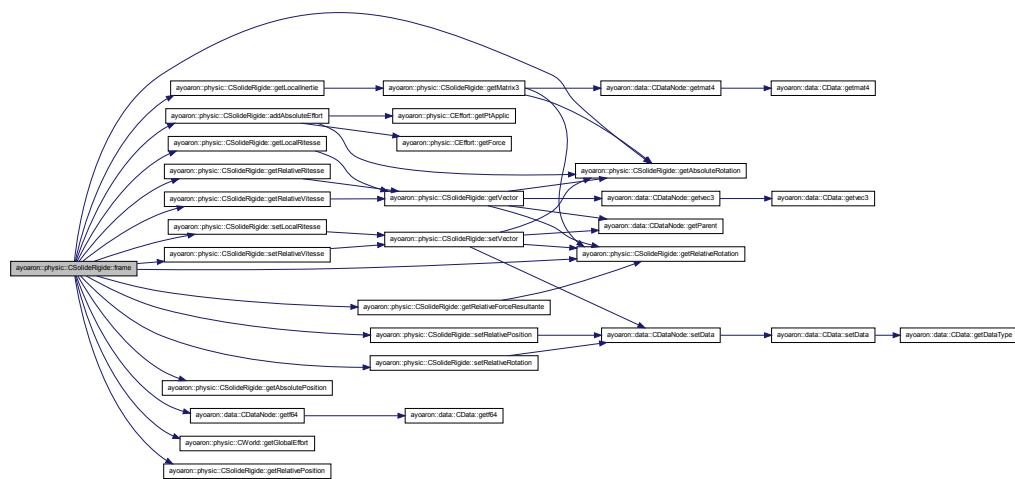
la rotation, calculée dans le repère parent

on post frame tous les objets enfants

Références ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::cam, ayoaron ::physic ::CSolideRigide ::caml, ayoaron ::physic ::CSolideRigide ::camp, ayoaron ::physic ::CSolideRigide ::camup, ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::forceResultante, ayoaron ::physic ::CSolideRigide ::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getf64(), ayoaron ::physic ::CWorld ::getGlobalEffort(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativePosition(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeVitesse(), ayoaron ::physic ::CSolideRigide ::hascamera, ayoaron ::physic ::CSolideRigide ::l_enfant, ayoaron ::physic ::CSolideRigide ::momentResultant, ayoaron ::physic ::CSolideRigide ::node, ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), ayoaron ::physic ::CSolideRigide ::setRe-

lativeVitesse(), et ayoaron ::physic ::CSolideRigide ::world.

Voici le graphe d'appel pour cette fonction :



7.46.2.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère global

Renvoie

la résultante des efforts exprimée dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation().

Voici le graphe d'appel pour cette fonction :



7.46.2.10 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante () [inherited]

renvoie le moment résultant exprimé dans le repère global

Renvoie

le moment résultant exprimé dans le référentiel global

Références ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.46.2.11 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsolutePosition ()`
 [inherited]

renvoie la position de l'objet dans le monde

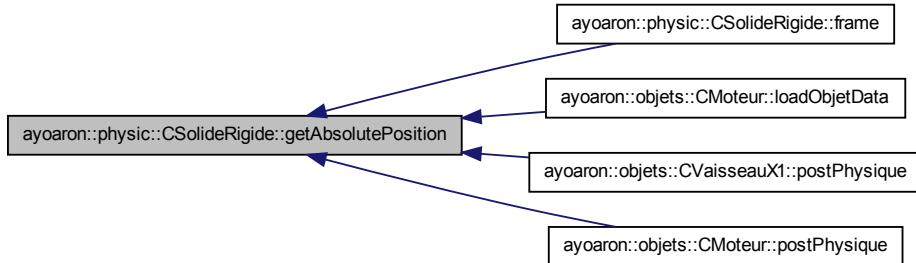
Renvoie

le vecteur position absolue de l'objet

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::objets ::CMoteur ::loadObjetData(), postPhysique(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.46.2.12 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse ()`
 [inherited]

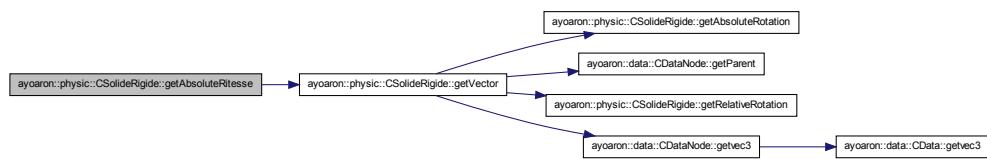
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse de rotation dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.46.2.13 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation () [inherited]

renvoie la matrice de rotation entre le repère global et le repère local

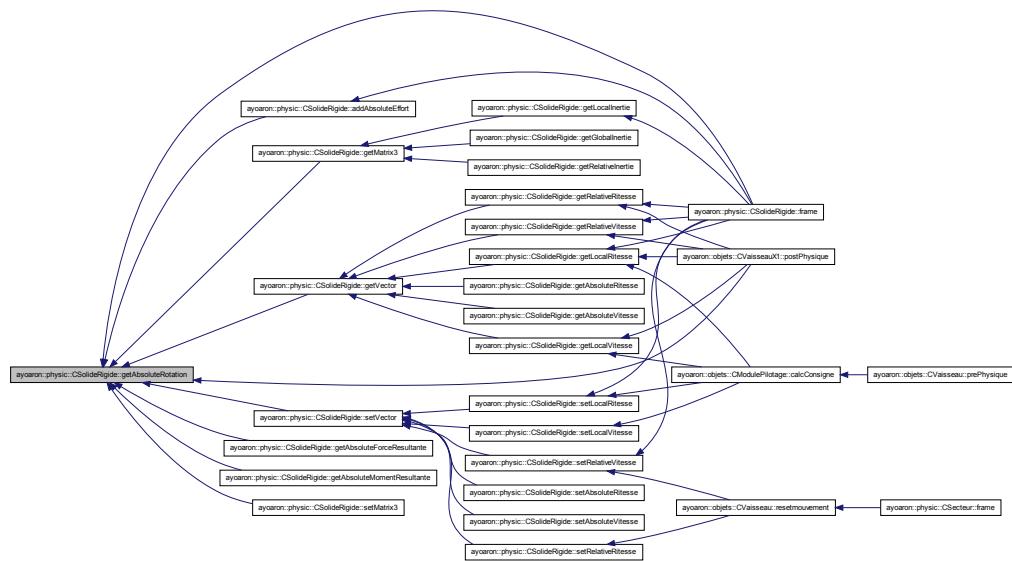
Renvoie

la matrice de rotation du repère global au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getVector(), postPhysique(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.46.2.14 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse () [inherited]

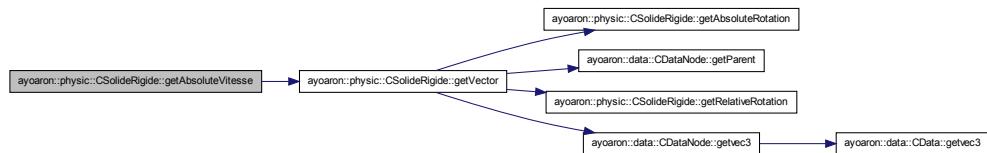
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.46.2.15 **data** : **CDataNode*** ayoaron ::physic ::CSolideRigide ::getDonn ()
 [inline, inherited]

acces aux données

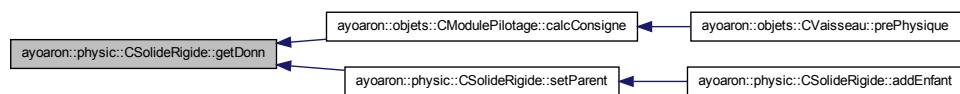
Renvoie

pointeur vers la structure de donnée

Références ayoaron ::physic ::CSolideRigide ::donn.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.46.2.16 **CSolideRigide *** ayoaron ::physic ::CSolideRigide ::getEnfant (**std ::string nom**)
 [inherited]

renvoie l'enfant donné

Paramètres

<i>nom</i>	le nom de l'enfant voulu
------------	--------------------------

Renvoie

pointeur vers l'enfant désiré, renvoie NULL si l'enfant n'existe pas

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.46.2.17 **irr ::core ::matrix4** ayoaron ::physic ::CSolideRigide ::getGlobalInertie ()
 [inline, inherited]

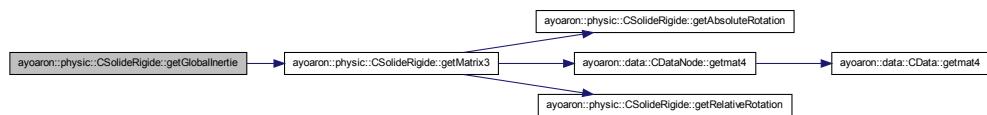
revoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère global

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LOCAL.

Voici le graphe d'appel pour cette fonction :



7.46.2.18 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin() [inherited]`

renvoie un itérator sur les enfant pointant au début de la map

Renvoie

iterator pointant au début de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.46.2.19 `std ::map< std ::string, CSolideRigide * > ::iterator ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd() [inherited]`

renvoie un itérator sur les enfant pointant à la fin de la map

Renvoie

iterator pointant à la fin de la liste des enfant

Références ayoaron ::physic ::CSolideRigide ::l_enfant.

7.46.2.20 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalForceResultante() [inherited]`

renvoie la force résultante exprimée dans le repère local

Renvoie

la résultante des efforts exprimée dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::forceResultante.

7.46.2.21 `irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getLocallnertie() [inline, inherited]`

revoie la matrice d'inertie

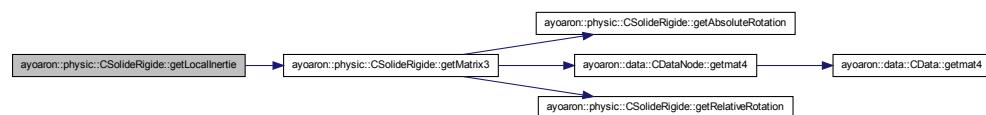
Renvoie

la matrice d'inertie dans le repère local

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), et ayoaron ::physic ::LO-CAL.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.22 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante ()` [inherited]

renvoie le moment résultant exprimé dans le repère local

Renvoie

le moment résultant exprimé dans le référentiel local

Références ayoaron ::physic ::CSolideRigide ::momentResultant.

7.46.2.23 `irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalRitesse ()` [inherited]

renvoie l'expression de la vitesse de rotation dans le repère local

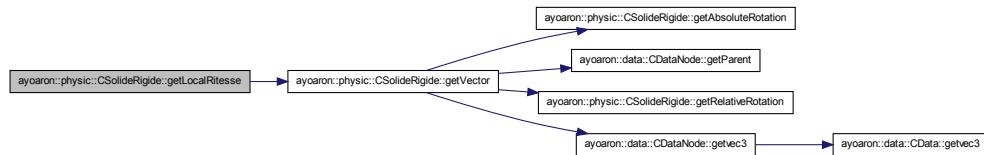
Renvoie

le vecteur vitesse de rotation dans le repère local

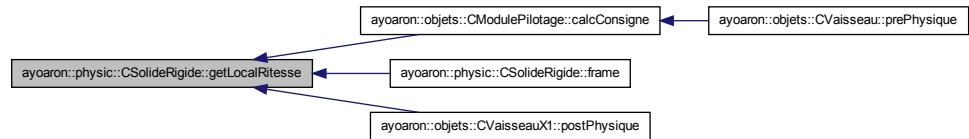
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::frame(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.24 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalVitesse () [inherited]

renvoie l'expression de la vitesse de translation dans le repère local

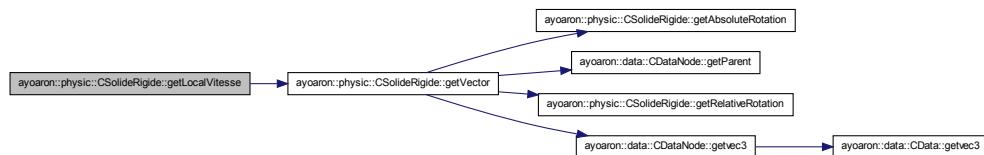
Renvoie

le vecteur vitesse dans le repère local

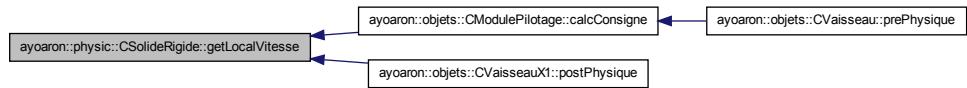
Références ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.25 irr : :f64 ayoaron ::physic ::CSolideRigide ::getMasse () [inline, inherited]

fonction qui renvoie la masse du solide

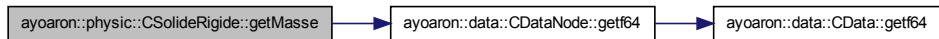
Renvoie

la masse du solide

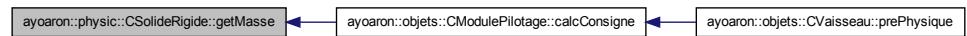
Références ayoaron ::physic ::CSolideRigide ::donn, et ayoaron ::data ::CData-Node ::getf64().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.26 irr : :core ::matrix4 ayoaron ::physic ::CSolideRigide ::getMatrix3 (std ::string nom, REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

nom	nom de la clé dans la base de données
ou	le repère dans lequel on veut le vecteur

<i>reperesto- ckage</i>	le repère dans lequel est stocker le vecteur
-----------------------------	--

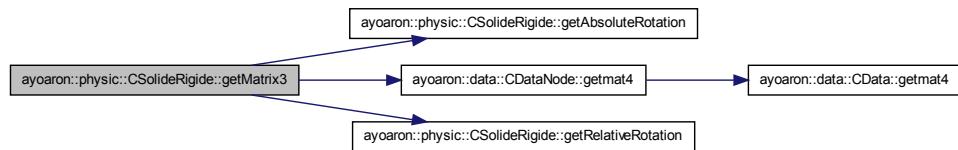
Renvoie

le vecteur voulu dans la base voulue

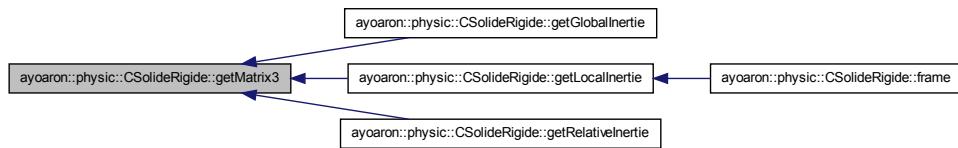
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CData-Node ::getmat4(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getGlobalInertie(), ayoaron ::physic ::CSolideRigide ::getLocalInertie(), et ayoaron ::physic ::CSolideRigide ::getRelativeInertie().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.27 `irr : :u32 ayoaron ::physic ::CSolideRigide ::getNbEnfant() [inline, inherited]`

renvoie le nombre d'enfants de cet objet

Renvoie

le nombre d'enfant

Références ayoaron ::physic ::CSolideRigide :::l_enfant.

7.46.2.28 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::getNode ()
[inline, inherited]

renvoie le scene node et le maillage associé

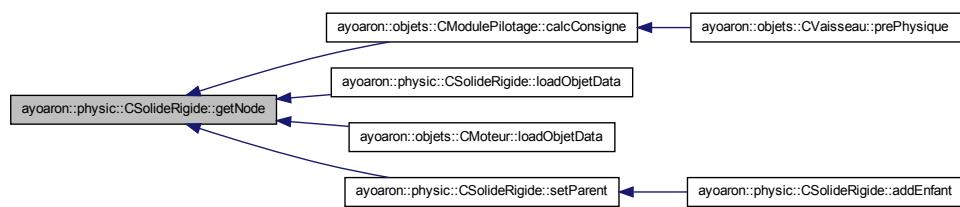
Renvoie

pointeur vers le IAnimatedMeshSceneNode

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), et ayoaron ::physic ::CSolideRigide ::setParent().

Voici le graphe d'appel pour cette fonction :



7.46.2.29 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante () [inherited]

renvoie la force résultante exprimée dans le repère du parent

Renvoie

la résultante des efforts exprimée dans le référentiel du parent

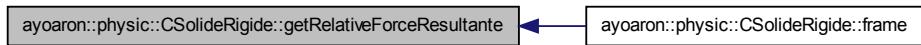
Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::getRelativeRotation().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.30 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativelnertie ()
 [inline, inherited]

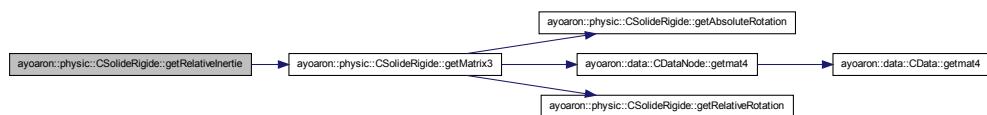
revoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère du parent

Références ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::LO-CAL, et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.46.2.31 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante () [inherited]

revoie le moment résultant exprimé dans le repère du parent

Renvoie

le moment résultant exprimé dans le référentiel du parent

Références ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::momentResultant.

Voici le graphe d'appel pour cette fonction :



7.46.2.32 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativePosition ()
[inherited]

renvoie la position de l'objet par rapport à l'objet parent

Renvoie

le vecteur position relative au parent

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.33 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeRitesse ()
[inherited]

renvoie l'expression de la vitesse de rotation dans le repère du parent

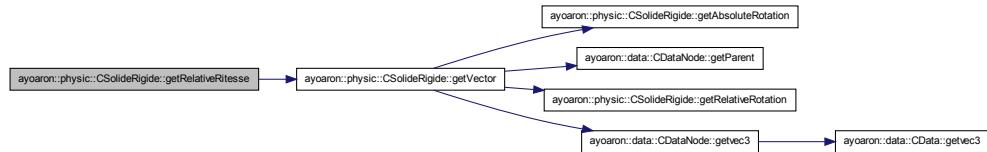
Renvoie

le vecteur vitesse de rotation dans le repère du parent

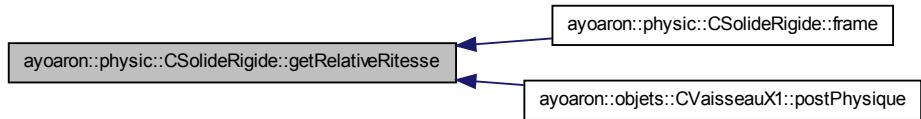
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.34 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeRotation ()
 [inherited]

renvoie la matrice de rotation entre le repère relatif et le repère local

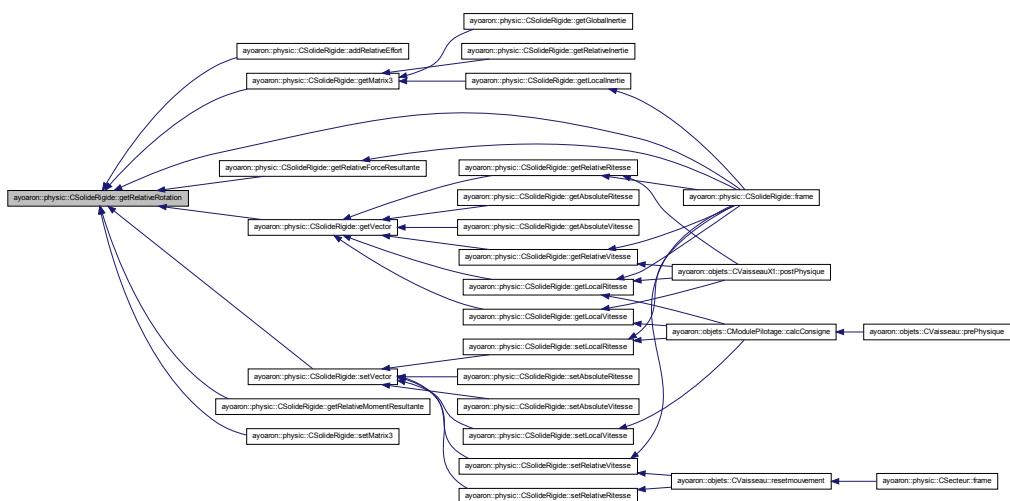
Renvoie

la matrice de rotation du repère relatif au repère local

Références ayoaron ::physic ::CSolideRigide ::node.

Référencé par ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getMatrix3(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), ayoaron ::physic ::CSolideRigide ::getVector(), ayoaron ::physic ::CSolideRigide ::setMatrix3(), et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.46.2.35 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeVitesse ()
[inherited]

renvoie l'expression de la vitesse de translation dans le repère du parent

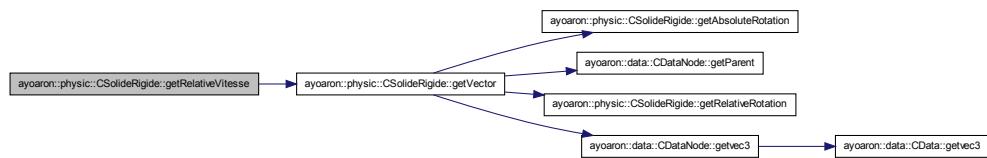
Renvoie

le vecteur vitesse dans le repère du parent

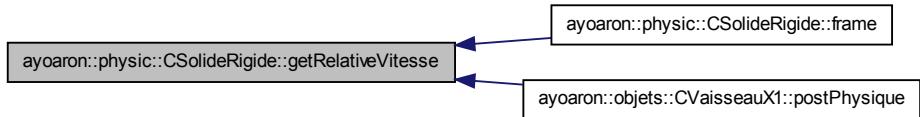
Références ayoaron ::physic ::CSolideRigide ::getVector(), et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.36 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getVector (std ::string nom,
REPERE ou, REPERE reperestockage) [protected, inherited]

renvoie le vecteur donné dans la base voulue

Paramètres

nom	nom de la clé dans la base de données
ou	le repère dans lequel on veut le vecteur
reperestockage	le repère dans lequel est stocker le vecteur

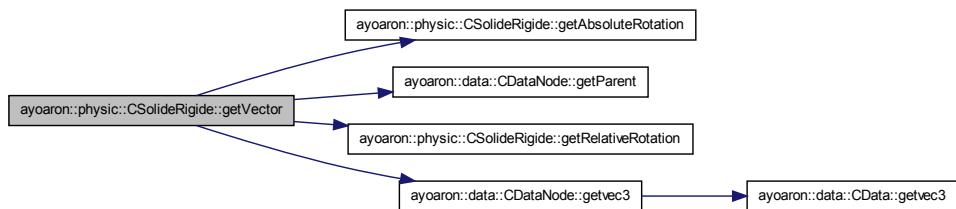
Renvoie

le vecteur voulu dans la base voulue

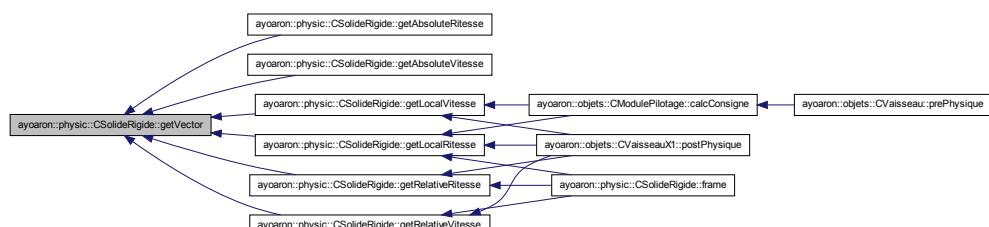
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::getLocalRitesse(), ayoaron ::physic ::CSolideRigide ::getLocalVitesse(), ayoaron ::physic ::CSolideRigide ::getRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::getRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.37 void ayoaron ::objets ::CVaisseauX1 ::loadObjetData (data ::CDataNode * d) [virtual]

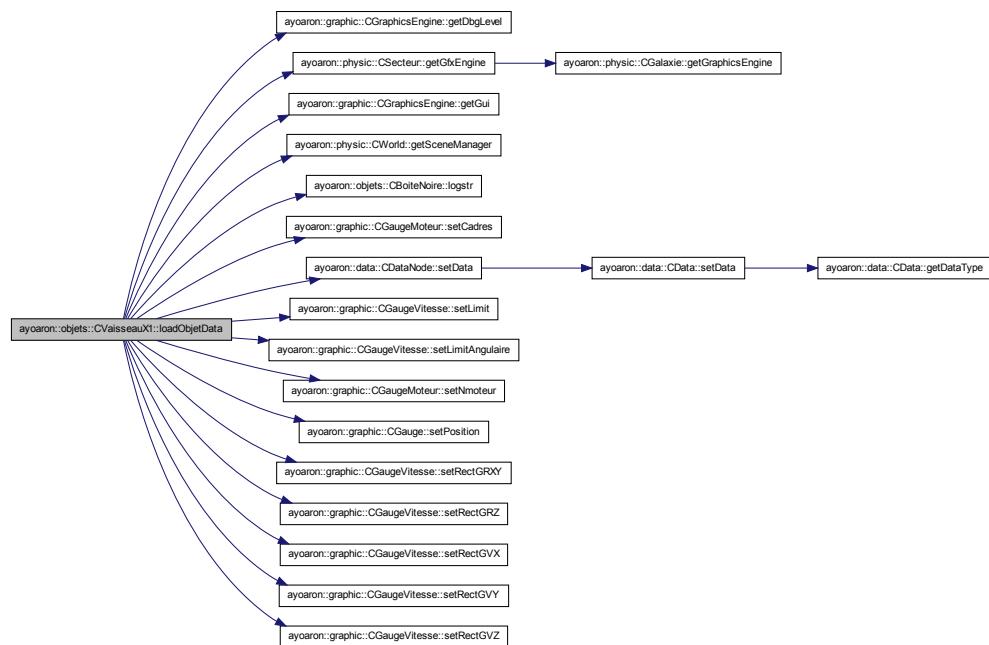
charge l'objet depuis un fichier

Réimplémentée à partir de [ayoaron ::objets ::CVaisseau](#).

Références ayoaron ::objets ::CVaisseau ::blackbox, ayoaron ::physic ::CSolideRi-

gide :: cam, ayoaron :: physic :: CSolideRigide :: caml, ayoaron :: physic :: CSolideRigide :: camp, ayoaron :: physic :: CSolideRigide :: camup, cntmot, cntvit, ayoaron :: objets :: CVaisseau :: consigne, ayoaron :: physic :: CSolideRigide :: donn, ayoaron :: graphic :: CGraphicsEngine :: getDbgLevel(), ayoaron :: physic :: CSecteur :: getGfxEngine(), ayoaron :: graphic :: CGraphicsEngine :: getGui(), ayoaron :: physic :: CWorld :: getSceneManager(), ayoaron :: physic :: CSolideRigide :: hascamera, ayoaron :: physic :: CSolideRigide :: l_enfant, ayoaron :: objets :: CBoiteNoire :: logstr(), ayoaron :: objets :: CVaisseau :: moteurs, ayoaron :: objets :: CVaisseau :: Nmoteur, ayoaron :: physic :: CSolideRigide :: node, ayoaron :: objets :: CVaisseau :: pilote, ayoaron :: graphic :: CGaugeMoteur :: setCadres(), ayoaron :: data :: CDataNode :: setData(), ayoaron :: graphic :: CGaugeVitesse :: setLimit(), ayoaron :: graphic :: CGaugeVitesse :: setLimitAngulaire(), ayoaron :: graphic :: CGaugeMoteur :: setNmoteur(), ayoaron :: graphic :: CGauge :: setPosition(), ayoaron :: graphic :: CGaugeVitesse :: setRectGRXY(), ayoaron :: graphic :: CGaugeVitesse :: setRectGRZ(), ayoaron :: graphic :: CGaugeVitesse :: setRectGVX(), ayoaron :: graphic :: CGaugeVitesse :: setRectGVY(), ayoaron :: graphic :: CGaugeVitesse :: setRectGVZ(), ayoaron :: objets :: CVaisseau :: vitmax, et ayoaron :: objets :: CVaisseau :: world.

Voici le graphe d'appel pour cette fonction :



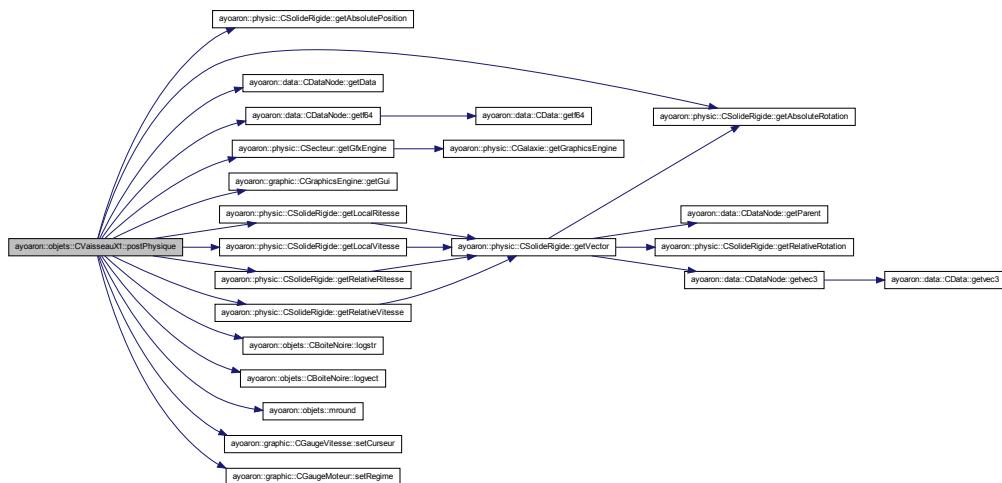
7.46.2.38 void ayoaron ::objets ::CVaisseauX1 ::postPhysique (double dt) [virtual]

fonction de mise à jour

Réimplémentée à partir de [ayoaron ::physic ::CSolideRigide](#).

Références ayoaron ::objets ::CVaisseau ::blackbox, cntmot, cntvit, ayoaron ::objets ::CVaisseau ::consignerotvitesse, ayoaron ::objets ::CVaisseau ::consignevitesse, ayoaron ::physic ::CSolideRigide :::donn, ayoaron ::physic ::CSolideRigide :::forceResultante, ayoaron ::physic ::CSolideRigide :::getAbsolutePosition(), ayoaron ::physic ::CSolideRigide :::getAbsoluteRotation(), ayoaron ::data ::CDataNode :::getData(), ayoaron ::data ::CDataNode :::getf64(), ayoaron ::physic ::CSecteur :::getGfxEngine(), ayoaron ::graphic ::CGraphicsEngine :::getGui(), ayoaron ::physic ::CSolideRigide :::getLocalRitesse(), ayoaron ::physic ::CSolideRigide :::getLocalVitesse(), ayoaron ::physic ::CSolideRigide :::getRelativeRitesse(), ayoaron ::physic ::CSolideRigide :::getRelativeVitesse(), ayoaron ::objets ::CBoiteNoire :::logstr(), ayoaron ::objets ::CBoiteNoire :::logvect(), ayoaron ::physic ::CSolideRigide :::momentResultant, ayoaron ::objets ::CVaisseau ::moteurs, ayoaron ::objets ::mrround(), ayoaron ::physic ::CSolideRigide :::node, ayoaron ::graphic ::CGaugeVitesse :::setCurseur(), ayoaron ::graphic ::CGaugeMoteur :::setRegime(), et ayoaron ::objets ::CVaisseau ::world.

Voici le graphe d'appel pour cette fonction :



7.46.2.39 void ayoaron ::objets ::CVaisseau ::prePhysique (double dt) [virtual, inherited]

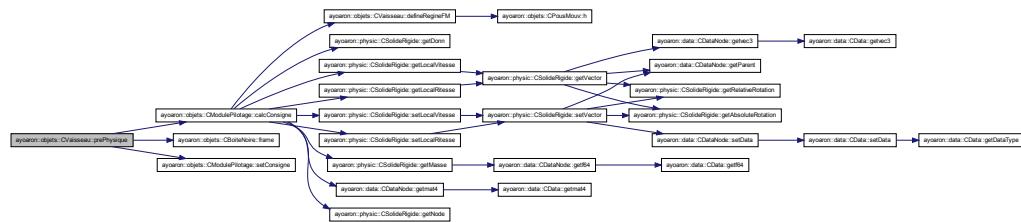
fonction de mise à jour

calcul de la consigne moteur

Réimplémentée à partir de ayoaron ::physic ::CSolideRigide.

Références ayoaron ::objets ::CVaisseau ::blackbox, ayoaron ::objets ::CModulePilotage ::calcConsigne(), ayoaron ::objets ::CVaisseau ::consigne, ayoaron ::objets ::CVaisseau ::consignerotvitesse, ayoaron ::objets ::CVaisseau ::consignevitesse, ayoaron ::objets ::CBoiteNoire :::frame(), ayoaron ::objets ::CVaisseau ::pilote, et ayoaron ::objets ::CModulePilotage ::setConsigne().

Voici le graphe d'appel pour cette fonction :



7.46.2.40 void ayoaron ::objets ::CVaisseau ::resetconsigne () [virtual, inherited]

reset les consignes

Références ayoaron ::objets ::CVaisseau ::consignerotvitesse, et ayoaron ::objets ::CVaisseau ::consignevitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.41 void ayoaron ::physic ::CSolideRigide ::resetEffort (void) [inherited]

remet les efforts à zéro

Références ayoaron ::physic ::CSolideRigide ::forceResultante, et ayoaron ::physic ::CSolideRigide ::momentResultant.

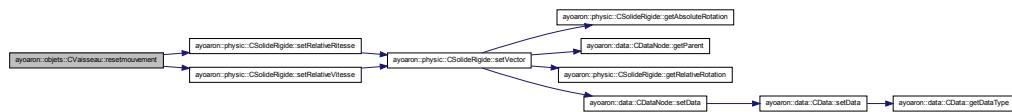
7.46.2.42 void ayoaron ::objets ::CVaisseau ::resetmouvement (void) [inline, inherited]

reset moteur

Références ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.43 void ayoaron ::objets ::CVaisseau ::resetpousse (void) [inherited]

reset moteur

Références ayoaron ::objets ::CVaisseau ::moteurs, et ayoaron ::objets ::CVaisseau ::Nmoteur.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.44 void ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse (irr ::core ::vector3df vit) [inherited]

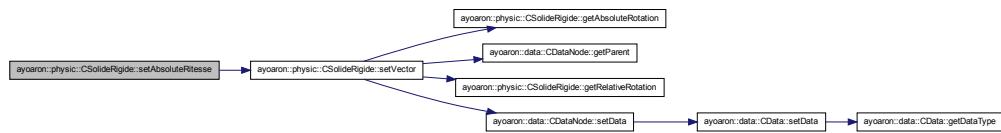
définit la vitesse de rotation du solide (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.46.2.45 void ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse (irr ::core ::vector3df vit) [inherited]

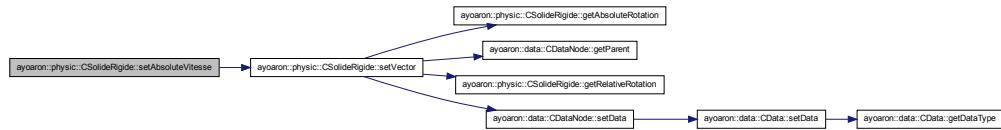
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Voici le graphe d'appel pour cette fonction :



7.46.2.46 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

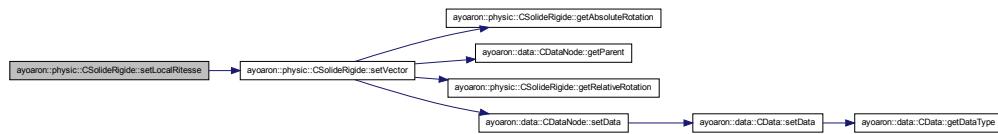
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère local
----	-----	---

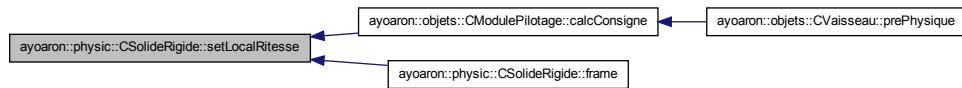
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.47 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

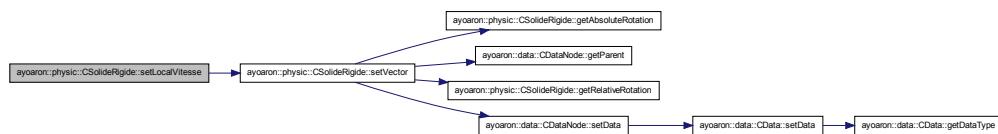
Paramètres

in	vit	le vecteur vitesse exprimé dans le repère local
----	-----	---

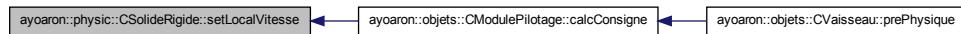
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.48 void ayoaron ::physic ::CSolideRigide ::setMatrix3 (std ::string *nom*, REPERE *source*, irr ::core ::matrix4 *mat*, REPERE *reperstockage*) [protected, inherited]

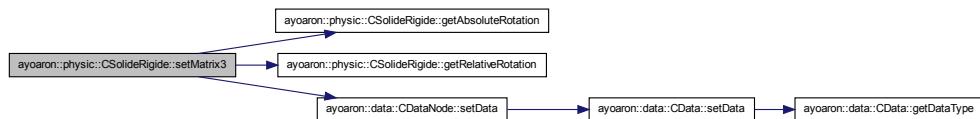
fonction permettant le changement de repère des matrices

Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>mat</i>	la matrice à stocker
<i>reperstockage</i>	le repère de stockage du vecteur

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.46.2.49 void ayoaron ::physic ::CSolideRigide ::setParent (CSolideRigide * *par*) [inherited]

change le parent de cet objet

Paramètres

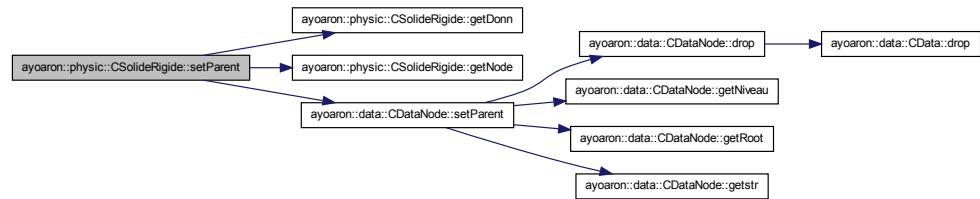
<i>in</i>	<i>par</i>	le nouveau parent
-----------	------------	-------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::getDonn(), ayoaron ::physic ::CSolideRigide ::getNode(), ayoaron ::physic ::CSolideRigide ::node, ayoaron ::physic ::CSolideRigide ::parent, et ayoaron ::data ::CData

taNode ::setParent().

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.50 void ayoaron ::physic ::CSolideRigide ::setRelativePosition (irr ::core ::vector3df pos) [inherited]

définit la position de l'objet par rapport à son parent

Paramètres

<i>pos</i>	la position relative de l'objet
------------	---------------------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.51 void ayoaron ::physic ::CSolideRigide ::setRelativeRitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse de rotation du solide

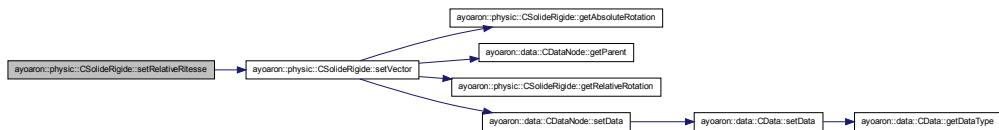
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère du parent
----	-----	---

Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.52 void ayoaron ::physic ::CSolideRigide ::setRelativeRotation (irr ::core ::vector3df rot) [inherited]

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.

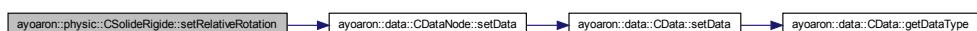
Paramètres

<i>rot</i>	le vecteur rotation
------------	---------------------

Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::CSolideRigide ::node, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.53 void ayoaron ::physic ::CSolideRigide ::setRelativeVitesse (irr ::core ::vector3df vit) [inherited]

définit la vitesse du solide

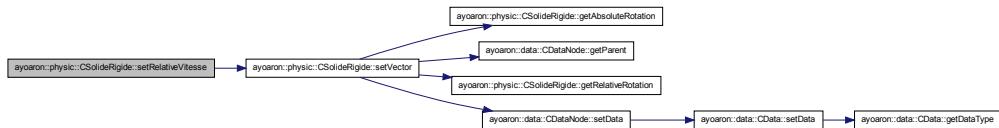
Paramètres

<i>in</i>	<i>vit</i>	le vecteur vitesse exprimé dans le repère du parent
-----------	------------	---

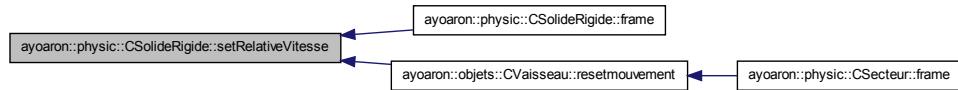
Références ayoaron ::physic ::PARENT, et ayoaron ::physic ::CSolideRigide ::setVector().

Référencé par ayoaron ::physic ::CSolideRigide ::frame(), et ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.54 void ayoaron ::physic ::CSolideRigide ::setVector (std ::string *nom*, REPERE *source*, irr ::core ::vector3df *vec*, REPERE *reperestockage*) [protected, inherited]

fonction permettant le changement de repère des vecteurs

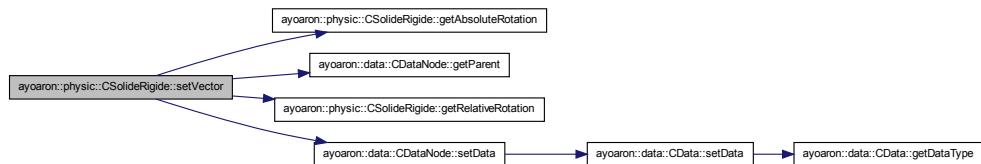
Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>vec</i>	le vecteur à stocker
<i>reperestockage</i>	le repère de stockage du vecteur

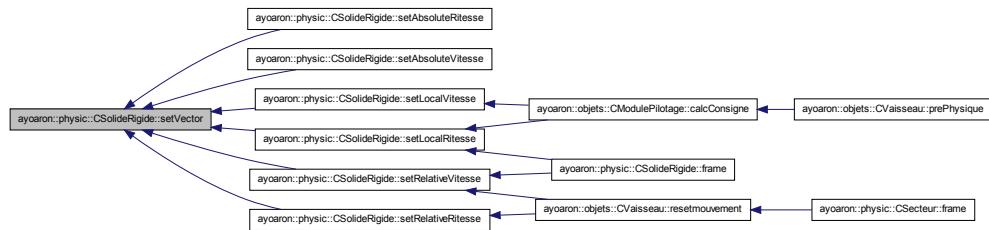
Références ayoaron ::physic ::CSolideRigide ::donn, ayoaron ::physic ::GENERAL, ayoaron ::physic ::CSolideRigide ::getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::physic ::CSolideRigide ::getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Référencé par ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse(), ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse(), ayoaron ::physic ::CSolideRigide ::setLocalRitesse(), ayoaron ::physic ::CSolideRigide ::setLocalVitesse(), ayoaron ::physic ::CSolideRigide ::setRelativeRitesse(), et ayoaron ::physic ::CSolideRigide ::setRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.46.2.55 void ayoaron ::objets ::CVaisseau ::toggleconsigne (void) [inline, inherited]

interrupteur de consigne

Références ayoaron ::objets ::CVaisseau ::consigne.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.56 void ayoaron ::objets ::CVaisseau ::vitavar (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de translation avant-arrière

Références ayoaron ::objets ::CVaisseau ::consigneVitesse, et ayoaron ::objets ::CVaisseau ::vitmax.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.57 void ayoaron ::objets ::CVaisseau ::vitgadr (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de translation gauche droite

Références ayoaron ::objets ::CVaisseau ::consignevitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



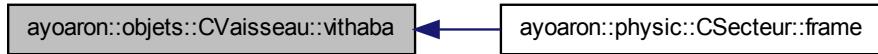
7.46.2.58 void ayoaron ::objets ::CVaisseau ::vithaba (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de translation haut bas

Références ayoaron ::objets ::CVaisseau ::consignevitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



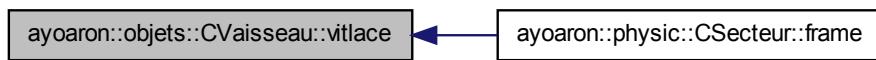
7.46.2.59 void ayoaron ::objets ::CVaisseau ::vitlace (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de rotation lacet

Références ayoaron ::objets ::CVaisseau ::consignerotvitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.60 void ayoaron ::objets ::CVaisseau ::vitroul (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de rotation roulis

Références ayoaron ::objets ::CVaisseau ::consignerotvitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.2.61 void ayoaron ::objets ::CVaisseau ::vittang (irr ::f32 cs) [virtual, inherited]

ajoute à la consigne de vitesse de rotation tangage

Références ayoaron ::objets ::CVaisseau ::consignerotvitesse.

Référencé par ayoaron ::physic ::CSecteur ::frame().

Voici le graphe d'appel pour cette fonction :



7.46.3 Documentation des données membres

7.46.3.1 CBoiteNoire* ayoaron ::objets ::CVaisseau ::blackbox [inherited]

pointeur vers une boite noire

Référencé par ayoaron ::objets ::CVaisseau ::CVaisseau(), CVaisseauX1(), loadObjetData(), postPhysique(), et ayoaron ::objets ::CVaisseau ::prePhysique().

7.46.3.2 irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CSolideRigide ::cam [protected, inherited]

pointeur vers le node de camera

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et loadObjetData().

7.46.3.3 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::caml [protected, inherited]

direction du regard de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et loadObjetData().

7.46.3.4 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camp [protected, inherited]

position de la caméra dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et loadObjetData().

7.46.3.5 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camup [protected, inherited]

vecteur de la verticale de la caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et loadObjetData().

7.46.3.6 **graphic** : **:CGaugeMoteur*** **ayoaron** : **:objets** : **:CVaisseauX1** : **:cntmot**
[protected]

Référencé par loadObjetData(), et postPhysique().

7.46.3.7 **graphic** : :CGaugeVitesse* **ayoaron** : :objets : :CVaisseauX1 : :cntvit
[protected]

defini les régimes moteur en fonction de la consigne de force et vitesse

Référencé par loadObjetData(), et postPhysique().

7.46.3.8 **bool ayoaron** : **:objets** : **:CVaisseau** : **:consigne** [protected, inherited]

interrupteur de la commande de consigne

Référencé par loadObjetData(), ayoaron ::objets ::CVaisseau ::prePhysique(), et ayoaron ::objets ::CVaisseau ::toggleconsigne().

7.46.3.9 **irr** : **:core** : **:vector3df** **ayoaron** : **:objets** : **:CVaisseau** : **:consignerotvitesse**
[protected, inherited]

consigne de vitesse de rotation

Référencé par postPhysique(), ayoaron ::objets ::CVaisseau ::prePhysique(), ayoaron ::objets ::CVaisseau ::resetconsigne(), ayoaron ::objets ::CVaisseau ::vitlace(), ayoaron ::objets ::CVaisseau ::vitroul(), et ayoaron ::objets ::CVaisseau ::vittang().

7.46.3.10 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::consignevesse
[protected, inherited]

consigne de vitesse de translation

Référencé par postPhysique(), ayoaron ::objets ::CVaisseau ::prePhysique(), ayoaron ::objets ::CVaisseau ::resetconsigne(), ayoaron ::objets ::CVaisseau ::vitavar(), ayoaron ::objets ::CVaisseau ::vitgadr(), et ayoaron ::objets ::CVaisseau ::vithaba().

7.46.3.11 **data** : :CDataNode* **ayoaron** : :physic : :CSolideRigide : :donn
[protected, inherited]

pointeur vers la structure de donnée

sic ::CSolideRigide ::setRelativePosition(), ayoaron ::physic ::CSolideRigide ::setRelativeRotation(), et ayoaron ::physic ::CSolideRigide ::setVector().

7.46.3.12 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::DTV [inherited]

7.46.3.13 irr ::core ::vector3df ayoaron ::objets ::CVaisseau ::DTVR
[inherited]

7.46.3.14 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::forceResultante
[protected, inherited]

le vecteur de la résultante des efforts, il est exprimé dans le repère local

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante(), ayoaron ::physic ::CSolideRigide ::getLocalForceResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante(), postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoaron ::physic ::CSolideRigide ::resetEffort().

7.46.3.15 bool ayoaron ::physic ::CSolideRigide ::hascamera [protected,
inherited]

interrupteur de caméra

Référencé par ayoaron ::physic ::CSolideRigide ::CSolideRigide(), ayoaron ::physic ::CSolideRigide ::frame(), et loadObjetData().

7.46.3.16 std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::CSolideRigide ::l_enfant [protected,
inherited]

liste des objets enfants

Référencé par ayoaron ::physic ::CSolideRigide ::addEnfant(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getEnfant(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin(), ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd(), ayoaron ::physic ::CSolideRigide ::getNbEnfant(), loadObjetData(), et ayoaron ::physic ::CSolideRigide ::loadObjetData().

7.46.3.17 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::momentResultant
[protected, inherited]

le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), ayoaron ::physic ::CSolideRigide ::frame(), ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante(), ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante(), ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante(), postPhysique(), ayoaron ::physic ::CSolideRigide ::prePhysique(), et ayoar-

ron :: physic :: CSolideRigide :: resetEffort().

7.46.3.18 std :: map<irr :: u32, objets :: CMoteur*> ayoaron :: objets :: CVaisseau :: moteurs [protected, inherited]

liste des moteurs

Référencé par ayoaron :: objets :: CVaisseau :: addpoussee(), ayoaron :: objets :: CVaisseau :: defineRegineFM(), loadObjetData(), postPhysique(), et ayoaron :: objets :: CVaisseau :: resetpousse().

7.46.3.19 irr :: u32 ayoaron :: objets :: CVaisseau :: Nmoteur [protected, inherited]

nombre de moteurs

Référencé par ayoaron :: objets :: CVaisseau :: addpoussee(), loadObjetData(), et ayoaron :: objets :: CVaisseau :: resetpousse().

7.46.3.20 irr :: scene :: !SceneNode* ayoaron :: physic :: CSolideRigide :: node [protected, inherited]

pointeur vers le maillage et le scene node

Référencé par ayoaron :: physic :: CSolideRigide :: frame(), ayoaron :: physic :: CSolideRigide :: getAbsolutePosition(), ayoaron :: physic :: CSolideRigide :: getAbsoluteRotation(), ayoaron :: physic :: CSolideRigide :: getNode(), ayoaron :: physic :: CSolideRigide :: getRelativePosition(), ayoaron :: physic :: CSolideRigide :: getRelativeRotation(), loadObjetData(), ayoaron :: physic :: CSolideRigide :: loadObjetData(), postPhysique(), ayoaron :: physic :: CSolideRigide :: setParent(), ayoaron :: physic :: CSolideRigide :: setRelativePosition(), et ayoaron :: physic :: CSolideRigide :: setRelativeRotation().

7.46.3.21 CSolideRigide* ayoaron :: physic :: CSolideRigide :: parent [protected, inherited]

pointeur vers l'objet parent

Réimplémentée dans ayoaron :: objets :: CMoteur.

Référencé par ayoaron :: physic :: CSolideRigide :: CSolideRigide(), ayoaron :: objets :: CVaisseau :: CVaisseau(), CVaisseauX1(), ayoaron :: physic :: CSolideRigide :: loadObjetData(), et ayoaron :: physic :: CSolideRigide :: setParent().

7.46.3.22 CModulePilotage* ayoaron :: objets :: CVaisseau :: pilote [protected, inherited]

module de pilotage

Référencé par ayoaron :: objets :: CVaisseau :: CVaisseau(), CVaisseauX1(), loadObjetData(), et ayoaron :: objets :: CVaisseau :: prePhysique().

7.46.3.23 **irr** : :f32 ayoaron ::objets ::CVaisseau ::vitmax [protected, inherited]

vitesse max

Référencé par loadObjetData(), et ayoaron ::objets ::CVaisseau ::vitavar().

7.46.3.24 **physic** : :CSecteur* ayoaron ::objets ::CVaisseau ::world [protected, inherited]

sur déclaration de world en CSecteur (dérivant d'un CWorld)

Réimplémentée à partir de ayoaron ::physic ::CSolideRigide.

Référencé par ayoaron ::objets ::CVaisseau ::CVaisseau(), CVaisseauX1(), loadObjetData(), et postPhysique().

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

- [CVaisseauX1.h](#)
- [CVaisseauX1.cpp](#)

7.47 Référence de la classe ayoaron ::physic ::CEffort

classe permettant de définir un effort appliqu     un objet

```
#include <CEffort.h>
```

Fonctions membres publiques

- **CEffort** (irr ::core ::vector3df posi=irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df forc=irr ::core ::vector3df(0, 0, 0))
Constructeur.
- irr ::core ::vector3df **getDirection** ()
renvoie le vecteur direction de la force
- irr ::core ::vector3df **getForce** ()
renvoie le vecteur force
- irr ::core ::vector3df **getMoment** ()
renvoie le moment   l'origine du rep  re qui a servit pour la d  finition
- irr ::f64 **getNorme** ()
renvoie la norme de la force
- irr ::core ::vector3df **getPtApplic** ()
renvoie le point d'application de la force
- void **setForce** (irr ::core ::vector3df posi=irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df forc=irr ::core ::vector3df(0, 0, 0))
definit la force et le point d'application de la force
- virtual **~CEffort** ()
Destructeur.

Attributs priv  s

- irr ::core ::vector3df **force**
le vecteur Force
- irr ::core ::vector3df **position**
le point d'application

7.47.1 Description détaillée

classe permettant de définir un effort appliqué à un objet

7.47.2 Documentation des constructeurs et destructeur

```
7.47.2.1 ayoaron : :physic : :CEffort : :CEffort( irr : :core : :vector3df posi =  
irr : :core : :vector3df( 0, 0, 0 ), irr : :core : :vector3df forc =  
irr : :core : :vector3df( 0, 0, 0 ) )
```

Constructeur.

Références force, et position.

7.47.2.2 ayoaron ::physic ::CEffort ::~CEffort() [virtual]

Destructeur.

7.47.3 Documentation des fonctions membres

7.47.3.1 irr::core::vector3df ayoaron::physic::CEffort::getDirection ()

renvoie le vecteur direction de la force

Renvoie

la direction de la force

Références force.

7.47.3.2 irr::core::vector3df ayoaron::physic::CEffort::getForce(void) [inline]

renvoie le vecteur force

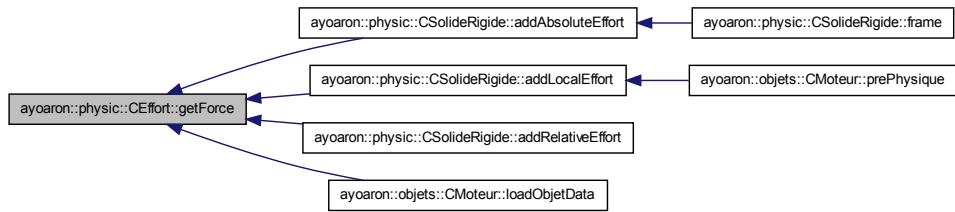
Renvoie

la force

Références force.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), ayoaron ::physic ::CSolideRigide ::addLocalEffort(), ayoaron ::physic ::CSolideRigide ::addRelativeEffort(), et ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.47.3.3 irr ::core ::vector3df ayoaron ::physic ::CEffort ::getMoment ()

renvoie le moment à l'origine du repère qui a servit pour la définition

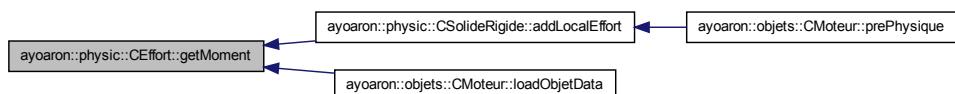
Renvoie

la force

Références force, et position.

Référencé par ayoaron ::physic ::CSolideRigide ::addLocalEffort(), et ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.47.3.4 irr ::f64 ayoaron ::physic ::CEffort ::getNorme ()

renvoie la norme de la force

Renvoie

la norme de la force

Références force.

7.47.3.5 irr ::core ::vector3df ayoaron ::physic ::CEffort ::getPtAppli() [inline]

renvoie le point d'application de la force

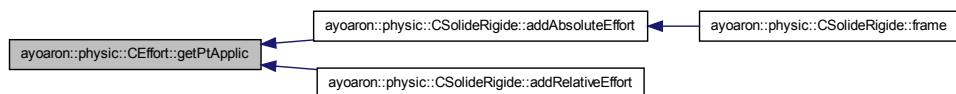
Renvoie

le point d'application

Références position.

Référencé par ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort(), et ayoaron ::physic ::CSolideRigide ::addRelativeEffort().

Voici le graphe d'appel pour cette fonction :



**7.47.3.6 void ayoaron ::physic ::CEffort ::setForce (irr ::core ::vector3df *posi* =
 irr ::core ::vector3df (0, 0, 0), irr ::core ::vector3df *forc* =
 irr ::core ::vector3df (0, 0, 0)) [inline]**

definit la force et le point d'application de la force

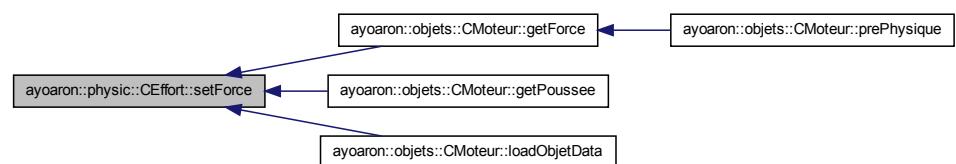
Paramètres

<i>posi</i>	la position de la force
<i>forc</i>	la valeur de la force

Références force, et position.

Référencé par ayoaron ::objets ::CMoteur ::getForce(), ayoaron ::objets ::CMoteur ::getPoussee(), et ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :

**7.47.4 Documentation des données membres**

7.47.4.1 irr ::core ::vector3df ayoaron ::physic ::CEffort ::force [private]

le vecteur Force

Référencé par CEffort(), getDirection(), getForce(), getMoment(), getNorme(), et setForce().

7.47.4.2 irr ::core ::vector3df ayoaron ::physic ::CEffort ::position [private]

le point d'application

Référencé par CEffort(), getMoment(), getPtApplic(), et setForce().

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

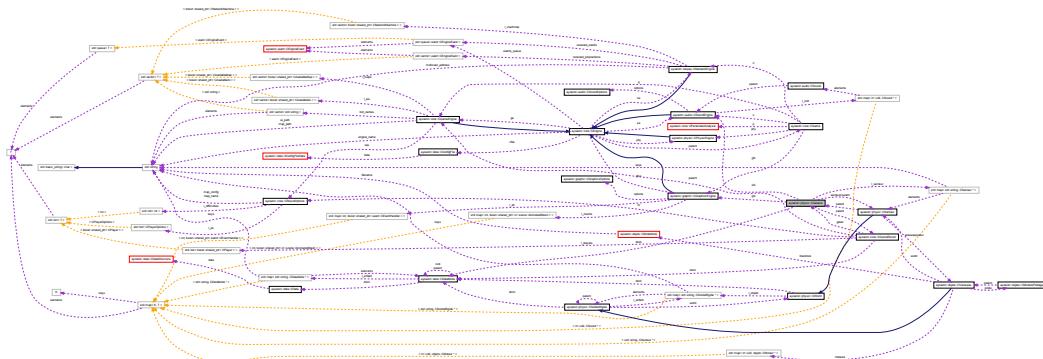
- [CEffort.h](#)
- [CEffort.cpp](#)

7.48 Référence de la classe ayoaron ::physic ::CGalaxie

la classe décrivant la galaxie

```
#include <CGalaxie.h>
```

Graphe de collaboration de ayoaron ::physic ::CGalaxie :



Fonctions membres publiques

- [CGalaxie \(core ::CGameRound *p, graphic ::CGraphicsEngine *ge, audio ::CSoundEngine *se\)](#)

Constructeur.
- [void frame \(double dt\)](#)

mise à jour
- [data ::CDataNode * getDonnee \(\)](#)

pointeur vers les données
- [graphic ::CGraphicsEngine * getGraphicsEngine \(void\)](#)

renvoie le moteur graphique parent
- [std ::string getNom \(void\)](#)

récupération du nom
- [irr ::u32 getNSecteur \(\)](#)

- **CSecteur * getSecteur (std ::string i)**
renvoie le secteur
- **CSecteur * getSecteurCourant ()**
renvoie le secteur courant
- **audio ::CSoundEngine * getSoundEngine (void)**
renvoie le moteur de sons parent
- **void loadData (data ::CDataNode *n)**
Charge la galaxie depuis un fichier.
- **void setNom (std ::string s)**
défini le nom
- **void setSecteurCourant (std ::string i)**
défini le secteur courant par numéro
- **virtual ~CGalaxie ()**
Destructeur.

Attributs privés

- **data ::CDataNode * donn**
pointeur vers les données
- **graphic ::CGraphicsEngine * gfx**
pointeur vers le moteur graphique
- **std ::map< std ::string, CSecteur * > l_secteur**
liste des secteurs de la galaxie
- **core ::CGameRound * parent**
pointeur vers le round parent
- **CSecteur * secteurcourant**
pointeur vers le secteur courant
- **audio ::CSoundEngine * sfx**
pointeur vers le moteur de son

7.48.1 Description détaillée

la classe décrivant la galaxie

7.48.2 Documentation des constructeurs et destructeur

7.48.2.1 ayoaron ::physic ::CGalaxie (core ::CGameRound * p, graphic ::CGraphicsEngine * ge, audio ::CSoundEngine * se)

Constructeur.

Références gfx, parent, et sfx.

7.48.2.2 ayoaron ::physic ::CGalaxie ::~CGalaxie () [virtual]

Destructeur.

7.48.3 Documentation des fonctions membres

7.48.3.1 void ayoaron ::physic ::CGalaxie ::frame (double *dt*)

mise à jour

Référencé par ayoaron ::core ::CGameRound ::frame().

Voici le graphe d'appel pour cette fonction :



7.48.3.2 data : :CDataNode* ayoaron ::physic ::CGalaxie ::getDonnee () [inline]

pointeur vers les données

Références donn.

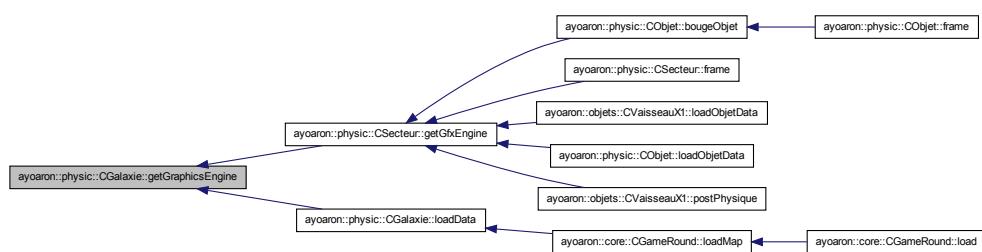
7.48.3.3 graphic : :CGraphicsEngine* ayoaron ::physic ::CGalaxie ::getGraphicsEngine (void) [inline]

renvoie le moteur graphique parent

Références gfx.

Référencé par ayoaron ::physic ::CSecteur ::getGfxEngine(), et loadData().

Voici le graphe d'appel pour cette fonction :



7.48.3.4 std ::string ayoaron ::physic ::CGalaxie ::getNom (void) [inline]

récupération du nom

Références donn, et ayoaron ::data ::CDataNode ::getstr().

Voici le graphe d'appel pour cette fonction :



7.48.3.5 irr : :u32 ayoaron ::physic ::CGalaxie ::getNSecteur() [inline]

renvoie le nombre de secteurs

Références l_secteur.

7.48.3.6 CSecteur* ayoaron ::physic ::CGalaxie ::getSecteur(std ::string i) [inline]

renvoie un secteur

Références l_secteur.

7.48.3.7 CSecteur* ayoaron ::physic ::CGalaxie ::getSecteurCourant() [inline]

renvoie le secteur courant

Références secteurcourant.

7.48.3.8 audio : :CSoundEngine* ayoaron ::physic ::CGalaxie ::getSoundEngine(void) [inline]

renvoie le moteur de sons parent

Références sfx.

Référencé par ayoaron ::physic ::CSecteur ::getSfxEngine().

Voici le graphe d'appel pour cette fonction :



7.48.3.9 void ayoaron ::physic ::CGalaxie ::loadData(data : :CDataNode * n)

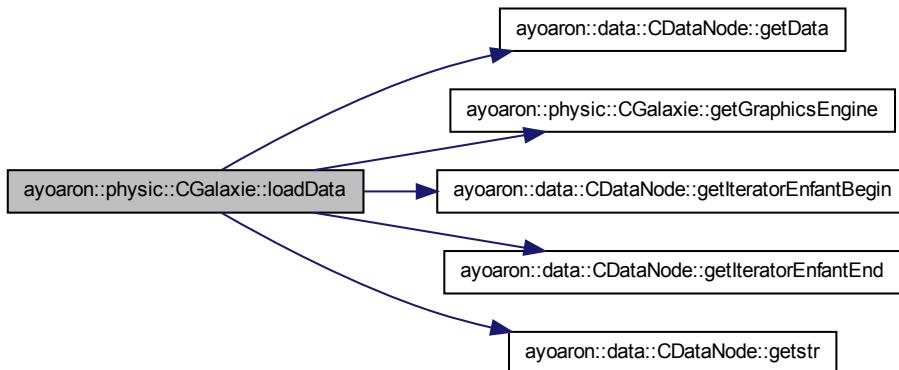
Charge la galaxie depuis un fichier.

Références donn, ayoaron ::data ::CDataNode ::getData(), getGraphicsEngine(), ayoaron ::data ::CDataNode ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::ge-

tIteratorEnfantEnd(), ayoaron ::data ::CDataNode ::getstr(), l_secteur, et secteurcourant.

Référencé par ayoaron ::core ::CGameRound ::loadMap().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.48.3.10 void ayoaron ::physic ::CGalaxie ::setNom (std ::string s) [inline]

défini le nom

Références donn, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.48.3.11 **void ayoaron ::physic ::CGalaxie ::setSecteurCourant (std ::string *i*)**
[inline]

défini le secteur courant par numéro

Références *I_secteur*, et *secteurcourant*.

7.48.4 Documentation des données membres

7.48.4.1 **data ::CDataNode* ayoaron ::physic ::CGalaxie ::donn** [private]

pointeur vers les données

Référencé par *getDonnee()*, *getNom()*, *loadData()*, et *setNom()*.

7.48.4.2 **graphic ::CGraphicsEngine* ayoaron ::physic ::CGalaxie ::gfx**
[private]

pointeur vers le moteur graphique

Référencé par *CGalaxie()*, et *getGraphicsEngine()*.

7.48.4.3 **std ::map<std ::string,CSecteur*> ayoaron ::physic ::CGalaxie ::I_-
secteur** [private]

liste des secteurs de la galaxie

Référencé par *getNSecteur()*, *getSecteur()*, *loadData()*, et *setSecteurCourant()*.

7.48.4.4 **core ::CGameRound* ayoaron ::physic ::CGalaxie ::parent**
[private]

pointeur vers le round parent

Référencé par *CGalaxie()*.

7.48.4.5 **CSecteur* ayoaron ::physic ::CGalaxie ::secteurcourant** [private]

pointeur vers le secteur courant

Référencé par *ayoaron ::reseau ::CNetworkEngine ::frame()*, *getSecteurCourant()*, *loadData()*, et *setSecteurCourant()*.

7.48.4.6 **audio ::CSoundEngine* ayoaron ::physic ::CGalaxie ::sfx**
[private]

pointeur vers le moteur de son

Référencé par *CGalaxie()*, et *getSoundEngine()*.

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

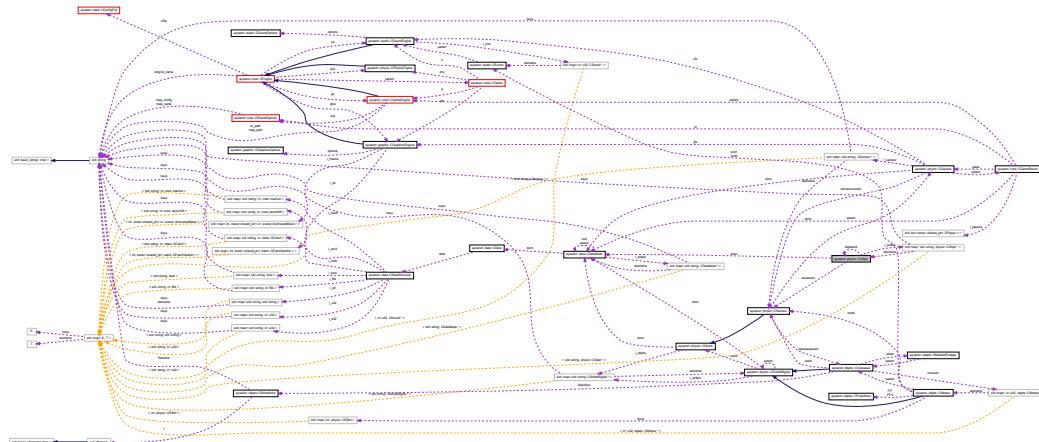
- [CGalaxie.h](#)
- [CGalaxie.cpp](#)

7.49 Référence de la classe ayoaron ::physic ::CObjet

Classe de base de tous les objets.

```
#include <CObjet.h>
```

Graphe de collaboration de ayoaron ::physic ::CObjet :



Fonctions membres publiques

- void **addAnimator** (irr ::scene ::ISceneNodeAnimator *a)
ajoute un animateur à l'objet
- virtual void **bougeObjet** (irr ::core ::vector3df translation=irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df rotation=irr ::core ::vector3df(0, 0, 0))
ajoute un effort dans le repère de l'objet le moment est donné au centre de l'objet
- **CObjet** (CSecteur *p, **CObjet** *po=NULL)
Constructeur.
- virtual void **frame** (irr ::f64 dt)
fonction d'incrément de temps
- irr ::core ::vector3df **getAbsoluteRotVelocity** ()
renvoie la vitesse de rotation dans le repère général
- irr ::core ::vector3df **getAbsoluteVelocity** ()
renvoie la vitesse dans le repère général
- virtual irr ::scene ::ICameraSceneNode * **getCam** ()
renvoie la camera
- virtual irr ::core ::matrix4 **getdInertie** ()
renvoie la variation de matrice d'inertie
- virtual irr ::f32 **getdMasse** ()
renvoie la variation de masse
- virtual irr ::core ::matrix4 **getInertie** ()
renvoie la matrice d'inertie
- irr ::core ::vector3df **getLocalRotVelocity** ()
renvoie la vitesse de rotation dans le repère local
- irr ::core ::vector3df **getLocalVelocity** ()
renvoie la vitesse dans le repère local
- irr ::f64 **getMasse** ()
renvoie la masse
- irr ::scene ::IAnimatedMesh * **getMesh** ()

- *renvoie le maillage*
– irr : :scene : :ISceneNode * **getNode** ()
 renvoie le scene node
- **CObjet * getObjetParent** (void)
 renvoie l'objet parent
- irr : :core : :vector3df **getRelativeRotVelocity** ()
 renvoie la vitesse de rotation dans le repère du parent... le stocké
- irr : :core : :vector3df **getRelativeVelocity** ()
 renvoie la vitesse dans le repère du parent (la vitesse stockée)
- irr : :scene : :ISceneManager * **getSceneManager** (void)
 renvoie le scene manager
- **CSecteur * getSecteurParent** (void)
 renvoie le secteur parent
- **virtual void loadObjetData** (data : :CDataNode *d)
 charge les données
- **virtual void preFrame** (irr : :f64 dt)
 preframe
- **void resetEffort** (void)
 mets à zero les efforts
- **void resetmouvement** (void)
 mets à zéro les vitesses
- **virtual void setPosition** (irr : :core : :vector3df pos=irr : :core : :vector3df(0, 0, 0),
 irr : :core : :vector3df rot=irr : :core : :vector3df(0, 0, 0))
 met en position le node
- **void setScale** (irr : :core : :vector3df scale=irr : :core : :vector3df(1, 1, 1))
 définit l'échelle
- **virtual void setVitesse** (irr : :core : :vector3df vit=irr : :core : :vector3df(0, 0, 0), irr : :core : :vector3df rit=irr : :core : :vector3df(0, 0, 0))
 definit la vitesse absolue de l'objet
- **virtual ~CObjet** ()
 Destructeur.

Fonctions membres protégées

- **virtual void setInertie** (irr : :core : :matrix4 a)
 définition de la matrice d'inertie

Attributs protégés

- irr : :scene : :ICameraSceneNode * **cam**
 la camera
- irr : :core : :vector3df **caml**
- irr : :core : :vector3df **camp**
- **data : :CDataNode * donn**
 pointeur vers la structure de donnée
- irr : :core : :vector3df **EffortResultant**
 l'effort résultant
- **bool gravityAffective**
 savoir si la gravité affecte cet objet
- **bool hascamera**
 si cet objet porte la camera
- **bool isResultanteNulle**
 si les resultantes sont nulles
- **std : :map< std : :string, physic ::CObjet * > l_enfant**
 liste des objets enfant
- irr : :scene : :IAnimatedMesh * **mesh**
 un pointeur vers le maillage de l'objet

- irr ::core ::vector3df **MomentResultant**
le moment résultant
- irr ::u32 **NID**
le numéro de l'objet
- irr ::scene ::ISceneNode * **node**
un pointeur vers le node dans la scenegraph
- irr ::video ::ITexture * **normalMap**
une texture des normales pour le bump mapping
- **CObjet * objparent**
pointeur vers l'objet parent
- **CSecteur * separent**
un pointeur vers le secteur parent

7.49.1 Description détaillée

Classe de base de tous les objets.

un objet est la combinaison d'un solide physique, d'un maillage graphique et d'un environnement sonore

7.49.2 Documentation des constructeurs et destructeur

7.49.2.1 ayoaron ::physic ::CObjet ::CObjet (CSecteur * *p*, CObjet * *po* = NULL)

Constructeur.

Références donn, EffortResultant, hascamera, isResultanteNulle, mesh, MomentResultant, node, normalMap, objparent, et separent.

7.49.2.2 ayoaron ::physic ::CObjet ::~CObjet () [virtual]

Destructeur.

la destruction de l'objet ne détruit pas les données correspondante ! mais détruit tous les objets enfant par récursivité.

Références *l_enfant*.

7.49.3 Documentation des fonctions membres

7.49.3.1 void ayoaron ::physic ::CObjet ::addAnimator (irr ::scene ::ISceneNodeAnimator * *a*) [inline]

ajoute un animateur à l'objet

Références node.

7.49.3.2 void ayoaron ::physic ::CObjet ::bougeObjet (irr ::core ::vector3df *translation* = irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df *rotation* = irr ::core ::vector3df(0, 0, 0)) [virtual]

ajoute un effort dans le repère de l'objet le moment est donné au centre de l'objet

ajoute un effort dans le repère du parent le moment est donné au centre de l'objet ajoute un effort donné dans le repère general le moment est donné au centre de l'objet fait bouger le node

la trans

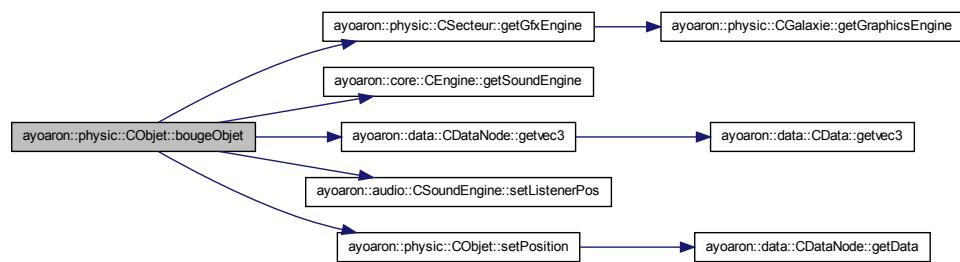
À faire

la camera est fixe par rapport à l'objet, la rendre bougeante

Références cam, caml, donn, ayoaron ::physic ::CSecteur ::getGfxEngine(), ayoaron ::core ::CEngine ::getSoundEngine(), ayoaron ::data ::CDataNode ::getvec3(), hascamera, node, secparent, ayoaron ::audio ::CSoundEngine ::setListenerPos(), et setPosition().

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.49.3.3 void ayoaron ::physic ::CObjet ::frame (irr ::f64 dt) [virtual]

fonction d'incrément de temps

on preframe tous les enfants éventuels

gestion du champ de force global

ce champ est donné dans le référentiel global

mouvement

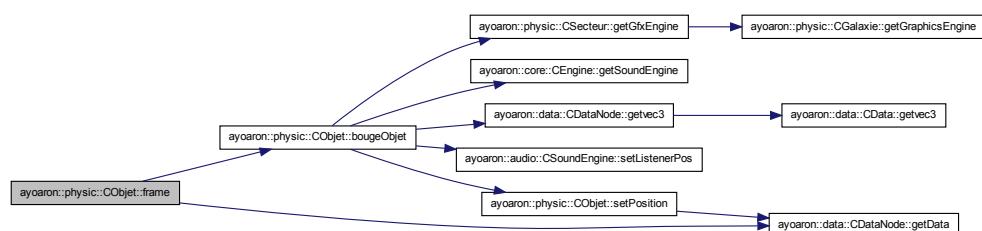
calcul du torseur dynamique dans le repère global (Efforts et moments résultants étant exprimés dans le repère local) dans le cas de l'accélération, étant donné que le repère local est lié au solide il n'y a pas d'accélération relative, il suffit de faire tourner le vecteur. pour le moment, on ne change pas le point d'application il suffit là encore de le faire tourner

on met à jour la position globale au cas où les enfants en ait besoin ou le secteur parent pour les collisions

on frame tous les enfants éventuels

Références bougeObjet(), donn, EffortResultant, ayoaron ::data ::CDataNode ::getData(), l_enfant, MomentResultant, et node.

Voici le graphe d'appel pour cette fonction :



7.49.3.4 irr ::core ::vector3df ayoaron ::physic ::CObjet ::getAbsoluteRotVelocity ()

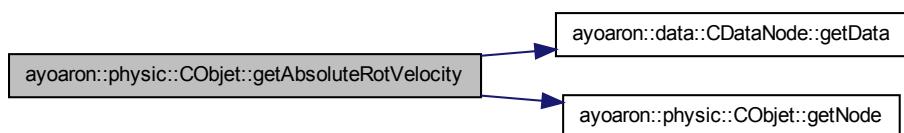
renvoie la vitesse de rotation dans le repère général

À faire

include la rotation du parent

Références donn, ayoaron ::data ::CDataNode ::getData(), getNode(), node, et obj-parent.

Voici le graphe d'appel pour cette fonction :



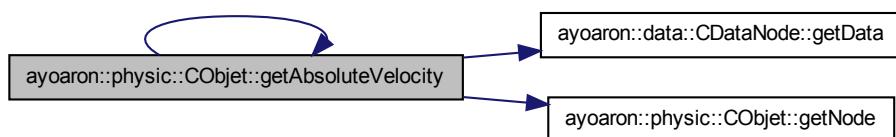
7.49.3.5 irr ::core ::vector3df ayoaron ::physic ::CObjet ::getAbsoluteVelocity ()

renvoie la vitesse dans le repère général

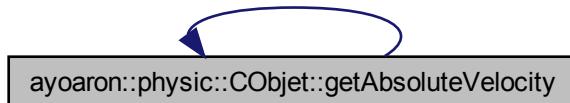
Références donn, getAbsoluteVelocity(), ayoaron ::data ::CDataNode ::getData(), getNode(), node, et objparent.

Référencé par getAbsoluteVelocity().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

7.49.3.6 virtual irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CObjet ::getCam ()
[inline, virtual]

renvoie la camera

Références cam.

7.49.3.7 irr ::core ::matrix4 ayoaron ::physic ::CObjet ::getdInertie () [virtual]

renvoie la variation de matrice d'inertie

7.49.3.8 virtual irr ::f32 ayoaron ::physic ::CObjet ::getdMasse () [inline, virtual]

renvoie la variation de masse

7.49.3.9 irr ::core ::matrix4 ayoaron ::physic ::CObjet ::getInertie () [virtual]

renvoie la matrice d'inertie

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :



7.49.3.10 irr ::core ::vector3df ayoaron ::physic ::CObjet ::getLocalRotVelocity ()

renvoie la vitesse de rotation dans le repère local

Références donn, ayoaron ::data ::CDataNode ::getData(), et node.

Voici le graphe d'appel pour cette fonction :



7.49.3.11 irr ::core ::vector3df ayoaron ::physic ::CObjet ::getLocalVelocity ()

renvoie la vitesse dans le repère local

Références donn, ayoaron ::data ::CDataNode ::getData(), et node.

Voici le graphe d'appel pour cette fonction :



7.49.3.12 irr ::f64 ayoaron ::physic ::CObjet ::getMasse ()

renvoie la masse

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :

7.49.3.13 irr ::scene ::IAnimatedMesh* ayoaron ::physic ::CObjet ::getMesh ()
[inline]

renvoie le maillage

Références mesh.

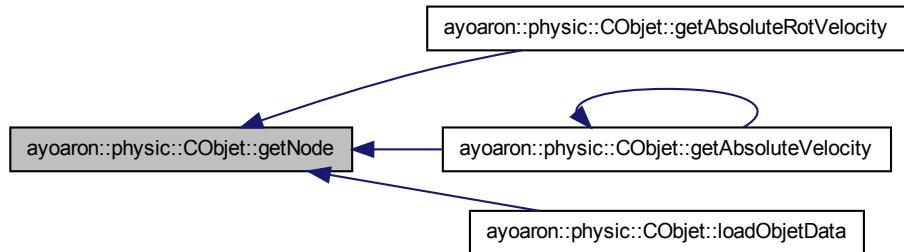
7.49.3.14 irr ::scene ::ISceneNode* ayoaron ::physic ::CObjet ::getNode () [inline]

renvoie le scene node

Références node.

Référencé par getAbsoluteRotVelocity(), getAbsoluteVelocity(), et loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.49.3.15 `irr ::core ::vector3df ayoaron ::physic ::CObjet ::getObjetParent (void) [inline]`

renvoie l'objet parent

Références objparent.

7.49.3.16 `irr ::core ::vector3df ayoaron ::physic ::CObjet ::getRelativeRotVelocity () [inline]`

renvoie la vitesse de rotation dans le repère du parent... le stocké

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :



7.49.3.17 `irr ::core ::vector3df ayoaron ::physic ::CObjet ::getRelativeVelocity () [inline]`

renvoie la vitesse dans le repère du parent (la vitesse stockée)

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :



7.49.3.18 `irr ::scene ::ISceneManager* ayoaron ::physic ::CObjet ::getSceneManager (void) [inline]`

renvoie le scene manager

Références node.

7.49.3.19 `CSecteur* ayoaron ::physic ::CObjet ::getSecteurParent (void) [inline]`

renvoie le secteur parent

Références separent.

7.49.3.20 `void ayoaron ::physic ::CObjet ::loadObjetData (data ::CDataNode * d)`
`[virtual]`

charge les données

À faire

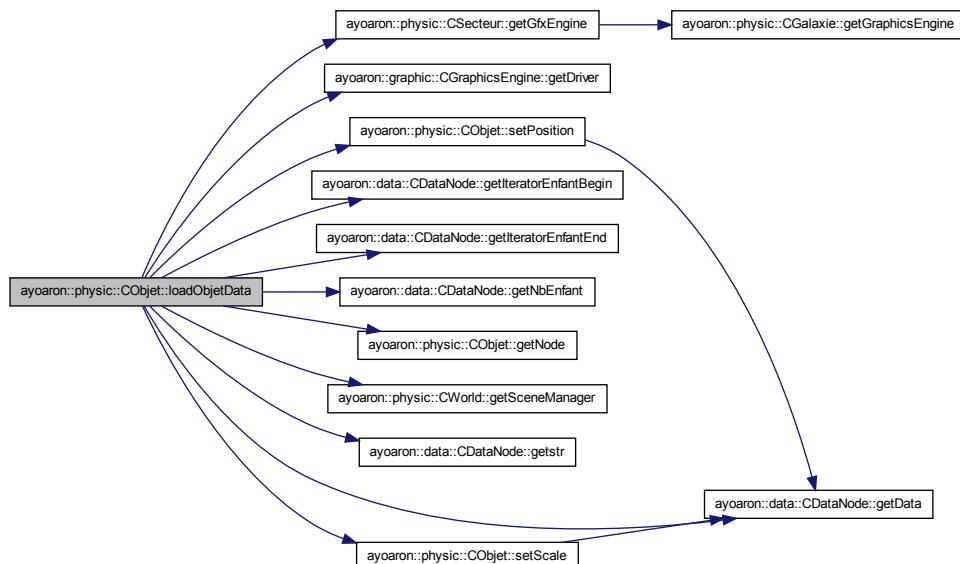
rendre disponible plus de paramètres de texture

À faire

faire le lien avec les options globales de traitement

Références donn, ayoaron ::data ::CDataNode ::getData(), ayoaron ::graphic ::CGraphicsEngine ::getDriver(), ayoaron ::physic ::CSecteur ::getGfxEngine(), ayoaron ::data ::CDataNode ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::getIteratorEnfantEnd(), ayoaron ::data ::CDataNode ::getNbEnfant(), getNode(), ayoaron ::physic ::CWorld ::getSceneManager(), ayoaron ::data ::CDataNode ::getstr(), mesh, node, normalMap, obj-parent, separent, setPosition(), setScale(), et ayoaron ::data ::STRING.

Voici le graphe d'appel pour cette fonction :



7.49.3.21 `virtual void ayoaron ::physic ::CObjet ::preFrame (irr ::f64 dt)` [inline,
`virtual`]

preframe

7.49.3.22 void ayoaron ::physic ::CObjet ::resetEffort (void)

mets à zero les efforts

Références EffortResultant, isResultanteNulle, et MomentResultant.

7.49.3.23 void ayoaron ::physic ::CObjet ::resetmouvement (void)

mets à zéro les vitesses

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :

7.49.3.24 void ayoaron ::physic ::CObjet ::setInertie (irr ::core ::matrix4 a)
[protected, virtual]

définition de la matrice d'inertie

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :

7.49.3.25 void ayoaron ::physic ::CObjet ::setPosition (irr ::core ::vector3df pos =
irr ::core ::vector3df(0,0,0), irr ::core ::vector3df rot =
irr ::core ::vector3df(0,0,0)) [virtual]

met en position le node

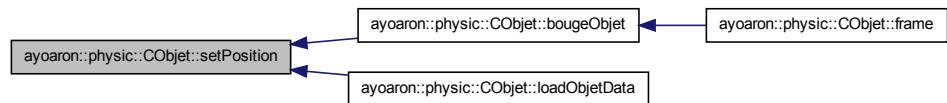
Références donn, ayoaron ::data ::CDataNode ::getData(), et node.

Référencé par bougeObjet(), et loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.49.3.26 void ayoaron :: physic :: CObjet :: setScale (irr :: core :: vector3df scale = irr :: core :: vector3df(1,1,1))

définit l'échelle

Références donn, ayoaron :: data :: CDataNode :: getData(), et node.

Référencé par loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.49.3.27 `void ayoaron ::physic ::CObjet ::setVitesse (irr ::core ::vector3df vit = irr ::core ::vector3df(0, 0, 0), irr ::core ::vector3df rit = irr ::core ::vector3df(0, 0, 0)) [virtual]`

definit la vitesse absolue de l'objet

Références donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :



7.49.4 Documentation des données membres

7.49.4.1 `irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CObjet ::cam [protected]`

la camera

Référencé par bougeObjet(), et getCam().

7.49.4.2 `irr ::core ::vector3df ayoaron ::physic ::CObjet ::cam1 [protected]`

Référencé par bougeObjet().

7.49.4.3 `irr ::core ::vector3df ayoaron ::physic ::CObjet ::camp [protected]`

7.49.4.4 `data ::CDataNode* ayoaron ::physic ::CObjet ::donn [protected]`

pointeur vers la structure de donnée

Référencé par bougeObjet(), CObjet(), frame(), getAbsoluteRotVelocity(), getAbsoluteVelocity(), getInertie(), getLocalRotVelocity(), getLocalVelocity(), getMasse(), getRelati-

veRotVelocity(), getRelativeVelocity(), loadObjetData(), resetmouvement(), setInertie(), setPosition(), setScale(), et setVitesse().

7.49.4.5 **irr ::core ::vector3df ayoaron ::physic ::CObjet ::EffortResultant** [protected]

l'effort résultant

Référencé par CObjet(), frame(), et resetEffort().

7.49.4.6 **bool ayoaron ::physic ::CObjet ::gravityAffective** [protected]

savoir si la gravité affecte cet objet

7.49.4.7 **bool ayoaron ::physic ::CObjet ::hascamera** [protected]

si cet objet porte la camera

Référencé par bougeObjet(), et CObjet().

7.49.4.8 **bool ayoaron ::physic ::CObjet ::isResultanteNulle** [protected]

si les resultantes sont nulles

Référencé par CObjet(), et resetEffort().

7.49.4.9 **std ::map<std ::string,physic ::CObjet*> ayoaron ::physic ::CObjet ::l_enfant** [protected]

liste des objets enfant

Référencé par frame(), et ~CObjet().

7.49.4.10 **irr ::scene ::IAnimatedMesh* ayoaron ::physic ::CObjet ::mesh** [protected]

un pointeur vers le maillage de l'objet

Référencé par CObjet(), getMesh(), et loadObjetData().

7.49.4.11 **irr ::core ::vector3df ayoaron ::physic ::CObjet ::MomentResultant** [protected]

le moment résultant

Référencé par CObjet(), frame(), et resetEffort().

7.49.4.12 **irr ::u32 ayoaron ::physic ::CObjet ::NID** [protected]

le numéro de l'objet

7.49.4.13 **irr ::scene ::ISceneNode* ayoaron ::physic ::CObjet ::node** [protected]

un pointeur vers le node dans la scenegraph

Référencé par addAnimator(), bougeObjet(), CObjet(), frame(), getAbsoluteRotVelocity(), getAbsoluteVelocity(), getLocalRotVelocity(), getLocalVelocity(), getNode(), getSceneManager(), loadObjetData(), setPosition(), et setScale().

7.49.4.14 irr ::video ::ITexture* ayoaron ::physic ::CObjet ::normalMap [protected]

une texture des normales pour le bump mapping

Référencé par CObjet(), et loadObjetData().

7.49.4.15 CObjet* ayoaron ::physic ::CObjet ::objparent [protected]

pointeur vers l'objet parent

Référencé par CObjet(), getAbsoluteRotVelocity(), getAbsoluteVelocity(), getObjetParent(), et loadObjetData().

7.49.4.16 CSecteur* ayoaron ::physic ::CObjet ::secparent [protected]

un pointeur vers le secteur parent

Référencé par bougeObjet(), CObjet(), getSecteurParent(), et loadObjetData().

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

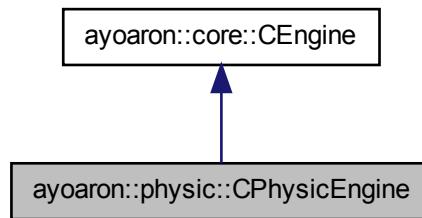
- [CObjet.h](#)
- [CObjet.cpp](#)

7.50 Référence de la classe ayoaron ::physic ::CPhysicEngine

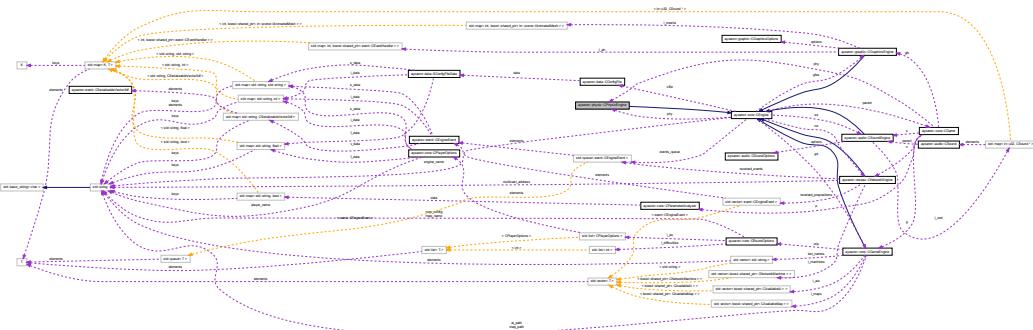
classe de description du moteur de physique

```
#include <CPhysicEngine.h>
```

Graphe d'héritage de ayoaron ::physic ::CPhysicEngine :



Graphe de collaboration de ayoaron ::physic ::CPhysicEngine :



Types publics

- enum {
 STATE_NOTHING, **STATE_QUITTING**, **STATE_WAITING_PLAYERS**, **STATE_PLAYING_INTRO**,
STATE_LOADING_ROUND, **STATE_PLAYING** }
- Constantes d'état.*

Fonctions membres publiques

- void **attachGameEngine** (CGameEngine *e)
lien avec le moteur de jeu
- void **attachGraphicsEngine** (graphic ::CGraphicsEngine *e)
lien avec le moteur de rendu graphique
- void **attachNetworkEngine** (reseau ::CNetworkEngine *e)
lien avec le moteur de réseau
- void **attachPhysicEngine** (physic ::CPhysicEngine *e)
lien avec le moteur de physique
- void **attachSoundEngine** (audio ::CSoundEngine *e)
lien avec le moteur de son
- **CPhysicEngine** (core ::CGame *p)
Constructeur.
- void **frame** (double dt)
Methode Frame.
- CGameEngine * **getGameEngine** ()
lien vers le moteur de jeu
- graphic ::CGraphicsEngine * **getGraphicsEngine** ()
lien vers le moteur de rendu graphique
- reseau ::CNetworkEngine * **getNetworkEngine** ()
lien vers le moteur de réseau
- CGame * **getParent** ()
lien vers la classe game parente
- audio ::CSoundEngine * **getSoundEngine** ()
lien vers le moteur de son
- void **processEvent** (event ::CEngineEvent &)
trate les événements
- void **processQueue** ()
vidange de l'event_queue

- void **pushEvent** (event : :CEngineEvent &e)
ajout d'un événement
- void **sendMessageToGame** (event : :CEngineEvent &e)
envoie un message au moteur de jeu
- void **sendMessageToGraphics** (event : :CEngineEvent &e)
envoie un message au moteur de rendu graphique
- void **sendMessageToNetwork** (event : :CEngineEvent &e)
envoie un message au moteur de réseau
- void **sendMessageToPhysic** (event : :CEngineEvent &e)
envoie un message au moteur physique
- void **sendMessageToSound** (event : :CEngineEvent &e)
envoie un message au moteur de son
- virtual ~CPhysicEngine ()
Destructeur.

Attributs publics

- int **state**
l'état actuel
- boost ::mutex **state_mutex**
Mutex de protection de l'état.

Attributs protégés

- **data** : :CConfigFile **cfile**
Fichier de configuration.
- std ::string **engine_name**
nom de ce moteur
- std ::queue< event : :CEngineEvent > **events_queue**
liste des événement à traiter
- CGameEngine * **ge**
pointeur vers le moteur de jeu
- CGraphicsEngine * **gixe**
pointeur vers le moteur de rendu graphique
- CNetworkEngine * **ne**
pointeur vers le moteur de réseau
- CGame * **parent**
pointeur vers le parent
- bool **passive_mode**
true si mode passif activé
- CPhysicEngine * **phy**
pointeur vers le moteur de physique
- boost ::mutex **queue_mutex**
Mutex de protection de la liste d'évenement event_queue.
- CSoundEngine * **se**
pointeur vers le moteur de son

Fonctions membres privées

- void **getConfigData** ()
récupération de la configuration actuelle

7.50.1 Description détaillée

classe de description du moteur de physique

7.50.2 Documentation des énumérations membres

7.50.2.1 anonymous enum [inherited]

Constantes d'état.

Valeurs énumérées :

STATE NOTHING
STATE QUITTING
STATE_WAITING_PLAYERS
STATE_PLAYING_INTRO
STATE_LOADING_ROUND
STATE_PLAYING

7.50.3 Documentation des constructeurs et destructeur

7.50.3.1 ayoaron ::physic ::CPhysicEngine ::CPhysicEngine (core ::CGame * p)

Constructeur.

7.50.3.2 ayoaron ::physic ::CPhysicEngine ::~CPhysicEngine () [virtual]

Destructeur.

7.50.4 Documentation des fonctions membres

7.50.4.1 void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e) [inline, inherited]

lien avec le moteur de jeu

Paramètres

in	e	pointeur vers le moteur de jeu à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.50.4.2 void ayoaron ::core ::CEngine ::attachGraphicsEngine (graphic : :CGraphicsEngine * e) [inline, inherited]

lien avec le moteur de rendu graphique

Paramètres

in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.50.4.3 void ayoaron ::core ::CEngine ::attachNetworkEngine (reseau ::CNetworkEngine * e) [inline, inherited]

lien avec le moteur de réseau

Paramètres

in	e	pointeur vers le moteur de réseau à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.50.4.4 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * e) [inline, inherited]

lien avec le moteur de physique

Paramètres

in	e	pointeur vers le moteur de physique à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références ayoaron ::core ::CEngine ::phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.50.4.5 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline, inherited]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.50.4.6 void ayoaron ::physic ::CPhysicEngine ::frame (double dt) [virtual]

Methode Frame.

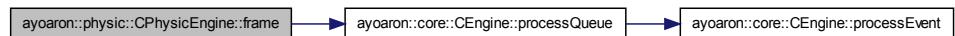
Cette méthode est utiliser pour mettre à jour le moteur de son

Implémenté [ayoaron ::core ::CEngine](#).

Références [ayoaron ::core ::CEngine ::processQueue\(\)](#).

Référencé par [ayoaron ::core ::CGame ::run\(\)](#).

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.50.4.7 void ayoaron ::physic ::CPhysicEngine ::getConfigData () [private, virtual]

récupération de la configuration actuelle

Implémenté [ayoaron ::core ::CEngine](#).

7.50.4.8 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine () [inline, inherited]

lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect(), ayoaron ::graphic ::CGuiSystem ::handleProcessCommand(), ayoaron ::graphic ::CGuiSystem ::loadGame(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::event ::CGameEventHandler ::onEvent(), et ayoaron ::graphic ::CGraphicsEngine ::processEvent().

Voici le graphe d'appel pour cette fonction :



7.50.4.9 graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine () [inline, inherited]

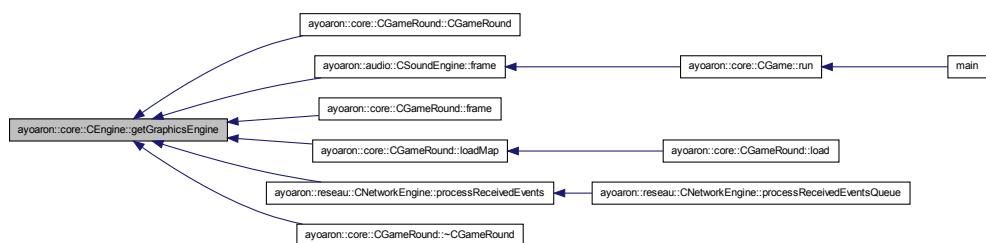
lien vers le moteur de rendu graphique

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::core ::CGameRound ::frame(), ayoaron ::core ::CGameRound ::loadMap(), ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.50.4.10 **reseau** : **CNetworkEngine*** ayoaron ::core ::CEngine ::getNetworkEngine ()
[inline, inherited]

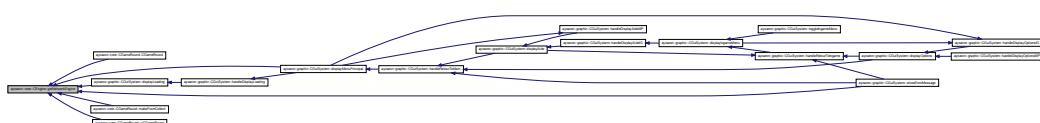
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.50.4.11 **CGame*** ayoaron ::core ::CEngine ::getParent () [inline, inherited]

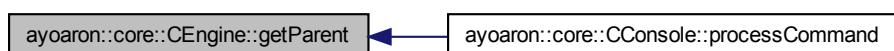
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.50.4.12 **audio** : **CSoundEngine*** ayoaron ::core ::CEngine ::getSoundEngine ()
[inline, inherited]

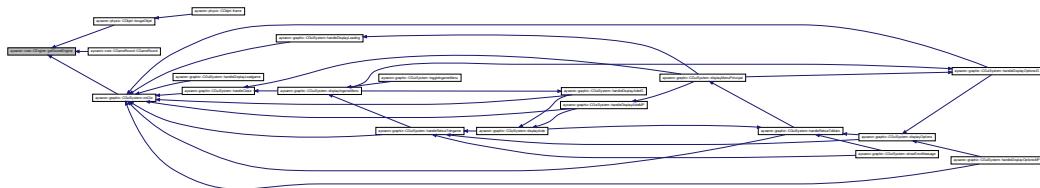
lien vers le moteur de son

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::core ::CGameRound ::CGameRound(), et ayoaron ::graphic ::CGuiSystem ::onClic().

Voici le graphe d'appel pour cette fonction :



7.50.4.13 void ayoaron ::physic ::CPhysicEngine ::processEvent (event : :CEvent &) [virtual]

traite les événements

Implémenté [ayoaron ::core ::CEngine](#).

7.50.4.14 void ayoaron ::core ::CEngine ::processQueue () [inherited]

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

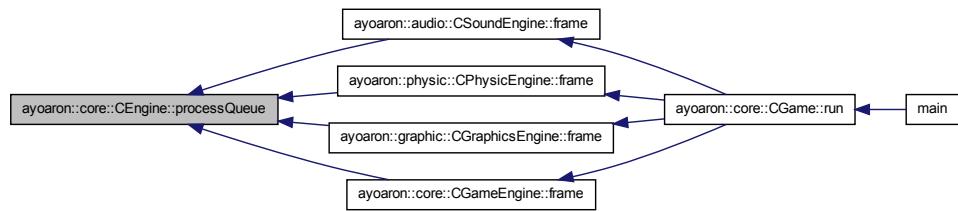
Références [ayoaron ::core ::CEngine ::events_queue](#), [ayoaron ::core ::CEngine ::processEvent\(\)](#), et [ayoaron ::core ::CEngine ::queue_mutex](#).

Référencé par [ayoaron ::audio ::CSoundEngine ::frame\(\)](#), [frame\(\)](#), [ayoaron ::graphic ::CGraphicsEngine ::frame\(\)](#), et [ayoaron ::core ::CGameEngine ::frame\(\)](#).

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.50.4.15 void ayoaron :: core :: CEngine :: pushEvent (event :: CEngineEvent & e)
 [inherited]

ajout d'un événement

Paramètres

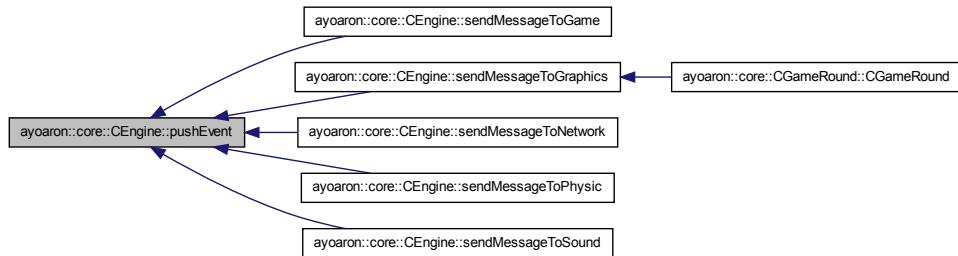
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références ayoaron :: core :: CEngine :: events_queue, et ayoaron :: core :: CEngine :: queue_mutex.

Référencé par ayoaron :: core :: CEngine :: sendMessageToGame(), ayoaron :: core :: CEngine :: sendMessageToGraphics(), ayoaron :: core :: CEngine :: sendMessageToNetwork(), ayoaron :: core :: CEngine :: sendMessageToPhysic(), et ayoaron :: core :: CEngine :: sendMessageToSound().

Voici le graphe d'appel pour cette fonction :



7.50.4.16 void ayoaron ::core ::CEngine ::sendMessageToGame (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de jeu

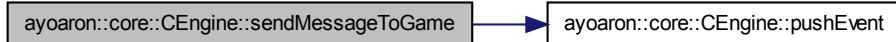
Paramètres

in	e	evenement à envoyer vers le moteur de jeu
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ayoaron ::core ::CEngine ::ge, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.50.4.17 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique

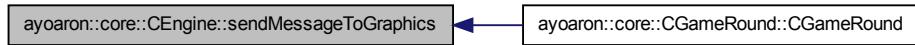
Références ayoaron ::core ::CEngine ::gixe, et ayoaron ::core ::CEngine ::pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.50.4.18 void ayoaron ::core ::CEngine ::sendMessageToNetwork (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ayoaron ::core ::CEngine ::ne, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.50.4.19 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e) [inherited]

envoie un message au moteur physique

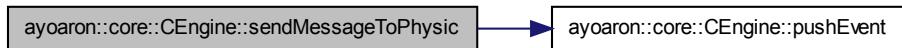
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de physique

Références ayoaron ::core ::CEngine ::phy, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.50.4.20 void ayoaron ::core ::CEngine ::sendMessageToSound (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de son

Références ayoaron ::core ::CEngine ::pushEvent(), et ayoaron ::core ::CEngine ::se.

Voici le graphe d'appel pour cette fonction :



7.50.5 Documentation des données membres

7.50.5.1 data ::CConfigFile ayoaron ::core ::CEngine ::cfile [protected, inherited]

Fichier de configuration.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), ayoaron ::réseau ::CNetworkEngine ::getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.50.5.2 std ::string ayoaron ::core ::CEngine ::engine_name [protected, inherited]

nom de ce moteur

Référencé par ayoaron ::core ::CGameEngine ::CGameEngine(), ayoaron ::gra-

phic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine().

7.50.5.3 std ::queue<event ::CEngineEvent> ayoaron ::core ::CEngine ::events _queue [protected, inherited]

liste des événement à traiter

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.50.5.4 CGameEngine* ayoaron ::core ::CEngine ::ge [protected, inherited]

pointeur vers le moteur de jeu

Référencé par ayoaron ::core ::CEngine ::attachGameEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::core ::CEngine ::sendMessageToGame(), et ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive().

7.50.5.5 graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::gixe [protected, inherited]

pointeur vers le moteur de rendu graphique

Référencé par ayoaron ::core ::CEngine ::attachGraphicsEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), et ayoaron ::core ::CGameEngine ::togglePause().

7.50.5.6 reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::ne [protected, inherited]

pointeur vers le moteur de réseau

Référencé par ayoaron ::core ::CEngine ::attachNetworkEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), ayoaron ::reseau ::CNetworkEngine ::sendEventTCP(), ayoaron ::reseau ::CNetworkEngine ::sendEventUDP(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive(), et ayoaron ::reseau ::CNetworkEngine ::udpAsyncRead().

7.50.5.7 CGame* ayoaron ::core ::CEngine ::parent [protected, inherited]

pointeur vers le parent

Référencé par ayoaron ::graphic ::CGraphicsEngine ::frame(), ayoaron ::core ::CEngine ::getParent(), et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

7.50.5.8 **bool ayoaron ::core ::CEngine ::passive_mode** [protected, inherited]

true si mode passif activé

Référencé par ayoaron ::core ::CEngine ::CEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::audio ::CSoundEngine ::CSoundEngine().

7.50.5.9 **physic ::CPhysicEngine* ayoaron ::core ::CEngine ::phy** [protected, inherited]

pointeur vers le moteur de physique

Référencé par ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::CEngine(), et ayoaron ::core ::CEngine ::sendMessageToPhysic().

7.50.5.10 **boost ::mutex ayoaron ::core ::CEngine ::queue_mutex** [protected, inherited]

Mutex de protection de la liste d'événement event_queue.

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.50.5.11 **audio ::CSoundEngine* ayoaron ::core ::CEngine ::se** [protected, inherited]

pointeur vers le moteur de son

Référencé par ayoaron ::core ::CEngine ::attachSoundEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getSoundEngine(), et ayoaron ::core ::CEngine ::sendMessageToSound().

7.50.5.12 **int ayoaron ::core ::CEngine ::state** [inherited]

l'état actuel

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::physic ::CSecteur ::frame(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP(), ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine ::listenMulti3secondsUDP(), ayoaron ::reseau ::CNetworkEngine ::multicastReceive(), ayoaron ::graphic ::CGraphicsEngine ::setState(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

7.50.5.13 **boost ::mutex ayoaron ::core ::CEngine ::state_mutex** [inherited]

Mutex de protection de l'état.

Référencé par ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP(),

ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), ayoaron ::reseau ::CNetworkEngine ::threadSendMulticast(), et ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine().

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

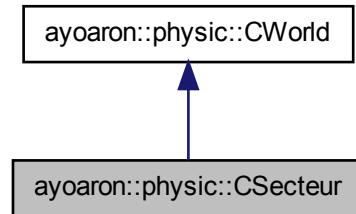
- [CPhysicEngine.h](#)
- [CPhysicEngine.cpp](#)

7.51 Référence de la classe ayoaron :: physic :: CSecteur

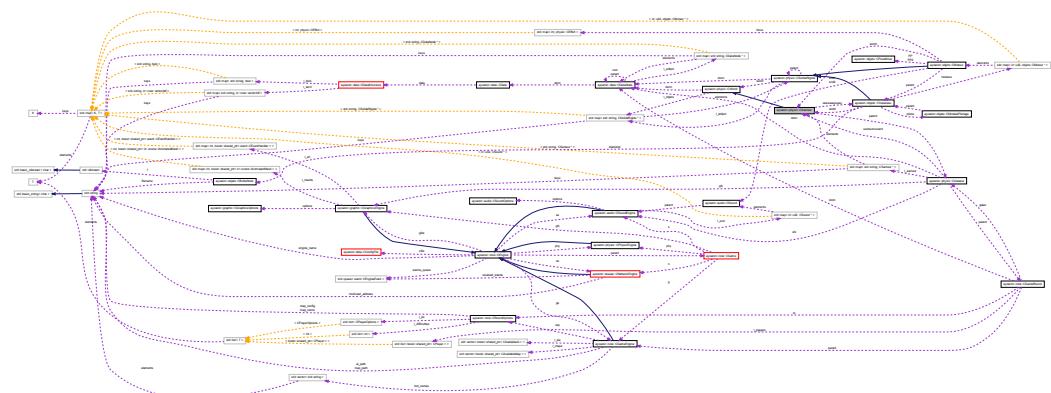
la classe décrivant un secteur de la galaxie

```
#include <CSecteur.h>
```

Graphe d'héritage de ayoaron :: physic :: CSecteur :



Graphe de collaboration de ayoaron :: physic :: CSecteur :



Fonctions membres publiques

- **CSecteur (CGalaxie *gal, data : :CDataNode *d, irr : :scene : :SceneManager *smg, unsigned int ID, bool actif)**
Constructeur.
- void **frame (double dt)**
la frame
- **data : :CDataNode * getDataNode ()**
renvoie un pointeur vers les options
- **graphic : :CGraphicsEngine * getGfxEngine (void)**
renvoie le moteur graphique
- **CEffort getGlobalEffort (CSolideRigide *obj)**
renvoie l'effort global au point donné (gravité, champ électrique, etc...)
- std : :string **getNom (void)**
récupération du nom
- **CGalaxie * getParent ()**
renvoie la galaxie parente
- irr : :scene : :SceneManager * **getSceneManager ()**
permet l'accès au gestionnaire de scène
- **audio : :CSoundEngine * getSfxEngine (void)**
renvoie le moteur de son
- virtual void **loadData (data : :CDataNode *o=NULL)**
Charge la galaxie depuis un fichier.
- void **setNom (std : :string s)**
défini le nom
- virtual **~CSecteur ()**
Destructeur.

Attributs protégés

- **data : :CDataNode * donn**
pointeur vers la structure de donnée
- std : :map< std : :string, CSolideRigide * > **l_objets**
liste des objets enfant
- irr : :scene : :SceneManager * **smgr**
pointeur vers le scene manager

Attributs privés

- irr : :u32 **IDsecteur**
numéro du secteur
- **CGalaxie * parent**
pointeur vers la galaxie parente
- **objets : :CVaisseau * vaisseaujoueur**
vaisseau du joueur

7.51.1 Description détaillée

la classe décrivant un secteur de la galaxie

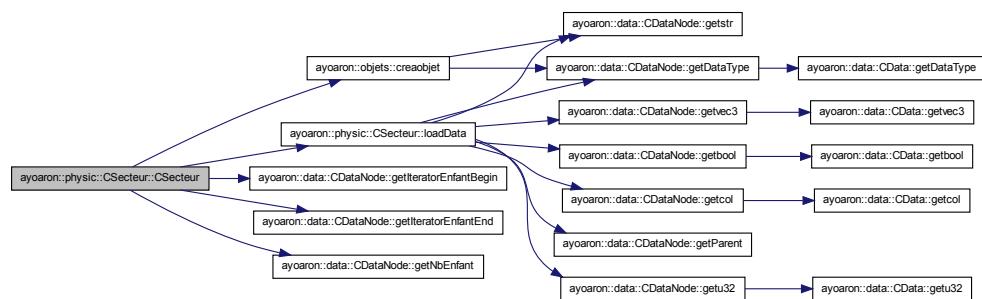
7.51.2 Documentation des constructeurs et destructeur

7.51.2.1 ayoaron ::physic ::CSecteur ::CSecteur (CGalaxie * gal, data ::CDataNode * d, irr ::scene ::SceneManager * smg, unsigned int ID, bool actif)

Constructeur.

Références ayoaron ::objets ::creaobjet(), ayoaron ::physic ::CWorld ::donn, ayoaron ::data ::CDataNode ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::getIteratorEnfantEnd(), ayoaron ::data ::CDataNode ::getNbEnfant(), IDsecteur, ayoaron ::physic ::CWorld ::l_objets, loadData(), parent, et vaisseaujoueur.

Voici le graphe d'appel pour cette fonction :



7.51.2.2 ayoaron ::physic ::CSecteur ::~CSecteur () [virtual]

Destructeur.

7.51.3 Documentation des fonctions membres

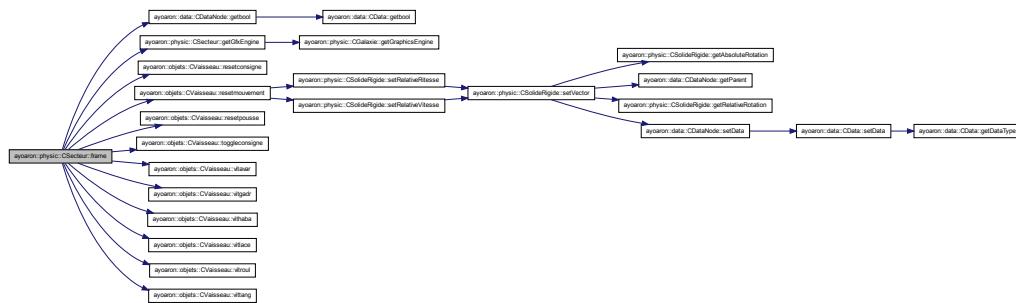
7.51.3.1 void ayoaron ::physic ::CSecteur ::frame (double dt) [virtual]

la frame

Réimplémentée à partir de [ayoaron ::physic ::CWorld](#).

Références ayoaron ::physic ::CWorld ::donn, ayoaron ::data ::CDataNode ::getBool(), getGfxEngine(), ayoaron ::graphic ::CGraphicsEngine ::l_eh, ayoaron ::physic ::CWorld ::l_objets, ayoaron ::objets ::CVaisseau ::resetconsigne(), ayoaron ::objets ::CVaisseau ::resetmouvement(), ayoaron ::objets ::CVaisseau ::resetpousse(), ayoaron ::core ::CEngine ::state, ayoaron ::objets ::CVaisseau ::toggleconsigne(), vaisseaujoueur, ayoaron ::objets ::CVaisseau ::vitavar(), ayoaron ::objets ::CVaisseau ::vitgadr(), ayoaron ::objets ::CVaisseau ::vithaba(), ayoaron ::objets ::CVaisseau ::vitlace(), ayoaron ::objets ::CVaisseau ::vitroul(), et ayoaron ::objets ::CVaisseau ::vittang().

Voici le graphe d'appel pour cette fonction :



7.51.3.2 **data** : **:CDataNode*** **ayoaron** : **:physic** : **:CSecteur** : **:getDataNode** ()
[inline]

renvoie un pointeur vers les options

Références ayoaron ::physic ::CWorld ::donn.

7.51.3.3 **graphic** : **:CGraphicsEngine*** ayoaron : **:physic** : **:CSecteur** : **:getGfxEngine** (**void**) [inline]

renvoie le moteur graphique

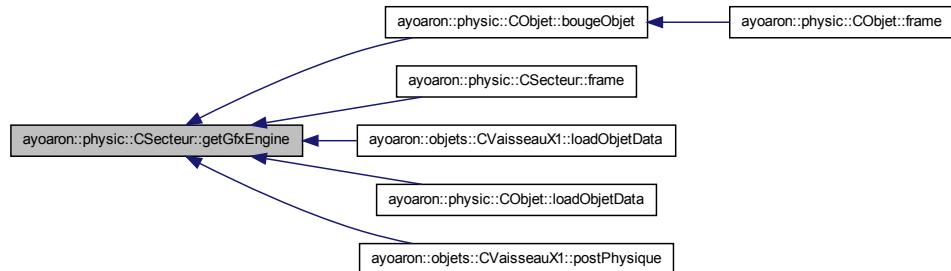
Références ayoaron : `:physic` : `:CGalaxie` : `:getGraphicsEngine()`, et parent.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), frame(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.51.3.4 CEffort ayoaron ::physic ::CWorld ::getGlobalEffort (CSolideRigide * obj) [inherited]

revoie l'effort global au point donné (gravité, champ électrique, etc...)

Paramètres

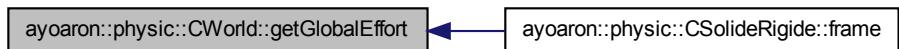
obj l'objet sur lequel dois s'appliquer l'effort global

Renvoie

la valeur de l'effort au centre de gravité de l'objet exprimé dans le repère global

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.51.3.5 std ::string ayoaron ::physic ::CSecteur ::getNom (void) [inline]

récupération du nom

Références ayoaron ::physic ::CWorld ::donn, et ayoaron ::data ::CDataNode ::getData().

Voici le graphe d'appel pour cette fonction :



7.51.3.6 CGalaxie* ayoaron ::physic ::CSecteur ::getParent() [inline]

renvoie la galaxie parente

Références parent.

7.51.3.7 irr ::scene ::ISceneManager* ayoaron ::physic ::CWorld ::getSceneManager() [inline, inherited]

permet l'accès au gestionnaire de scène

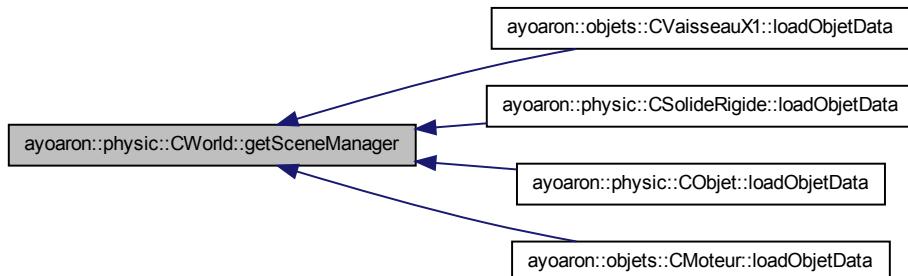
Renvoie

un pointeur vers le ISceneManager courrant

Références ayoaron ::physic ::CWorld ::smgr.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), et ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.51.3.8 **audio** : **CSoundEngine*** ayoaron ::physic ::CSecteur ::getSfxEngine (**void**)
 [inline]

renvoie le moteur de son

Références ayoaron ::physic ::CGalaxie ::getSoundEngine(), et parent.

Référencé par ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.51.3.9 **void** ayoaron ::physic ::CSecteur ::loadData (**data** : **CDataNode** * **o** = **NULL**)
 [virtual]

Charge la galaxie depuis un fichier.

recuperation des options

charge la skybox si la clé existe

les lumières

definition de la lumière ambiante si la clé existe

Definition de la lumière distante si la couleur et la direction ont été définies

le brouillard

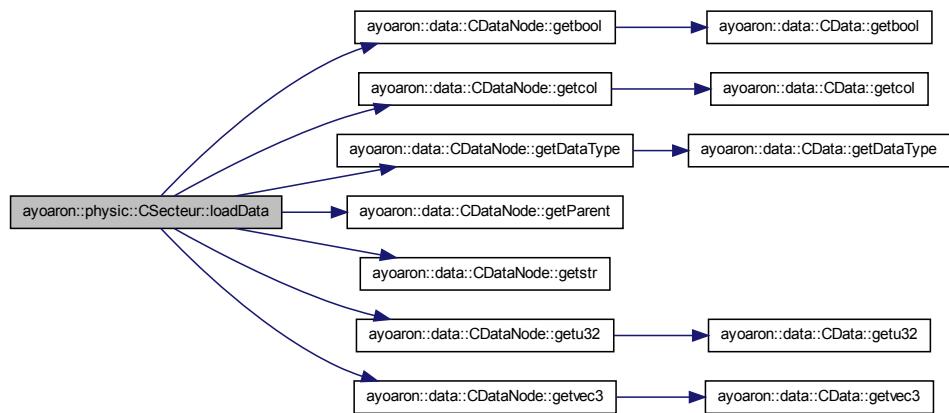
Réimplémentée à partir de [ayoaron ::physic ::CWorld](#).

Références ayoaron ::data ::BOOL, ayoaron ::data ::COLOR, ayoaron ::physic ::CWorld ::donn, ayoaron ::data ::CDataNode ::getbool(), ayoaron ::data ::CDataNode ::getcol(), ayoaron ::data ::CDataNode ::getDataType(), ayoaron ::data ::CDataNode ::getParent(), ayoaron ::data ::CDataNode ::getstr(), ayoaron ::data ::CDataNode ::getu32(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::CWorld ::smgr, ayoa-

ron : :data : :STRING, ayoaron : :data : :UINT, et ayoaron : :data : :VECTOR.

Référencé par CSecteur().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.51.3.10 void ayoaron ::physic ::CSecteur ::setNom (std ::string s) [inline]

défini le nom

Références ayoaron ::physic ::CWorld ::donn, et ayoaron ::data ::CDataNode ::get-Data().

Voici le graphe d'appel pour cette fonction :



7.51.4 Documentation des données membres

7.51.4.1 **data : :CDataNode* ayoaron ::physic ::CWorld ::donn** [protected, inherited]

pointeur vers la structure de donnée

Référencé par CSecteur(), ayoaron ::physic ::CWorld ::CWorld(), ayoaron ::physic ::CWorld ::frame(), frame(), getDataNode(), getNom(), ayoaron ::physic ::CWorld ::loadData(), loadData(), setNom(), et ayoaron ::physic ::CWorld ::~CWorld().

7.51.4.2 **irr : :u32 ayoaron ::physic ::CSecteur ::IDsecteur** [private]

numero du secteur

Référencé par CSecteur().

7.51.4.3 **std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::CWorld ::I_objets** [protected, inherited]

liste des objets enfant

Référencé par CSecteur(), ayoaron ::physic ::CWorld ::frame(), frame(), et ayoaron ::physic ::CWorld ::~CWorld().

7.51.4.4 **CGalaxie* ayoaron ::physic ::CSecteur ::parent** [private]

pointeur vers la galaxie parente

Référencé par CSecteur(), getGfxEngine(), getParent(), et getSfxEngine().

7.51.4.5 **irr ::scene ::ISceneManager* ayoaron ::physic ::CWorld ::smgr** [protected, inherited]

pointeur vers le scene manager

Référencé par ayoaron ::physic ::CWorld ::CWorld(), ayoaron ::physic ::CWorld ::getSceneManager(), et loadData().

7.51.4.6 objets : :CVaisseau* ayoaron : :physic : :CSecteur : :vaisseaujoueur
[private]

vaisseau du joueur

Référencé par CSecteur(), et frame().

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

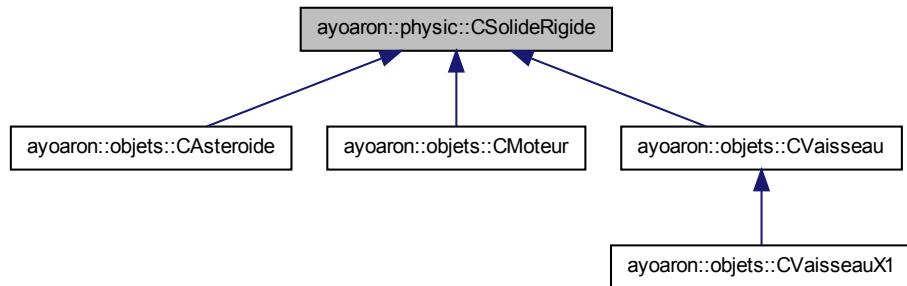
- CSecteur.h
 - CSecteur.cpp

7.52 Référence de la classe ayoaron ::physic ::CSolideRigide

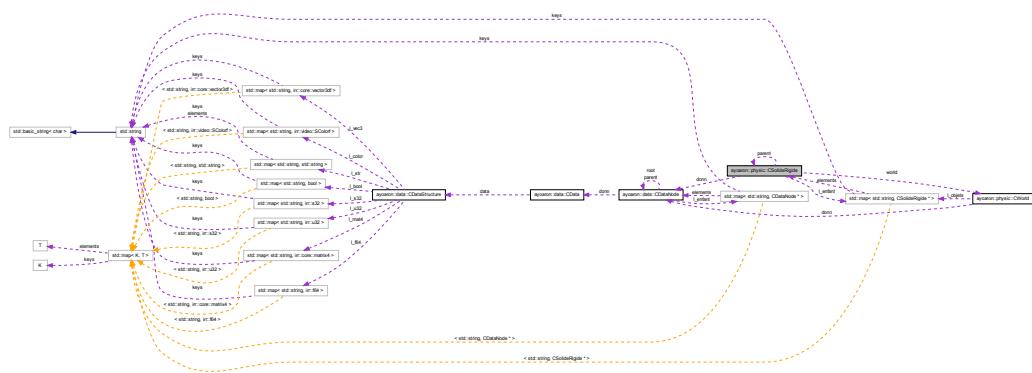
une classe d'objet physique

```
#include <CSolideRigide.h>
```

Graphe d'héritage de ayoaron : :physic : :CSolideRigide :



Graphe de collaboration de ayoaron ::physic ::CSolideRigide ::



Fonctions membres publiques

- void **addAbsoluteEffort** (CEffort F)
ajoute un effort exprimé dans le repère global
- void **addEnfant** (std ::string nom, CSolideRigide *enfant)
ajoute un nouvel enfant (s'il n'existe pas déjà)
- void **addLocalEffort** (CEffort F)
ajoute un effort exprimé dans le repère local
- void **addRelativeEffort** (CEffort F)
ajoute un effort exprimé dans le repère du parent
- CSolideRigide (data ::CDataNode *d, CWorld *worl, CSolideRigide *paren=NULL)

Constructeur.- void **dropEnfant** (std ::string nom)
supprime un enfant de la liste
- void **frame** (double dt)
effectue les diverses opération de mise à jour
- irr ::core ::vector3df **getAbsoluteForceResultante** ()
renvoie la force résultante exprimée dans le repère global
- irr ::core ::vector3df **getAbsoluteMomentResultante** ()
renvoie le moment résultant exprimé dans le repère global
- irr ::core ::vector3df **getAbsolutePosition** ()
renvoie la position de l'objet dans le monde
- irr ::core ::vector3df **getAbsoluteRitesse** ()
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- irr ::core ::matrix4 **getAbsoluteRotation** ()
renvoie la matrice de rotation entre le repère global et le repère local
- irr ::core ::vector3df **getAbsoluteVitesse** ()
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)
- data ::CDataNode * **getDonn** ()
accès aux données
- CSolideRigide * **getEnfant** (std ::string nom)
renvoie l'enfant donné
- irr ::core ::matrix4 **getGlobalInertie** ()
renvoie la matrice d'inertie
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantBegin** ()
renvoie un itérateur sur les enfant pointant au début de la map
- std ::map< std ::string, CSolideRigide * > ::iterator **getIteratorEnfantEnd** ()
renvoie un itérateur sur les enfant pointant à la fin de la map
- irr ::core ::vector3df **getLocalForceResultante** ()
renvoie la force résultante exprimée dans le repère local
- irr ::core ::matrix4 **getLocalInertie** ()
renvoie la matrice d'inertie
- irr ::core ::vector3df **getLocalMomentResultante** ()
renvoie le moment résultant exprimé dans le repère local
- irr ::core ::vector3df **getLocalRitesse** ()
renvoie l'expression de la vitesse de rotation dans le repère local
- irr ::core ::vector3df **getLocalVitesse** ()
renvoie l'expression de la vitesse de translation dans le repère local
- irr ::f64 **getMasse** ()
fonction qui renvoie la masse du solide
- irr ::u32 **getNbEnfant** ()
renvoie le nombre d'enfants de cet objet
- irr ::scene ::ISceneNode * **getNode** ()
renvoie le scene node et le maillage associé
- irr ::core ::vector3df **getRelativeForceResultante** ()
renvoie la force résultante exprimée dans le repère du parent

- irr ::core ::matrix4 **getRelativeInertie** ()

renvoie la matrice d'inertie
- irr ::core ::vector3df **getRelativeMomentResultante** ()

renvoie le moment résultant exprimé dans le repère du parent
- irr ::core ::vector3df **getRelativePosition** ()

renvoie la position de l'objet par rapport à l'objet parent
- irr ::core ::vector3df **getRelativeRitesse** ()

renvoie l'expression de la vitesse de rotation dans le repère du parent
- irr ::core ::matrix4 **getRelativeRotation** ()

renvoie la matrice de rotation entre le repère relatif et le repère local
- irr ::core ::vector3df **getRelativeVitesse** ()

renvoie l'expression de la vitesse de translation dans le repère du parent
- virtual void **loadObjetData** (data ::CDataNode *d=NULL)

permet un chargement personnalisé
- virtual void **postPhysique** (double dt)

effectue les diverses opération de mise à jour après le calcul de la physique
- virtual void **prePhysique** (double dt)

effectue les diverses opération de mise à jour avant de calculer la physique
- void **resetEffort** ()

remet les efforts à zéro
- void **setAbsoluteRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide (tient compte de la vitesse d'entraînement du repère parent)
- void **setAbsoluteVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entraînement du repère parent)
- void **setLocalRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide
- void **setLocalVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide
- void **setParent** (CSolideRigide *par)

change le parent de cet objet
- void **setRelativePosition** (irr ::core ::vector3df pos)

définit la position de l'objet par rapport à son parent
- void **setRelativeRitesse** (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide
- void **setRelativeRotation** (irr ::core ::vector3df rot)

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.
- void **setRelativeVitesse** (irr ::core ::vector3df vit)

définit la vitesse du solide
- virtual **~CSolideRigide** ()

Destructeur.

Fonctions membres protégées

- irr ::core ::matrix4 **getMatrix3** (std ::string nom, REPERE ou, REPERE reperestockage)

renvoie le vecteur donné dans la base voulue
- irr ::core ::vector3df **getVector** (std ::string nom, REPERE ou, REPERE reperestockage)

renvoie le vecteur donné dans la base voulue
- void **setMatrix3** (std ::string nom, REPERE source, irr ::core ::matrix4 mat, REPERE reperestockage)

fonction permettant le changement de repère des matrices
- void **setVector** (std ::string nom, REPERE source, irr ::core ::vector3df vec, REPERE reperestockage)

fonction permettant le changement de repère des vecteurs

Attributs protégés

- irr : :scene : :ICameraSceneNode * **cam**
pointeur vers le node de camera
- irr : :core : :vector3df **caml**
direction du regard de la caméra
- irr : :core : :vector3df **camp**
position de la caméra dans le repère local
- irr : :core : :vector3df **camup**
vecteur de la verticale de la caméra
- **data** : :CDataNode * **donn**
pointeur vers la structure de donnée
- irr : :core : :vector3df **forceResultante**
le vecteur de la résultante des efforts, il est exprimé dans le repère local
- bool **hascamera**
interrupteur de caméra
- std : :map< std : :string, CSolideRigide * > **l_enfant**
liste des objets enfants
- irr : :core : :vector3df **momentResultant**
le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)
- irr : :scene : :ISceneNode * **node**
pointeur vers le maillage et le scene node
- CSolideRigide * **parent**
pointeur vers l'objet parent
- CWorld * **world**
pointeur vers le monde

7.52.1 Description détaillée

une classe d'objet physique

La structure de donnée contient au minimum (créés à la construction si inexistant) : la masse de l'objet défaut : 1 nom :"masse" la matrice de l'opérateur d'inertie dans le repère local défaut : identité*masse nom :"inertie" le vecteur vitesse de translation (repère du parent) défaut : (0,0,0) nom :"vitesse" le vecteur vitesse de rotation (repère du parent) défaut : (0,0,0) nom :"ritesse"

Dans ce contexte, trois repère existent :

- le repère local : c'est le repère lié au solide, il existe toujours
- le repère relatif : c'est le repère lié à l'objet parent (identique au repère global s'il n'y a pas de parent)
- le repère global : c'est le repère lié au monde, supposé fixe et galiléen. L'utilisation de ces termes dans les fonction permettent d'identifier dans quel repère sont les données

7.52.2 Documentation des constructeurs et destructeur

7.52.2.1 ayoaron ::physic ::CSolideRigide (**data** : :CDataNode * *d*, CWorld * *wor*, CSolideRigide * *paren* = NULL)

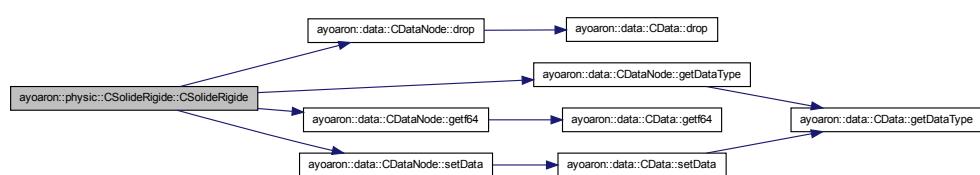
Constructeur.

Paramètres

in	<i>d</i>	pointeur vers la structure de donnée associée à l'objet (obligatoire)
in	<i>worl</i>	pointeur vers le monde (obligatoire)
in	<i>paren</i>	pointeur vers l'objet parent si NULL, le parent c'est le monde lui-même

Références cam, caml, camp, camup, donn, ayoaron ::data ::CDataNode ::drop(), ayoaron ::data ::FLOAT, ayoaron ::data ::CDataNode ::getDataType(), ayoaron ::data ::CDataNode ::getf64(), hascamera, ayoaron ::data ::MATRIX, parent, ayoaron ::data ::CDataNode ::setData(), ayoaron ::data ::VECTOR, et world.

Voici le graphe d'appel pour cette fonction :

**7.52.2.2 ayoaron ::physic ::CSolideRigide ::~CSolideRigide() [virtual]**

Destructeur.

7.52.3 Documentation des fonctions membres**7.52.3.1 void ayoaron ::physic ::CSolideRigide ::addAbsoluteEffort (CEffort *F*)**

ajoute un effort exprimé dans le repère global

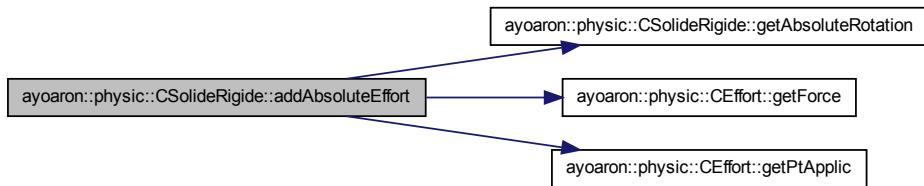
Paramètres

in	<i>F</i>	l'effort à ajouterexprimé dans le repère global, le point d'application est relatif au centre de gravité de l'objet (le O du repère local)
----	----------	--

Références forceResultante, getAbsoluteRotation(), ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), et momentResultant.

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.2 void ayoaron ::physic ::CSolideRigide ::addEnfant (std ::string nom, CSolideRigide * enfant)

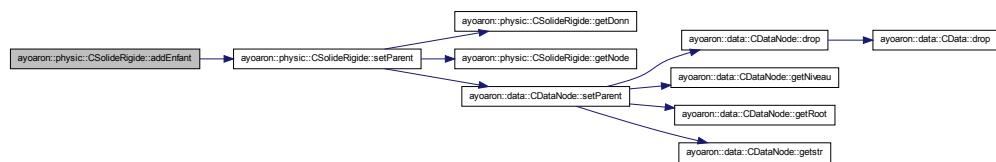
ajoute un nouvel enfant (s'il n'existe pas déjà)

Paramètres

<i>nom</i>	le nom (unique) de l'enfant
<i>enfant</i>	pointeur vers l'enfant en question

Références l_enfant, et setParent().

Voici le graphe d'appel pour cette fonction :



7.52.3.3 void ayoaron ::physic ::CSolideRigide ::addLocalEffort (CEffort F)

ajoute un effort exprimé dans le repère local

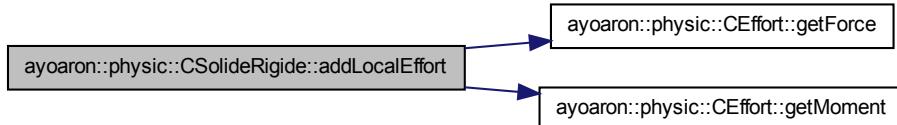
Paramètres

in	F	l'effort à ajouterexprimé dans le repère local, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	-----	--

Références forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getMoment(), et momentResultant.

Référencé par ayoaron ::objets ::CMoteur ::prePhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

7.52.3.4 void ayoaron ::physic ::CSolideRigide ::addRelativeEffort (CEffort F)

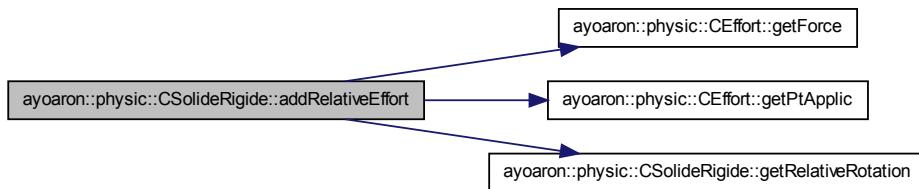
ajoute un effort exprimé dans le repère du parent

Paramètres

in	F	l'effort à ajouterexprimé dans le repère du parent, le point d'application est relatif au chentre de gravité de l'objet (le O du repère local)
----	-----	--

Références forceResultante, ayoaron ::physic ::CEffort ::getForce(), ayoaron ::physic ::CEffort ::getPtApplic(), getRelativeRotation(), et momentResultant.

Voici le graphe d'appel pour cette fonction :



7.52.3.5 void ayoaron ::physic ::CSolideRigide ::dropEnfant (std ::string nom)

supprime un enfant de la liste

Paramètres

nom	nom de l'enfant à supprimer
-----	-----------------------------

7.52.3.6 void ayoaron ::physic ::CSolideRigide ::frame (double dt)

effectue les diverses opération de mise à jour

Paramètres

in	dt	le pas de temps
----	----	-----------------

la frame :

tout d'abord, l'effort global

on pre frame tous les objets enfants

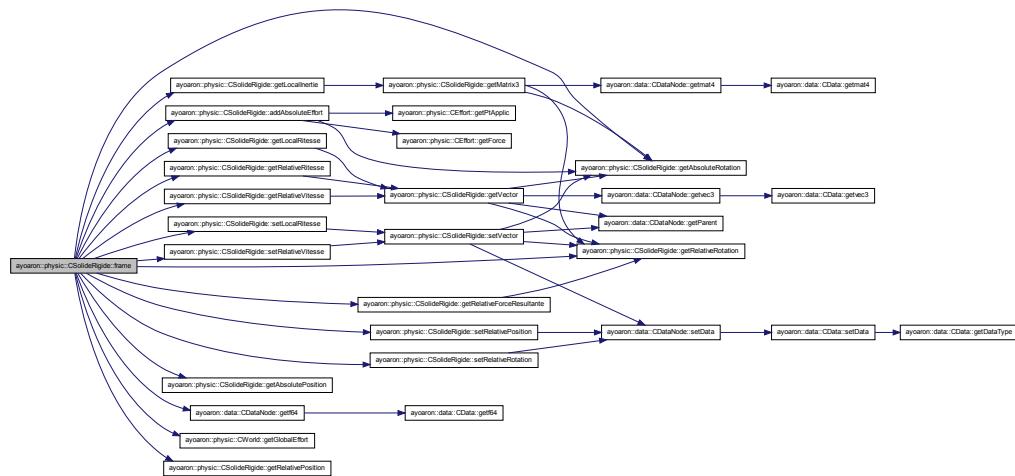
la translation, calculée dans le repère du parent

la rotation, calculée dans le repère parent

on post frame tous les objets enfants

Références addAbsoluteEffort(), cam, caml, camp, camup, donn, forceResultante, getAbsolutePosition(), getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getf64(), ayoaron ::physic ::CWorld ::getGlobalEffort(), getLocalInertie(), getLocalRitesse(), getRelativeForceResultante(), getRelativePosition(), getRelativeRitesse(), getRelativeRotation(), getRelativeVitesse(), hascamera, l_enfant, momentResultant, node, setLocalRitesse(), setRelativePosition(), setRelativeRotation(), setRelativeVitesse(), et world.

Voici le graphe d'appel pour cette fonction :



7.52.3.7 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteForceResultante ()

renvoie la force résultante exprimée dans le repère global

Renvoie

la résultante des efforts exprimée dans le référentiel global

Références forceResultante, et getAbsoluteRotation().

Voici le graphe d'appel pour cette fonction :



7.52.3.8 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteMomentResultante ()

renvoie le moment résultant exprimé dans le repère global

Renvoie

le moment résultant exprimé dans le référentiel global

Références getAbsoluteRotation(), et momentResultant.

Voici le graphe d'appel pour cette fonction :



7.52.3.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsolutePosition ()

renvoie la position de l'objet dans le monde

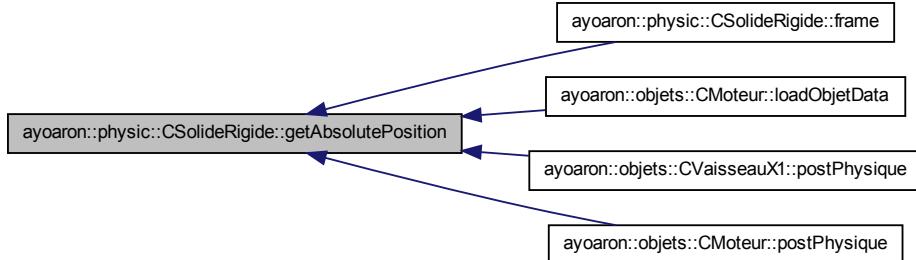
Renvoie

le vecteur position absolue de l'objet

Références node.

Référencé par frame(), ayoaron ::objets ::CMoteur ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), et ayoaron ::objets ::CMoteur ::postPhysique().

Voici le graphe d'appel pour cette fonction :



7.52.3.10 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteRitesse ()

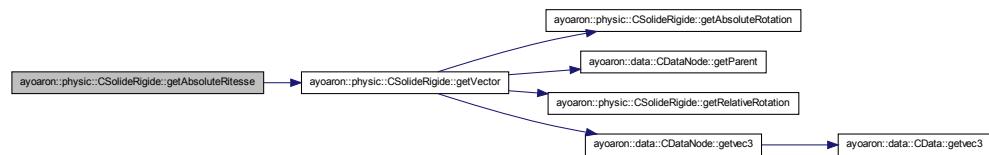
renvoie l'expression de la vitesse de rotation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse de rotation dans le repère global

Références ayoaron ::physic ::GENERAL, getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.52.3.11 irr::core::matrix4 ayoaron::physic::CSolideRigide::getAbsoluteRotation()

renvoie la matrice de rotation entre le repère global et le repère local

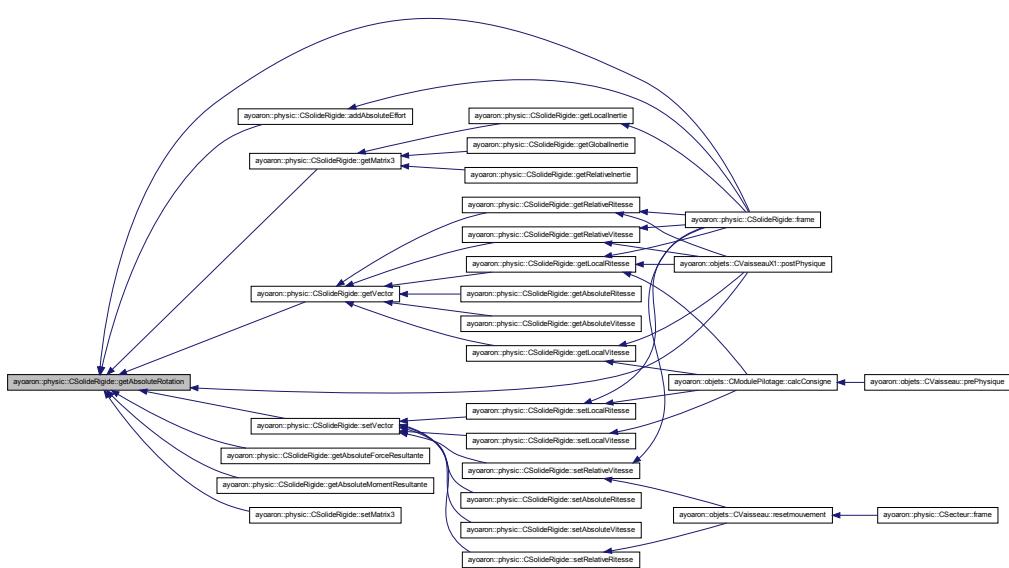
Renvoie

la matrice de rotation du repère global au repère local

Références node.

Référencé par addAbsoluteEffort(), frame(), getAbsoluteForceResultante(), getAbsoluteMomentResultante(), getMatrix3(), getVector(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), setMatrix3(), et setVector().

Voici le graphe d'appel pour cette fonction :



7.52.3.12 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getAbsoluteVitesse ()

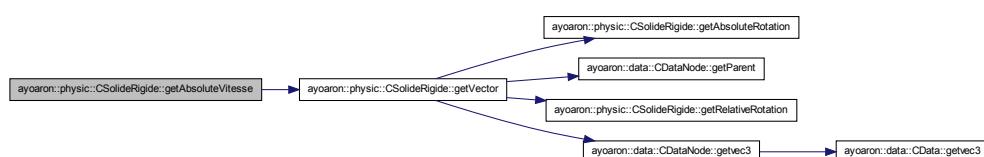
renvoie l'expression de la vitesse de translation dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Renvoie

le vecteur vitesse dans le repère global

Références ayoaron ::physic ::GENERAL, getVector(), et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :

7.52.3.13 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::getDonn ()
[inline]

acces aux données

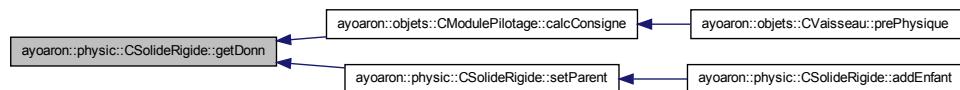
Renvoie

pointeur vers la structure de donnée

Références donn.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et setParent().

Voici le graphe d'appel pour cette fonction :



7.52.3.14 CSolideRigide * ayoaron ::physic ::CSolideRigide ::getEnfant (std ::string nom)

renvoie l'enfant donné

Paramètres

nom	le nom de l'enfant voulu
-----	--------------------------

Renvoie

pointeur vers l'enfant désiré, renvoie NULL si l'enfant n'existe pas

Références l_enfant.

7.52.3.15 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getGlobalInertie ()
[inline]

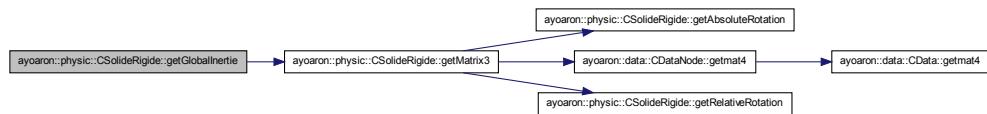
renvoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère global

Références ayoaron ::physic ::GENERAL, getMatrix3(), et ayoaron ::physic ::LOCAL.

Voici le graphe d'appel pour cette fonction :



7.52.3.16 std ::map< std ::string, CSolideRigide * > ::iterator
ayoaron ::physic ::CSolideRigide ::getIteratorEnfantBegin ()

renvoie un itérator sur les enfant pointant au début de la map

Renvoie

iterator pointant au début de la liste des enfant

Références l_enfant.

7.52.3.17 std ::map< std ::string, CSolideRigide * > ::iterator
ayoaron ::physic ::CSolideRigide ::getIteratorEnfantEnd ()

renvoie un itérator sur les enfant pointant à la fin de la map

Renvoie

iterator pointant à la fin de la liste des enfant

Références l_enfant.

7.52.3.18 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalForceResultante ()

renvoie la force résultante exprimée dans le repère local

Renvoie

la résultante des efforts exprimée dans le référentiel local

Références forceResultante.

7.52.3.19 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getLocallnertie ()
[inline]

revoie la matrice d'inertie

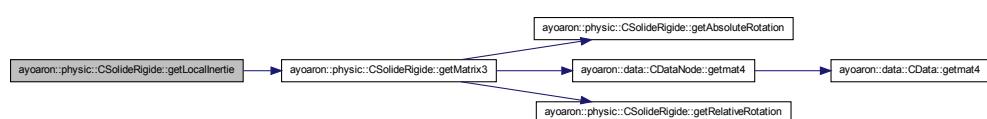
Renvoie

la matrice d'inertie dans le repère local

Références getMatrix3(), et ayoaron ::physic ::LOCAL.

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.20 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalMomentResultante ()

renvoie le moment résultant exprimé dans le repère local

Renvoie

le moment résultant exprimé dans le référentiel local

Références momentResultant.

7.52.3.21 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalRitesse ()

renvoie l'expression de la vitesse de rotation dans le repère local

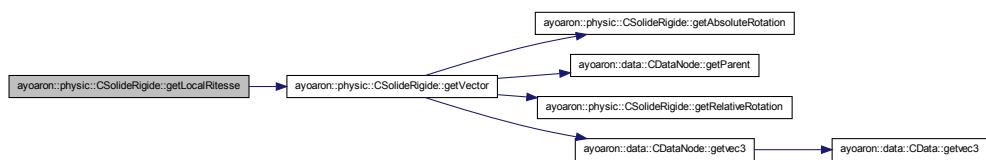
Renvoie

le vecteur vitesse de rotation dans le repère local

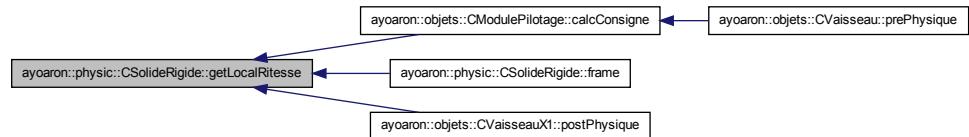
Références getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

**7.52.3.22 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getLocalVitesse ()**

renvoie l'expression de la vitesse de translation dans le repère local

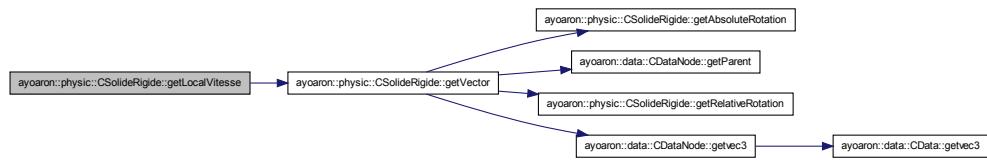
Renvoie

le vecteur vitesse dans le repère local

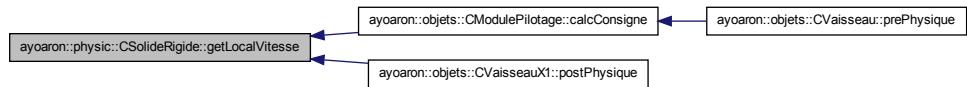
Références getVector(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.23 irr ::f64 ayoaron ::physic ::CSolideRigide ::getMasse () [inline]

fonction qui renvoie la masse du solide

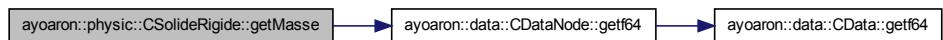
Renvoie

la masse du solide

Références donn, et ayoaron ::data ::CDataNode ::getf64().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.24 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getMatrix3 (std ::string *nom*,
REPERE *ou*, **REPERE** *reperestockage*) [protected]

renvoie le vecteur donné dans la base voulue

Paramètres

<i>nom</i>	nom de la clé dans la base de données
<i>ou</i>	le repère dans lequel on veut le vecteur
<i>reperestockage</i>	le repère dans lequel est stocker le vecteur

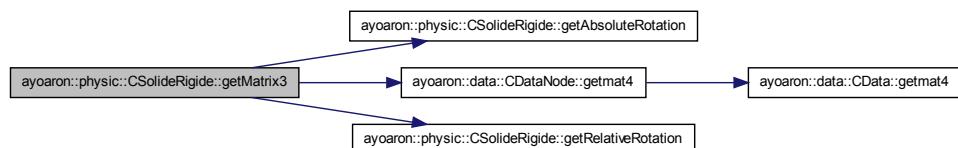
Renvoie

le vecteur voulu dans la base voulue

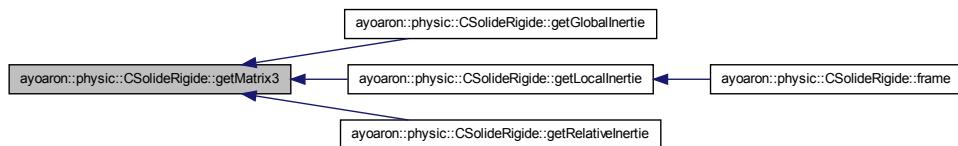
Références donn, ayoaron ::physic ::GENERAL, getAbsoluteRotation(), ayoaron ::data ::CDa-
taNode ::getmat4(), getRelativeRotation(), ayoaron ::physic ::LOCAL, et ayoaron ::phy-
sic ::PARENT.

Référencé par getGlobalInertie(), getLocalInertie(), et getRelativeInertie().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.25 irr ::u32 ayoaron ::physic ::CSolideRigide ::getNbEnfant () [inline]

renvoie le nombre d'enfants de cet objet

Renvoie

le nombre d'enfant

Références l_enfant.

7.52.3.26 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::getNode ()
[inline]

renvoie le scene node et le maillage associé

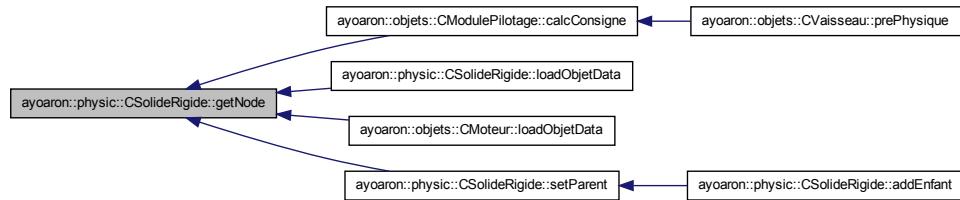
Renvoie

pointeur vers le IAnimatedMeshSceneNode

Références node.

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), loadObjetData(),
ayoaron ::objets ::CMoteur ::loadObjetData(), et setParent().

Voici le graphe d'appel pour cette fonction :



7.52.3.27 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeForceResultante ()

renvoie la force résultante exprimée dans le repère du parent

Renvoie

la résultante des efforts exprimée dans le référentiel du parent

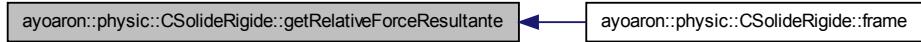
Références forceResultante, et getRelativeRotation().

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.28 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeInertie ()
 [inline]

revoie la matrice d'inertie

Renvoie

la matrice d'inertie dans le repère du parent

Références getMatrix3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Voici le graphe d'appel pour cette fonction :



7.52.3.29 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeMomentResultante ()

renvoie le moment résultant exprimé dans le repère du parent

Renvoie

le moment résultant exprimé dans le référentiel du parent

Références getRelativeRotation(), et momentResultant.

Voici le graphe d'appel pour cette fonction :



7.52.3.30 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativePosition ()

renvoie la position de l'objet par rapport à l'objet parent

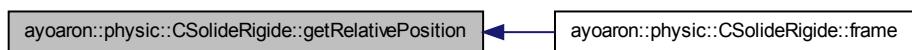
Renvoie

le vecteur position relative au parent

Références node.

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



7.52.3.31 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeRitesse ()

renvoie l'expression de la vitesse de rotation dans le repère du parent

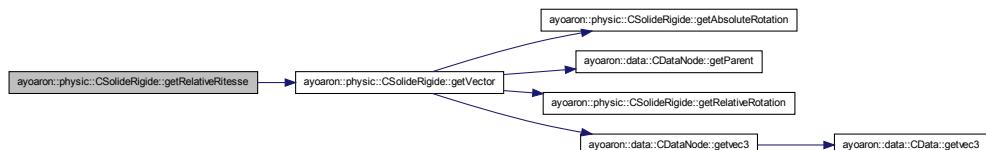
Renvoie

le vecteur vitesse de rotation dans le repère du parent

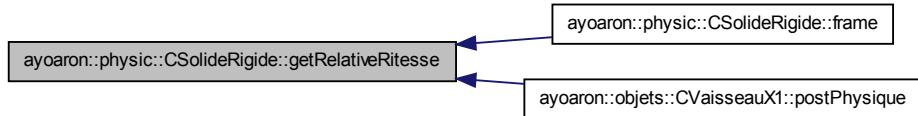
Références getVector(), et ayoaron ::physic ::PARENT.

Référencé par frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.32 irr ::core ::matrix4 ayoaron ::physic ::CSolideRigide ::getRelativeRotation ()

renvoie la matrice de rotation entre le repère relatif et le repère local

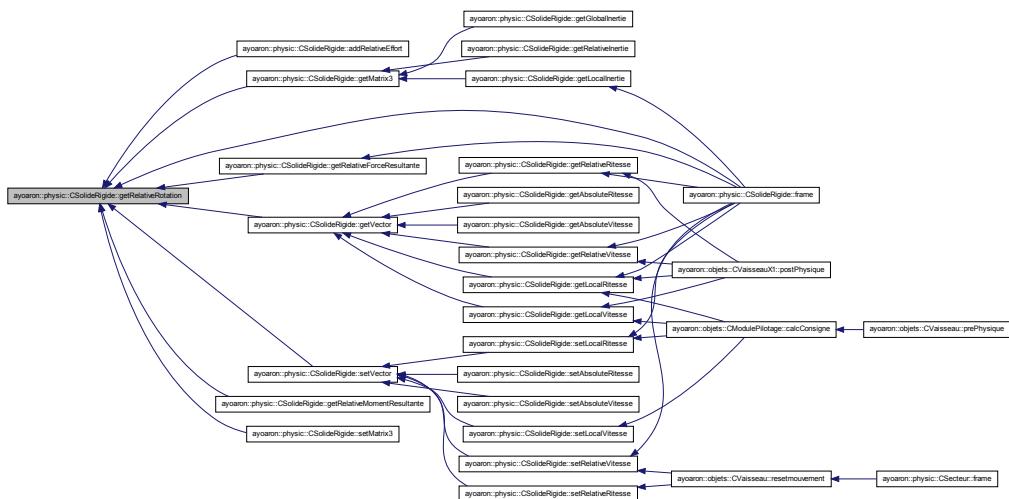
Renvoie

la matrice de rotation du repère relatif au repère local

Références node.

Référencé par addRelativeEffort(), frame(), getMatrix3(), getRelativeForceResultante(), getRelativeMomentResultante(), getVector(), setMatrix3(), et setVector().

Voici le graphe d'appel pour cette fonction :



7.52.3.33 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getRelativeVitesse ()

renvoie l'expression de la vitesse de translation dans le repère du parent

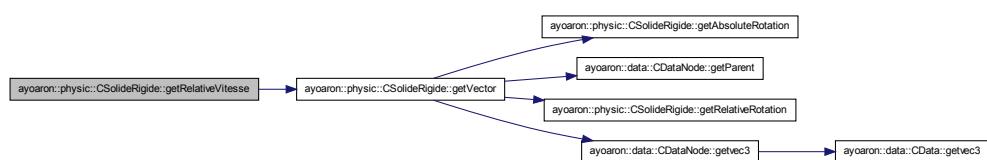
Renvoie

le vecteur vitesse dans le repère du parent

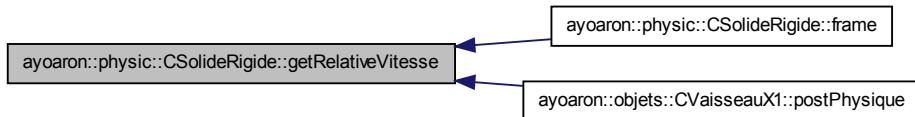
Références getVector(), et ayoaron ::physic ::PARENT.

Référencé par frame(), et ayoaron ::objets ::CVaisseauX1 ::postPhysique().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.34 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::getVector (std ::string nom, REPERE ou, REPERE reperestockage) [protected]

renvoie le vecteur donné dans la base voulue

Paramètres

nom	nom de la clé dans la base de données
ou	le repère dans lequel on veut le vecteur
reperestockage	le repère dans lequel est stocker le vecteur

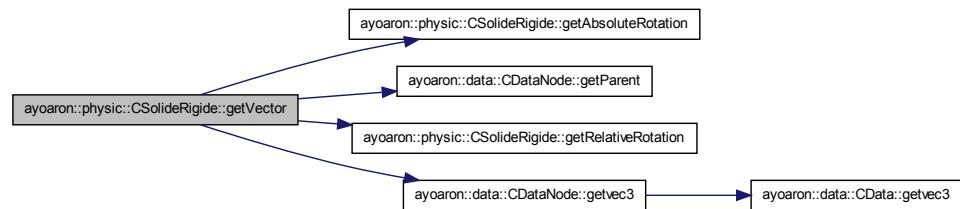
Renvoie

le vecteur voulu dans la base voulue

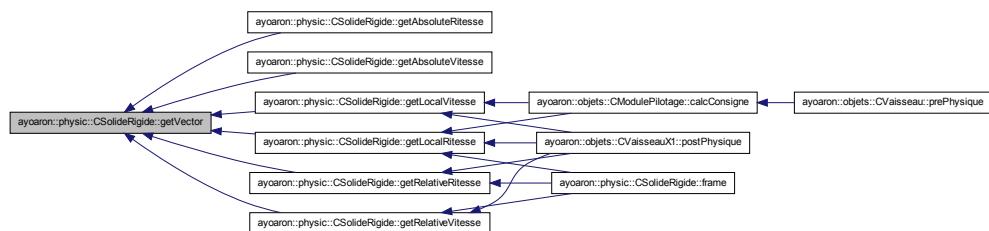
Références nom, ayoaron ::physic ::GENERAL, getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), getRelativeRotation(), ayoaron ::data ::CDataNode ::getvec3(), ayoaron ::physic ::LOCAL, et ayoaron ::physic ::PARENT.

Référencé par `getAbsoluteRitesse()`, `getAbsoluteVitesse()`, `getLocalRitesse()`, `getLocalVitesse()`, `getRelativeRitesse()`, et `getRelativeVitesse()`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.35 `void ayoaron ::physic ::CSolideRigide ::loadObjetData (data ::CDataNode * d = NULL) [virtual]`

permet un chargement personnalisé

Paramètres

`d` les données à charger

À faire

rendre disponible plus de paramètres de texture

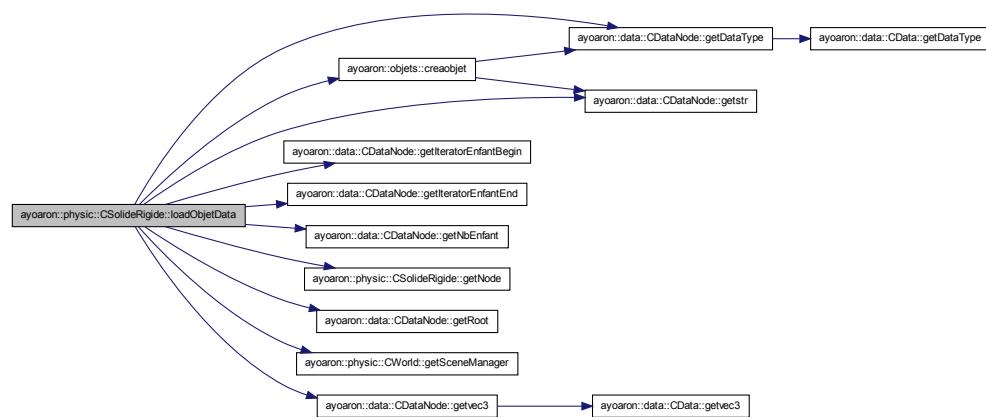
À faire

faire le lien avec les options globales de traitement

Réimplémentée dans `ayoaron ::objets ::CAsteriode`, `ayoaron ::objets ::CMoteur`, `ayoaron ::objets ::CVaisseau`, et `ayoaron ::objets ::CVaisseauX1`.

Références ayoaron ::objets ::creaobjet(), donn, ayoaron ::data ::CDataNode ::getDataType(), ayoaron ::data ::CDataNode ::getIteratorEnfantBegin(), ayoaron ::data ::CDataNode ::getIteratorEnfantEnd(), ayoaron ::data ::CDataNode ::getNbEnfant(), getNode(), ayoaron ::data ::CDataNode ::getRoot(), ayoaron ::physic ::CWorld ::getSceneManager(), ayoaron ::data ::CDataNode ::getstr(), ayoaron ::data ::CDataNode ::getvec3(), l_enfant, node, parent, ayoaron ::data ::STRING, et world.

Voici le graphe d'appel pour cette fonction :



**7.52.3.36 virtual void ayoaron ::physic ::CSolideRigide ::postPhysique (double dt)
[inline, virtual]**

effectue les diverses opération de mise à jour après le calcul de la physique

Paramètres

in	dt	le pas de temps
----	----	-----------------

Réimplémentée dans ayoaron ::objets ::CMoteur, et ayoaron ::objets ::CVaisseauX1.

**7.52.3.37 virtual void ayoaron ::physic ::CSolideRigide ::prePhysique (double dt)
[inline, virtual]**

effectue les diverses opération de mise à jour avant de calculer la physique

Paramètres

in	dt	le pas de temps
----	----	-----------------

Réimplémentée dans ayoaron ::objets ::CMoteur, et ayoaron ::objets ::CVaisseau.

Références forceResultante, et momentResultant.

7.52.3.38 void ayoaron ::physic ::CSolideRigide ::resetEffort (void)

remet les efforts à zéro

Références forceResultante, et momentResultant.

7.52.3.39 void ayoaron ::physic ::CSolideRigide ::setAbsoluteRitesse (irr ::core ::vector3df vit)

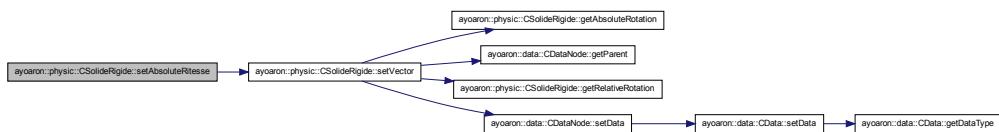
définit la vitesse de rotation du solide (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et setVector().

Voici le graphe d'appel pour cette fonction :



7.52.3.40 void ayoaron ::physic ::CSolideRigide ::setAbsoluteVitesse (irr ::core ::vector3df vit)

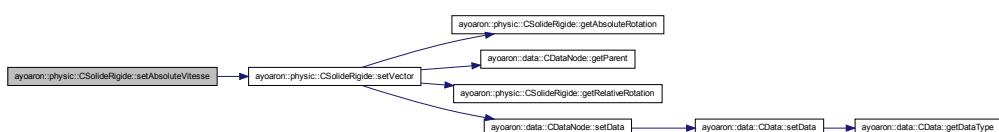
définit la vitesse du solide dans le repère global (tient compte de la vitesse d'entrainement du repère parent)

Paramètres

in	vit	le vecteur vitesse exprimé dans le repère global
----	-----	--

Références ayoaron ::physic ::GENERAL, ayoaron ::physic ::PARENT, et setVector().

Voici le graphe d'appel pour cette fonction :



7.52.3.41 void ayoaron ::physic ::CSolideRigide ::setLocalRitesse (irr ::core ::vector3df *vit*)

définit la vitesse de rotation du solide

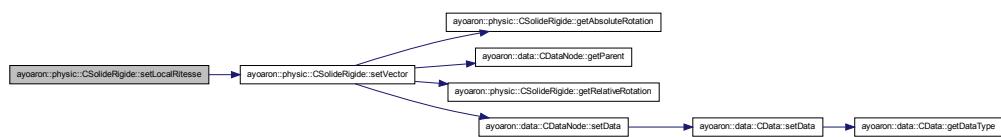
Paramètres

in	<i>vit</i>	le vecteur vitesse de rotation exprimé dans le repère local
----	------------	---

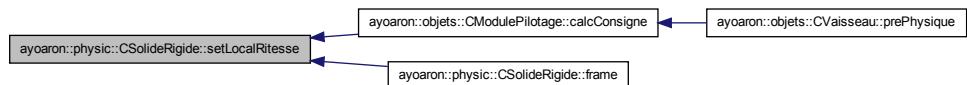
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne(), et frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.42 void ayoaron ::physic ::CSolideRigide ::setLocalVitesse (irr ::core ::vector3df *vit*)

définit la vitesse du solide

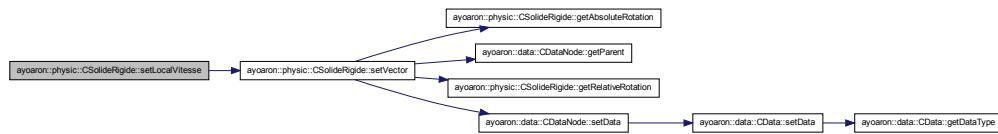
Paramètres

in	<i>vit</i>	le vecteur vitesse exprimé dans le repère local
----	------------	---

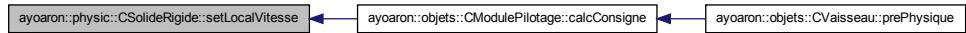
Références ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et setVector().

Référencé par ayoaron ::objets ::CModulePilotage ::calcConsigne().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.43 void ayoaron ::physic ::CSolideRigide ::setMatrix3 (std ::string *nom*, REPERE *source*, irr ::core ::matrix4 *mat*, REPERE *reperstockage*) [protected]

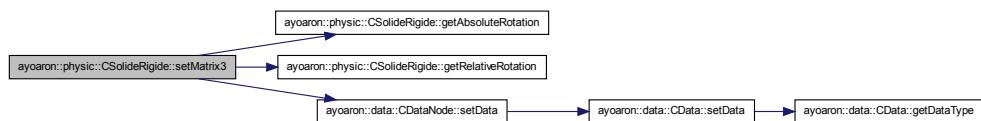
fonction permettant le changement de repère des matrices

Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>mat</i>	la matrice à stocker
<i>reperstockage</i>	le repère de stockage du vecteur

Références donn, ayoaron ::physic ::GENERAL, getAbsoluteRotation(), getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.52.3.44 void ayoaron ::physic ::CSolideRigide ::setParent (CSolideRigide * *par*)

change le parent de cet objet

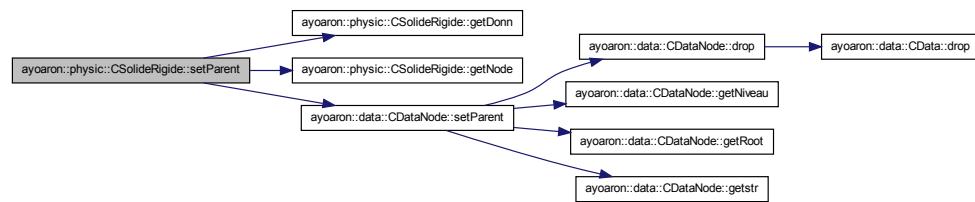
Paramètres

in	par	le nouveau parent
----	-----	-------------------

Références donn, getDonn(), getNode(), node, parent, et ayoaron ::data ::CDataNode ::setParent().

Référencé par addEnfant().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.45 void ayoaron ::physic ::CSolideRigide ::setRelativePosition (irr ::core ::vector3df pos)

définit la position de l'objet par rapport à son parent

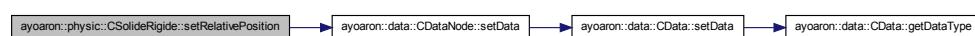
Paramètres

pos	la position relative de l'objet
-----	---------------------------------

Références donn, node, et ayoaron ::data ::CDataNode ::setData().

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.46 void ayoaron ::physic ::CSolideRigide ::setRelativeRitesse (irr ::core ::vector3df vit)

définit la vitesse de rotation du solide

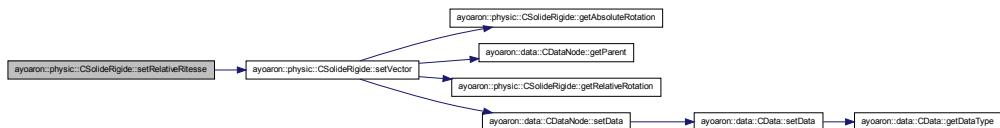
Paramètres

in	vit	le vecteur vitesse de rotation exprimé dans le repère du parent
----	-----	---

Références ayoaron ::physic ::PARENT, et setVector().

Référencé par ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.47 void ayoaron ::physic ::CSolideRigide ::setRelativeRotation (irr ::core ::vector3df rot)

à l'aide d'un vecteur de rotation on définit la rotation entre le repère relatif et le repère local.

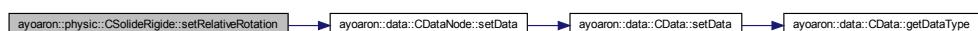
Paramètres

rot	le vecteur rotation
-----	---------------------

Références donn, node, et ayoaron ::data ::CDataNode ::setData().

Référencé par frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.48 void ayoaron ::physic ::CSolideRigide ::setRelativeVitesse (irr ::core ::vector3df vit)

définit la vitesse du solide

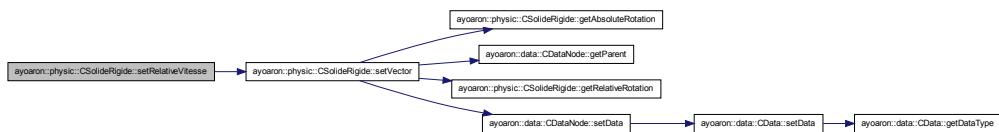
Paramètres

in	vit	le vecteur vitesse exprimé dans le repère du parent
----	-----	---

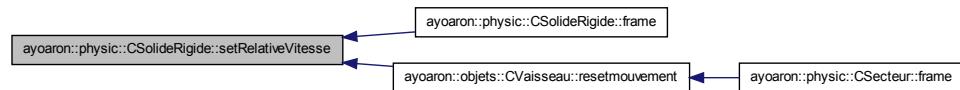
Références ayoaron ::physic ::PARENT, et setVector().

Référencé par frame(), et ayoaron ::objets ::CVaisseau ::resetmouvement().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.3.49 void ayoaron ::physic ::CSolideRigide ::setVector (std ::string *nom*, REPERE *source*, irr ::core ::vector3df *vec*, REPERE *reperestockage*) [protected]

fonction permettant le changement de repère des vecteurs

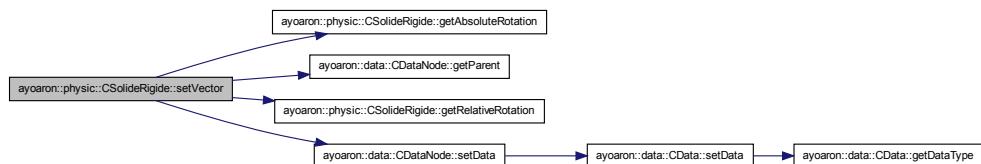
Paramètres

<i>nom</i>	nom de la clé vecteur dans la base de données
<i>source</i>	le repère dans lequel est donné le vecteur
<i>vec</i>	le vecteur à stacker
<i>reperestockage</i>	le repère de stockage du vecteur

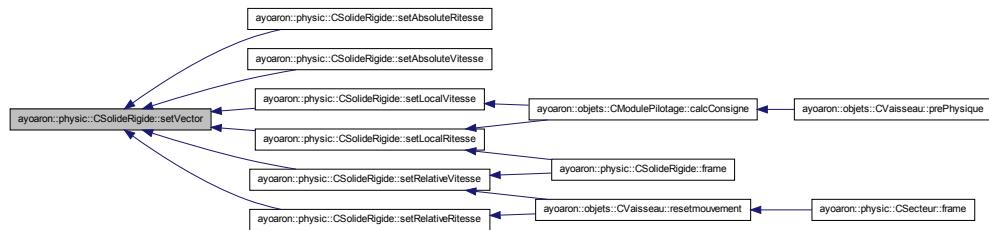
Références donn, ayoaron ::physic ::GENERAL, getAbsoluteRotation(), ayoaron ::data ::CDataNode ::getParent(), getRelativeRotation(), ayoaron ::physic ::LOCAL, ayoaron ::physic ::PARENT, et ayoaron ::data ::CDataNode ::setData().

Référencé par setAbsoluteRitesse(), setAbsoluteVitesse(), setLocalRitesse(), setLocalVitesse(), setRelativeRitesse(), et setRelativeVitesse().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.52.4 Documentation des données membres

7.52.4.1 irr ::scene ::ICameraSceneNode* ayoaron ::physic ::CSolideRigide ::cam
[protected]

pointeur vers le node de camera

Référencé par CSolideRigide(), frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjectData().

7.52.4.2 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::caml
[protected]

direction du regard de la caméra

Référencé par CSolideRigide(), frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjectData().

7.52.4.3 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camp
[protected]

position de la caméra dans le repère local

Référencé par CSolideRigide(), frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjectData().

7.52.4.4 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::camup
[protected]

vecteur de la verticale de la caméra

Référencé par CSolideRigide(), frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjectData().

7.52.4.5 data ::CDataNode* ayoaron ::physic ::CSolideRigide ::donn
[protected]

pointeur vers la structure de donnée

Référencé par CSolideRigide(), frame(), getDonn(), ayoaron ::objets ::CMoteur ::getForce(), getMasse(), getMatrix3(), ayoaron ::objets ::CMoteur ::getPousse(), ayoaron ::objets ::CMoteur ::getPousseMax(), ayoaron ::objets ::CMoteur ::getRegime(), getVector(), ayoaron ::objets ::CMoteur ::isReversible(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), loadObjetData(), ayoaron ::objets ::CMoteur ::loadObjetData(), ayoaron ::objets ::CAsteroid ::loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), ayoaron ::objets ::CMoteur ::postPhysique(), setMatrix3(), setParent(), ayoaron ::objets ::CMoteur ::setRegime(), setRelativePosition(), setRelativeRotation(), et setVector().

7.52.4.6 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::forceResultante [protected]

le vecteur de la résultante des efforts, il est exprimé dans le repère local

Référencé par addAbsoluteEffort(), addLocalEffort(), addRelativeEffort(), frame(), getAbsoluteForceResultante(), getLocalForceResultante(), getRelativeForceResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), prePhysique(), et resetEffort().

7.52.4.7 bool ayoaron ::physic ::CSolideRigide ::hascamera [protected]

interrupteur de caméra

Référencé par CSolideRigide(), frame(), et ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

7.52.4.8 std ::map<std ::string,CSolideRigide*> ayoaron ::physic ::CSolideRigide :::_enfant [protected]

liste des objets enfants

Référencé par addEnfant(), frame(), getEnfant(), getIteratorEnfantBegin(), getIteratorEnfantEnd(), getNbEnfant(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), et loadObjetData().

7.52.4.9 irr ::core ::vector3df ayoaron ::physic ::CSolideRigide ::momentResultant [protected]

le vecteur du moment résultant en G, il est exprimé dans le repère local, dans lequel G=(0,0,0)

Référencé par addAbsoluteEffort(), addLocalEffort(), addRelativeEffort(), frame(), getAbsoluteMomentResultante(), getLocalMomentResultante(), getRelativeMomentResultante(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), prePhysique(), et resetEffort().

7.52.4.10 irr ::scene ::ISceneNode* ayoaron ::physic ::CSolideRigide ::node [protected]

pointeur vers le maillage et le scene node

Référencé par frame(), getAbsolutePosition(), getAbsoluteRotation(), getNode(), getRelativePosition(), getRelativeRotation(), ayoaron ::objets ::CVaisseauX1 ::loadObjetData().

Data(), loadObjetData(), ayoaron ::objets ::CVaisseauX1 ::postPhysique(), setParent(), setRelativePosition(), et setRelativeRotation().

7.52.4.11 CSolideRigide* ayoaron ::physic ::CSolideRigide ::parent [protected]

pointeur vers l'objet parent

Réimplémentée dans ayoaron ::objets ::CMoteur.

Référencé par CSolideRigide(), ayoaron ::objets ::CVaisseau ::CVaisseau(), ayoaron ::objets ::CVaisseauX1 ::CVaisseauX1(), loadObjetData(), et setParent().

7.52.4.12 CWorld* ayoaron ::physic ::CSolideRigide ::world [protected]

pointeur vers le monde

Réimplémentée dans ayoaron ::objets ::CAsteroide, ayoaron ::objets ::CMoteur, et ayoaron ::objets ::CVaisseau.

Référencé par CSolideRigide(), frame(), et loadObjetData().

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

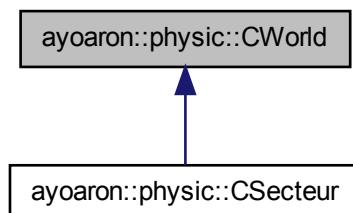
- [CSolideRigide.h](#)
- [CSolideRigide.cpp](#)

7.53 Référence de la classe ayoaron ::physic ::CWorld

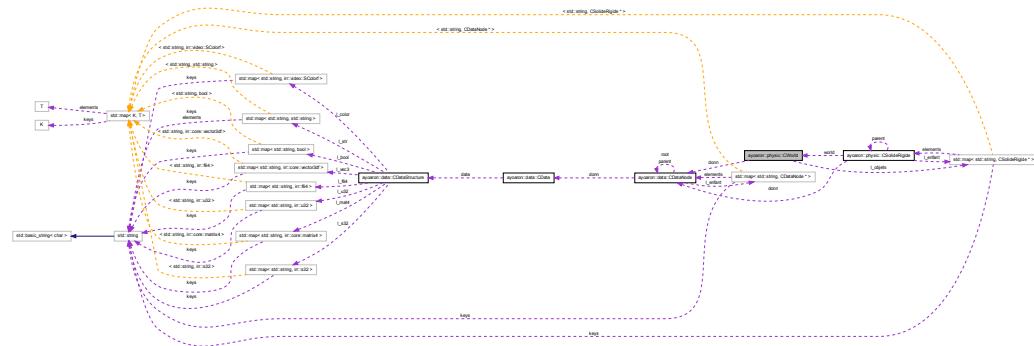
Classe de définition d'un monde d'objet.

```
#include <CWorld.h>
```

Graphe d'héritage de ayoaron ::physic ::CWorld :



Graphe de collaboration de ayoaron ::physic ::CWorld :



Fonctions membres publiques

- **CWorld (data : :CDataNode *d, irr : :scene : :ISceneManager *smg)**
Constructeur.
- **virtual void frame (double dt)**
mise à jour
- **CEffort getGlobalEffort (CSolideRigide *obj)**
revoie l'effort global au point donné (gravité, champ électrique, etc...)
- **irr : :scene : :ISceneManager * getSceneManager ()**
permet l'accès au gestionnaire de scène
- **virtual void loadData (data : :CDataNode *o)**
Charge la galaxie depuis une structure de données.
- **virtual ~CWorld ()**
destructeur

Attributs protégés

- **data : :CDataNode * donn**
pointeur vers la structure de donnée
- **std : :map< std : :string, CSolideRigide * > _objets**
liste des objets enfant
- **irr : :scene : :ISceneManager * smgr**
pointeur vers le scene manager

7.53.1 Description détaillée

Classe de définition d'un monde d'objet.

7.53.2 Documentation des constructeurs et destructeur

7.53.2.1 ayoaron ::physic ::CWorld (data : :CDataNode * d, irr : :scene : :ISceneManager * smg)

Constructeur.

Paramètres

in	<i>d</i>	pointeur vers la structure de donnée
in	<i>smgr</i>	pointeur vers le manger de scene

Références donn, loadData(), ayoaron ::data ::CDataNode ::setData(), et smgr.

Voici le graphe d'appel pour cette fonction :

**7.53.2.2 ayoaron ::physic ::CWorld ::~CWorld() [virtual]**

destructeur

Références donn, et l_objets.

7.53.3 Documentation des fonctions membres**7.53.3.1 void ayoaron ::physic ::CWorld ::frame(double *dt*) [virtual]**

mise à jour

Paramètres

in	<i>dt</i>	le pas de temps
----	-----------	-----------------

on pre frame tous les objets enfants

on frame tous les objets enfants

A faire

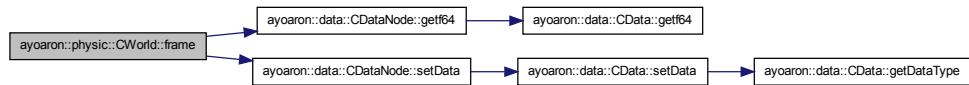
gestion des collisions

on post frame tous les objets enfants

Réimplémentée dans [ayoaron ::physic ::CSecteur](#).

Références donn, ayoaron ::data ::CDataNode ::getf64(), l_objets, et ayoaron ::data ::CDataNode ::setData().

Voici le graphe d'appel pour cette fonction :



7.53.3.2 CEffort ayoaron ::physic ::CWorld ::getGlobalEffort (CSolideRigide * *obj*)

revoie l'effort global au point donné (gravité, champ électrique, etc...)

Paramètres

obj l'objet sur lequel doit s'appliquer l'effort global

Renvoie

la valeur de l'effort au centre de gravité de l'objet exprimé dans le repère global

Référencé par ayoaron ::physic ::CSolideRigide ::frame().

Voici le graphe d'appel pour cette fonction :



7.53.3.3 irr ::scene ::ISceneManager* ayoaron ::physic ::CWorld ::getSceneManager () [inline]

permet l'accès au gestionnaire de scène

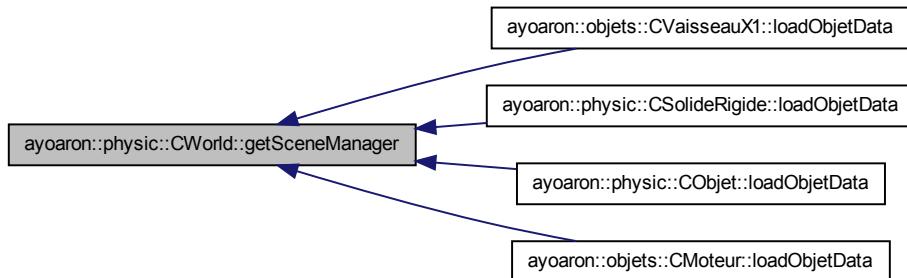
Renvoie

un pointeur vers le ISceneManager courrant

Références smgr.

Référencé par ayoaron ::objets ::CVaisseauX1 ::loadObjetData(), ayoaron ::physic ::CSolideRigide ::loadObjetData(), ayoaron ::physic ::CObjet ::loadObjetData(), et ayoaron ::objets ::CMoteur ::loadObjetData().

Voici le graphe d'appel pour cette fonction :



7.53.3.4 void ayoaron ::physic ::CWorld ::loadData (data : :CDataNode * o) [virtual]

Charge la galaxie depuis une structure de données.

Paramètres

in	<i>o</i>	la structure de données (remplie) à charger
----	----------	---

Réimplémentée dans [ayoaron ::physic ::CSecteur](#).

Références donn.

Référencé par CWorld().

Voici le graphe d'appel pour cette fonction :



7.53.4 Documentation des données membres

7.53.4.1 data : :CDataNode* ayoaron ::physic ::CWorld ::donn [protected]

pointeur vers la structure de donnée

Référencé par ayoaron ::physic ::CSecteur ::CSecteur(), CWorld(), frame(), ayo-

ron :: physic :: CSecteur :: frame(), ayoaron :: physic :: CSecteur :: getDataNode(), ayoaron :: physic :: CSecteur :: getNom(), loadData(), ayoaron :: physic :: CSecteur :: loadData(), ayoaron :: physic :: CSecteur :: setNom(), et ~CWorld().

7.53.4.2 std :: map<std :: string, CSolideRigide*> ayoaron :: physic :: CWorld :: I_-
objets [protected]

liste des objets enfant

Référencé par ayoaron :: physic :: CSecteur :: CSecteur(), frame(), ayoaron :: physic :: CSecteur :: frame(), et ~CWorld().

7.53.4.3 irr :: scene :: ISceneManager* ayoaron :: physic :: CWorld :: smgr
[protected]

pointeur vers le scene manager

Référencé par CWorld(), getSceneManager(), et ayoaron :: physic :: CSecteur :: loadData().

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

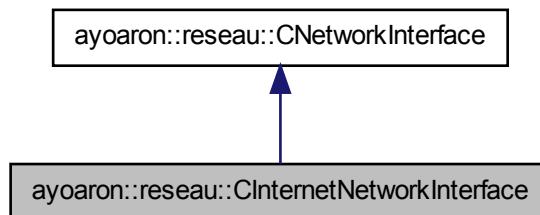
- [CWorld.h](#)
- [CWorld.cpp](#)

7.54 Référence de la classe ayoaron ::reseau ::CInternetNetworkInterface

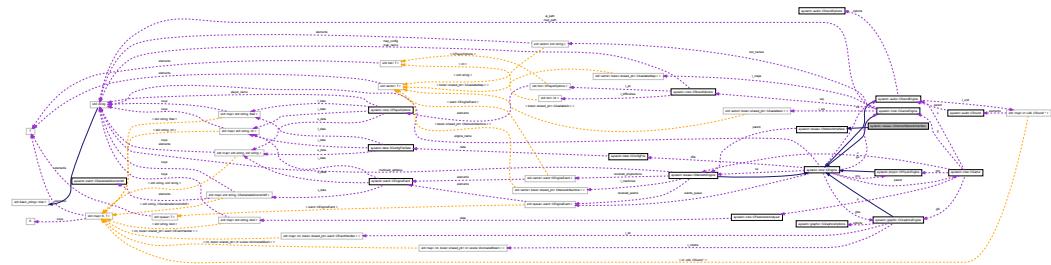
interface réseau internet

```
#include <CInternetNetworkInterface.h>
```

Graphe d'héritage de ayoaron ::reseau ::CInternetNetworkInterface :



Graphe de collaboration de ayoaron ::reseau ::CInternetNetworkInterface :



Types publics

- enum { **LAN, INTERNET** }
type d'interface

Fonctions membres publiques

- **CInternetNetworkInterface (CNetworkEngine *)**
constructeur
- int **getType ()**
revoie le type d'interface
- void **sendAdd (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendMove (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendRem (int, int)**
- void **sendToAllGameTCP (event : :CEngineEvent &, int)**
envoie du serveur vers tous les clients
- void **sendToAllGameUDP (event : :CEngineEvent &, int)**
diffuse un message à tout le monde
- void **sendToServerTCP (event : :CEngineEvent &)**
envoie du client vers le serveur
- virtual **~CInternetNetworkInterface ()**
destructeur

Attributs protégés

- **CNetworkEngine * parent**
le moteur réseau parent
- int **type**
type d'interface

7.54.1 Description détaillée

interface réseau internet

7.54.2 Documentation des énumérations membres

7.54.2.1 anonymous enum [inherited]

type d'interface

Valeurs énumérées :

LAN

INTERNET

7.54.3 Documentation des constructeurs et destructeur

7.54.3.1 ayoaron ::reseau ::CInternetNetworkInterface ::CInternetNetworkInterface (CNetworkEngine * *n*)

constructeur

Paramètres

<i>n</i>	moteur réseau parent
----------	----------------------

Références ayoaron ::reseau ::CNetworkInterface ::INTERNET, et ayoaron ::reseau ::CNetworkInterface ::type.

7.54.3.2 ayoaron ::reseau ::CInternetNetworkInterface ::~CInternetNetworkInterface () [virtual]

destructeur

7.54.4 Documentation des fonctions membres

7.54.4.1 int ayoaron ::reseau ::CNetworkInterface ::getType () [inline, inherited]

revoie le type d'interface

Renvoie

le type d'interface

Références ayoaron ::reseau ::CNetworkInterface ::type.

7.54.4.2 void ayoaron ::reseau ::CNetworkInterface ::sendAdd (int *id*, event ::CSerializableVector3df *pos*, event ::CSerializableVector3df *rot*, int *expeditor*) [inherited]

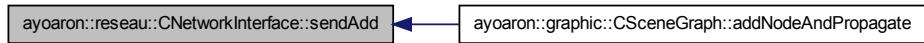
Références ayoaron ::event ::CEngineEvent ::ADD_NODE, ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::addNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.54.4.3 void ayoaron ::reseau ::CNetworkInterface ::sendMove (int *id*,
event ::CSerializableVector3df *pos*, event ::CSerializableVector3df *rot*,
int *expeditor*) [inherited]

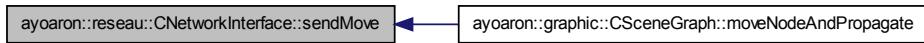
Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::MOVE_-
NODE, ayoaron ::reseau ::CNetworkInterface ::sendToAllGameUDP(), ayoaron ::event ::CEn-
gineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::moveNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.54.4.4 void ayoaron ::reseau ::CNetworkInterface ::sendRem (int *id*, int *expeditor*)
[inherited]

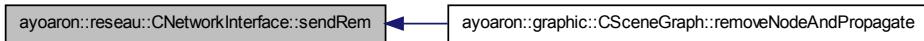
Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::REM_NODE, ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), et ayoaron ::event ::CEngineEvent ::type.

Référencé par ayoaron ::graphic ::CSceneGraph ::removeNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.54.4.5 void ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP (event ::CEngineEvent & *e*, int *exp*) [inherited]

envoie du serveur vers tous les clients

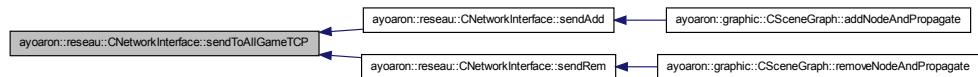
Paramètres

in	<i>e</i>	l'événement à envoyer
in	<i>exp</i>	la machine à qui il ne faut pas envoyer

Références ayoaron ::reseau ::CNetworkEngine ::l_machines, et ayoaron ::reseau ::CNetworkInterface ::parent.

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendAdd(), et ayoaron ::reseau ::CNetworkInterface ::sendRem().

Voici le graphe d'appel pour cette fonction :



7.54.4.6 void ayoaron ::reseau ::CInternetNetworkInterface ::sendToAllGameUDP (event : :CEngineEvent & e, int expeditor) [virtual]

diffuse un message à tout le monde

Paramètres

in	e	le messag
in	expeditor	l'expéditeur

Implémente [ayooron ::reseau ::CNetworkInterface](#).

Références ayoaron ::reseau ::CNetworkEngine ::!_machines, et ayoaron ::reseau ::CNetworkInterface ::parent.

7.54.4.7 void ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP (event : :CEngineEvent & e) [inherited]

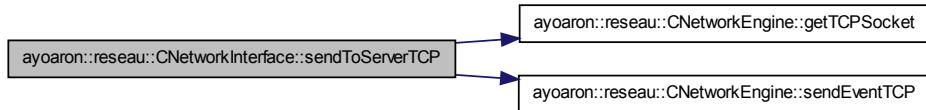
envoie du client vers le serveur

Paramètres

in	e	l'evenement à envoeyr
----	---	-----------------------

Références ayoaron ::reseau ::CNetworkEngine ::getTCPSocket(), ayoaron ::reseau ::CNetworkInterface ::parent, et ayoaron ::reseau ::CNetworkEngine ::sendEventTCP().

Voici le graphe d'appel pour cette fonction :



7.54.5 Documentation des données membres

7.54.5.1 CNetworkEngine* ayoaron ::reseau ::CNetworkInterface ::parent
 [protected, inherited]

le moteur réseau parent

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), ayoaron ::reseau ::CLANNetworkInterface ::sendToAllGameUDP(), sendToAllGameUDP(), et ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP().

7.54.5.2 int ayoaron ::reseau ::CNetworkInterface ::type [protected, inherited]

type d'interface

Référencé par CInternetNetworkInterface(), ayoaron ::reseau ::CLANNetworkInterface ::CLANNetworkInterface(), et ayoaron ::reseau ::CNetworkInterface ::getType().

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

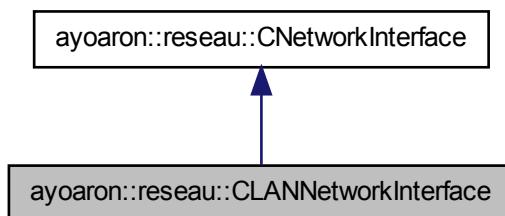
- [CInternetNetworkInterface.h](#)
- [CInternetNetworkInterface.cpp](#)

7.55 Référence de la classe ayoaron ::reseau ::CLANNetworkInterface

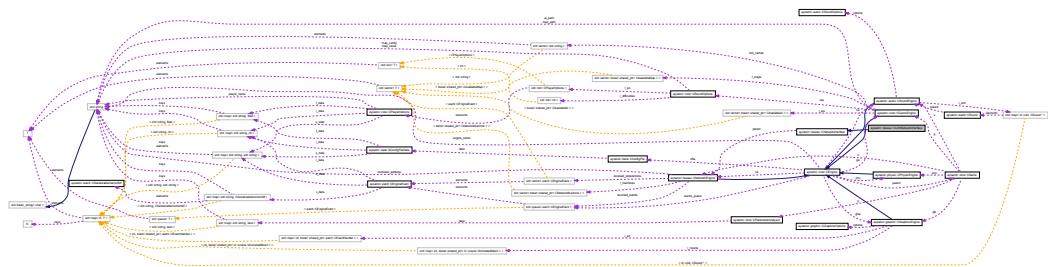
interface réseau LAN

```
#include <CLANNetworkInterface.h>
```

Graphe d'héritage de ayoaron ::reseau ::CLANNetworkInterface :



Graphe de collaboration de ayoaron ::reseau ::CLANNetworkInterface :



Types publics

- enum { **LAN, INTERNET** }
type d'interface

Fonctions membres publiques

- **CLANNetworkInterface (CNetworkEngine *)**
constructeur
- int **getType ()**
revoie le type d'interface
- void **sendAdd (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendMove (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendRem (int, int)**
- void **sendToAllGameTCP (event : :CEngineEvent &, int)**
envoie du serveur vers tous les clients
- void **sendToAllGameUDP (event : :CEngineEvent &, int)**
diffuse un message à tout le monde
- void **sendToServerTCP (event : :CEngineEvent &)**
envoie du client vers le serveur
- virtual ~**CLANNetworkInterface ()**
destructeur

Attributs protégés

- **CNetworkEngine * parent**
le moteur réseau parent
- int **type**
type d'interface

7.55.1 Description détaillée

interface réseau LAN

7.55.2 Documentation des énumérations membres

7.55.2.1 anonymous enum [inherited]

type d'interface

Valeurs énumérées :

LAN

INTERNET

7.55.3 Documentation des constructeurs et destructeur

7.55.3.1 ayoaron ::reseau ::CLANNetworkInterface ::CLANNetworkInterface (CNetworkEngine * *n*)

constructeur

Paramètres

<i>n</i>	moteur réseau parent
----------	----------------------

Références ayoaron ::reseau ::CNetworkInterface ::LAN, et ayoaron ::reseau ::CNetworkInterface ::type.

7.55.3.2 ayoaron ::reseau ::CLANNetworkInterface ::~CLANNetworkInterface () [virtual]

destructeur

7.55.4 Documentation des fonctions membres

7.55.4.1 int ayoaron ::reseau ::CNetworkInterface ::getType () [inline, inherited]

revoie le type d'interface

Renvoie

le type d'interface

Références ayoaron ::reseau ::CNetworkInterface ::type.

7.55.4.2 void ayoaron ::reseau ::CNetworkInterface ::sendAdd (int *id*, event ::CSerializableVector3df *pos*, event ::CSerializableVector3df *rot*, int *expeditor*) [inherited]

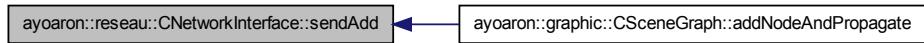
Références ayoaron ::event ::CEngineEvent ::ADD_NODE, ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::addNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

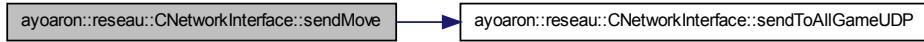


7.55.4.3 `void ayoaron ::reseau ::CLANNetworkInterface ::sendMove (int id, event ::CSerializableVector3df pos, event ::CSerializableVector3df rot, int expeditor) [inherited]`

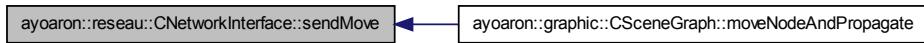
Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::MOVE_NODE, ayoaron ::reseau ::CLANNetworkInterface ::sendToAllGameUDP(), ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::moveNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.55.4.4 void ayoaron ::reseau ::CNetworkInterface ::sendRem (int *id*, int *expeditor*)
 [inherited]

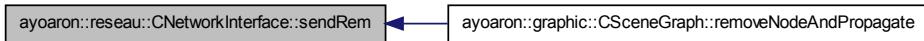
Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::REM_NODE, ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), et ayoaron ::event ::CEngineEvent ::type.

Référencé par ayoaron ::graphic ::CSceneGraph ::removeNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.55.4.5 void ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP (event ::CEngineEvent & *e*, int *exp*) [inherited]

envoie du serveur vers tous les clients

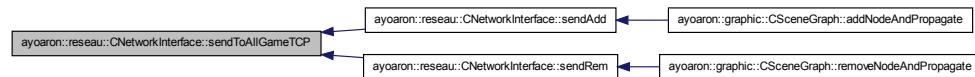
Paramètres

in	<i>e</i>	l'événement à envoyer
in	<i>exp</i>	la machine à qui il ne faut pas envoyer

Références ayoaron ::reseau ::CNetworkEngine ::l_machines, et ayoaron ::reseau ::CNetworkInterface ::parent.

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendAdd(), et ayoaron ::reseau ::CNetworkInterface ::sendRem().

Voici le graphe d'appel pour cette fonction :



7.55.4.6 void ayoaron ::reseau ::CLANNetworkInterface ::sendToAllGameUDP (event : :CEngineEvent & e, int expeditor) [virtual]

diffuse un message à tout le monde

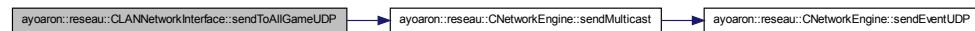
Paramètres

in	e	le messag
in	expeditor	l'expéditeur

Implémente ayoaron ::reseau ::CNetworkInterface.

Références ayoaron ::reseau ::CNetworkInterface ::parent, et ayoaron ::reseau ::CNetworkEngine ::sendMulticast().

Voici le graphe d'appel pour cette fonction :



7.55.4.7 void ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP (event : :CEngineEvent & e) [inherited]

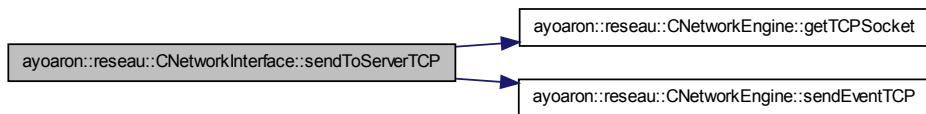
envoie du client vers le serveur

Paramètres

in	e	l'événement à envoyer
----	---	-----------------------

Références ayoaron ::reseau ::CNetworkEngine ::getTCPSocket(), ayoaron ::reseau ::CNetworkInterface ::parent, et ayoaron ::reseau ::CNetworkEngine ::sendEventTCP().

Voici le graphe d'appel pour cette fonction :



7.55.5 Documentation des données membres

7.55.5.1 **CNetworkEngine* ayoaron ::reseau ::CNetworkInterface ::parent** [protected, inherited]

le moteur réseau parent

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), sendToAllGameUDP(), ayoaron ::reseau ::CIinternetNetworkInterface ::sendToAllGameUDP(), et ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP().

7.55.5.2 **int ayoaron ::reseau ::CNetworkInterface ::type** [protected, inherited]

type d'interface

Référencé par ayoaron ::reseau ::CIinternetNetworkInterface ::CIinternetNetworkInterface(), CLANNetworkInterface(), et ayoaron ::reseau ::CNetworkInterface ::getType().

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

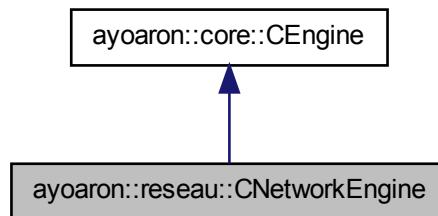
- [CLANNetworkInterface.h](#)
- [CLANNetworkInterface.cpp](#)

7.56 Référence de la classe ayoaron ::reseau ::CNetworkEngine

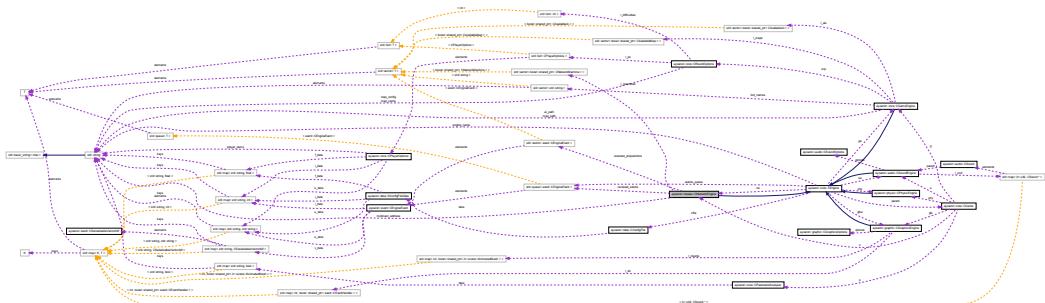
moteur réseau

```
#include <CNetworkEngine.h>
```

Graphe d'héritage de ayoaron ::reseau ::CNetworkEngine :



Graphe de collaboration de ayoaron ::reseau ::CNetworkEngine :



Types publics

- enum {
 `STATE NOTHING, STATE QUITTING, STATE_WAITING_PLAYERS, STATE_PLAYING_INTRO,`
`STATE_LOADING_ROUND, STATE_PLAYING }`

Constantes d'état.
- enum {
 `LAN_SERVER, LAN_CLIENT, INTERNET_SERVER, INTERNET_CLIENT,`
`NONE }`

Fonctions membres publiques

- void `attachGameEngine (CGameEngine *e)`
lien avec le moteur de jeu
- void `attachGraphicsEngine (graphic ::CGraphicsEngine *e)`
lien avec le moteur de rendu graphique
- void `attachNetworkEngine (reseau ::CNetworkEngine *e)`

- void **attachPhysicEngine** (physic : :CPhysicEngine *e)
 - lien avec le moteur de physique*
- void **attachSoundEngine** (audio : :CSoundEngine *e)
 - lien avec le moteur de son*
- void **broadcastGameProposition** (event : :CEngineEvent &)
 - lien vers le moteur de jeu*
- void **buildInternetInterface** ()
 - lien vers le moteur de réseau*
- void **buildLANInterface** ()
 - lien vers le moteur de rendu graphique*
- void **cleanInterface** ()
 - lien vers le moteur de réseau*
- void **clientConnectsToServer** (std : :string &, std : :string &, std : :string &)
 - lien vers le moteur de réseau*
- **CNetworkEngine** (core : :CGame *)
 - construteur*
- void **frame** (double dt)
 - lien vers le moteur de jeu*
- **CGameEngine** * **getGameEngine** ()
 - lien vers le moteur de jeu*
- **graphic** : :CGraphicsEngine * **getGraphicsEngine** ()
 - lien vers le moteur de rendu graphique*
- boost : :shared_ptr< **CNetworkMachine** > **getMachineFromAddress** (std : :string)
 - lien vers le moteur de réseau*
- boost : :shared_ptr< **CNetworkMachine** > **getMachineFromId** (int)
 - lien vers le moteur de réseau*
- boost : :shared_ptr< **CNetworkMachine** > **getMachineFromPlayerName** (std : :string &)
 - lien vers le moteur de réseau*
- **reseau** : :CNetworkEngine * **getNetworkEngine** ()
 - lien vers le moteur de réseau*
- int **getNextMachineId** ()
 - lien vers le moteur de réseau*
- **CGame** * **getParent** ()
 - lien vers la classe game parente*
- **audio** : :CSoundEngine * **getSoundEngine** ()
 - lien vers le moteur de son*
- boost : :asio : :ip : :tcp : :socket * **getTCPSocket** ()
 - lien vers le moteur de réseau*
- bool **is_running** ()
 - lien vers le moteur de réseau*
- bool **isAddressRegistered** (std : :string)
 - lien vers le moteur de réseau*
- void **listenGamePropositions** ()
 - lien vers le moteur de jeu*
- int **playAsWhat** ()
 - lien vers le moteur de jeu*
- void **processQueue** ()
 - vidange de l'event_queue*
- void **pushEvent** (event : :CEngineEvent &e)
 - ajout d'un événement*
- void **pushReceivedEvent** (event : :CEngineEvent &)
 - lien vers le moteur de jeu*
- void **sendEventTCP** (event : :CEngineEvent &, boost : :asio : :ip : :tcp : :socket *)
 - envoie un message au moteur de réseau*
- void **sendEventUDP** (event : :CEngineEvent &, boost : :asio : :ip : :udp : :socket *)
 - envoie un message au moteur de réseau*
- void **sendMessageToGame** (event : :CEngineEvent &e)
 - envoie un message au moteur de jeu*
- void **sendMessageToGraphics** (event : :CEngineEvent &e)
 - envoie un message au moteur de rendu graphique*
- void **sendMessageToNetwork** (event : :CEngineEvent &e)
 - envoie un message au moteur de réseau*
- void **sendMessageToPhysic** (event : :CEngineEvent &e)
 - envoie un message au moteur physique*
- void **sendMessageToSound** (event : :CEngineEvent &e)
 - envoie un message au moteur de son*
- void **sendMulticast** (event : :CEngineEvent &)
 - lien vers le moteur de réseau*
- void **setPlayAs** (int i)
 - lien vers le moteur de jeu*
- virtual ~**CNetworkEngine** ()
 - Destructeur.*

Attributs publics

- std : :vector< boost : :shared_ptr< **CNetworkMachine** > > **I_machines**
 - lien vers le moteur de réseau*
- boost : :shared_ptr< **CNetworkInterface** > **ni**
 - lien vers le moteur de rendu graphique*
- int **port_multicast**
 - lien vers le moteur de réseau*

- int **port_tcp_server_connection**
- int **port_udp_reception**
- int **state**
l'état actuel
- boost ::mutex **state_mutex**
Mutex de protection de l'état.

Attributs protégés

- **data** ::CConfigFile *cfile*
Fichier de configuration.
- std ::string **engine_name**
nom de ce moteur
- std ::queue<event ::CEngineEvent> **events_queue**
liste des événement à traiter
- CGameEngine * **ge**
pointeur vers le moteur de jeu
- **graphic** ::CGraphicsEngine * *gixe*
pointeur vers le moteur de rendu graphique
- **reseau** ::CNetworkEngine * *ne*
pointeur vers le moteur de réseau
- CGame * **parent**
pointeur vers le parent
- bool **passive_mode**
true si mode passif activé
- **physic** ::CPhysicEngine * *phy*
pointeur vers le moteur de physique
- boost ::mutex **queue_mutex**
Mutex de protection de la liste d'événement event_queue.
- **audio** ::CSoundEngine * *se*
pointeur vers le moteur de son

Fonctions membres privées

- void **getConfigData** ()
récupération de la configuration actuelle
- void **handleAcceptTCP** (const boost ::system ::error_code &, boost ::asio ::ip ::tcp ::socket *)
- void **listenMulti3secondsUDP** ()
- void **multicastReceive** (const boost ::system ::error_code &, std ::size_t)
- void **processEvent** (event ::CEngineEvent &)
- void **processReceivedEvents** (event ::CEngineEvent &)
- void **processReceivedEventsQueue** ()
- void **queueEventInGame** (event ::CEngineEvent &)
- void **sendEventToAll** (event ::CEngineEvent &)
- void **serverThreadTCPReceive** (boost ::asio ::ip ::tcp ::socket *)
- void **threadSendMulticast** (event ::CEngineEvent)
- void **udpAsyncRead** (const boost ::system ::error_code &, std ::size_t)

Attributs privés

- boost ::asio ::io_service **io**
- boost ::mutex **l_receive_queue**
- std ::string **multicast_address**
- boost ::array<char, BUFFER_SIZE> **network_buffer**
- int **next_machine_id**
- int **play_as**

- boost ::mutex `propositions_mutex`
- std ::queue< `event` ::CEngineEvent > `received_events`
- std ::vector< `event` ::CEngineEvent > `received_propositions`
- bool `running`
- boost ::asio ::ip ::udp ::socket * `s_multicast`
- boost ::asio ::ip ::tcp ::socket * `s_tcp`
- boost ::asio ::ip ::udp ::socket * `s_udp_in`
- boost ::asio ::ip ::tcp ::acceptor * `tcp_acceptor`
- boost ::asio ::ip ::udp ::endpoint `udp_remote_endpoint`

7.56.1 Description détaillée

moteur réseau

appelé depuis le moteur de jeu

7.56.2 Documentation des énumérations membres

7.56.2.1 anonymous enum [inherited]

Constantes d'état.

Valeurs énumérées :

- `STATE_NOTHING`
- `STATE_QUITTING`
- `STATE_WAITING_PLAYERS`
- `STATE_PLAYING_INTRO`
- `STATE_LOADING_ROUND`
- `STATE_PLAYING`

7.56.2.2 anonymous enum

Valeurs énumérées :

- `LAN_SERVER`
- `LAN_CLIENT`
- `INTERNET_SERVER`
- `INTERNET_CLIENT`
- `NONE`

7.56.3 Documentation des constructeurs et destructeur

7.56.3.1 ayoaron ::reseau ::CNetworkEngine ::CNetworkEngine (core ::CGame * `g`)

constructeur

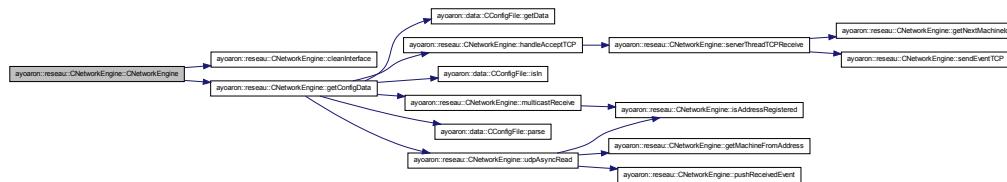
Paramètres

in	<code>g</code>	jeu parent
----	----------------	------------

constructor

Références cleanInterface(), ayoaron ::core ::CEngine ::engine_name, getConfigData(), next_machine_id, ni, running, ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::STATE NOTHING, et tcp acceptor.

Voici le graphe d'appel pour cette fonction :



7.56.3.2 ayoaron ::reseau ::CNetworkEngine ::~CNetworkEngine () [virtual]

Destructeur.

Références io, ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, et ayoaron ::core ::CEngine ::STATE QUITTING.

7.56.4 Documentation des fonctions membres

7.56.4.1 void ayoaron ::core ::CEngine ::attachGameEngine (CGameEngine * e) [inline, inherited]

lien avec le moteur de jeu

Paramètres

in	e	pointeur vers le moteur de jeu à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.56.4.2 void ayoaron ::core ::CEngine ::attachGraphicsEngine (graphic : :CGraphicsEngine * e) [inline, inherited]

lien avec le moteur de rendu graphique

Paramètres

in	e	pointeur vers le moteur de rendu graphique à lier
----	---	---

Cette méthode est utiliser pour lier le moteur courant au moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gfxe.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.56.4.3 void ayoaron ::core ::CEngine ::attachNetworkEngine (reseau ::CNetworkEngine * e) [inline, inherited]

lien avec le moteur de réseau

Paramètres

in	e	pointeur vers le moteur de réseau à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.56.4.4 void ayoaron ::core ::CEngine ::attachPhysicEngine (physic ::CPhysicEngine * e) [inline, inherited]

lien avec le moteur de physique

Paramètres

in	e	pointeur vers le moteur de physique à lier
----	---	--

Cette méthode est utiliser pour lier le moteur courant au moteur de physique

Références ayoaron ::core ::CEngine ::phy.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.56.4.5 void ayoaron ::core ::CEngine ::attachSoundEngine (audio ::CSoundEngine * e) [inline, inherited]

lien avec le moteur de son

Paramètres

in	e	pointeur vers le moteur de son à lier
----	---	---------------------------------------

Cette méthode est utiliser pour lier le moteur courant au moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::core ::CGame ::CGame().

Voici le graphe d'appel pour cette fonction :



7.56.4.6 void ayoaron ::reseau ::CNetworkEngine ::broadcastGameProposition (event : :CEngineEvent & gp)

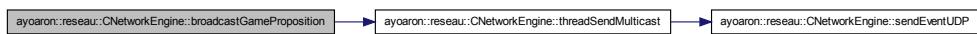
the server who has created the game informs the whole network that the game exists game proposition is sent by multicast on port port_broadcast_game_proposition this function is non blocking, it starts a thread. only used on LAN server

Paramètres

gp CEngineEvent to broadcast

Références ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, ayoaron ::core ::CEngine ::STATE_WAITING_PLAYERS, et threadSendMulticast().

Voici le graphe d'appel pour cette fonction :



7.56.4.7 void ayoaron ::reseau ::CNetworkEngine ::buildInternetInterface ()

select if LAN or internet has to be created almost same as build_LAN_interface

Références cleanInterface(), et ni.

Voici le graphe d'appel pour cette fonction :



7.56.4.8 void ayoaron ::reseau ::CNetworkEngine ::buildLANInterface ()

select if LAN or internet has to be created almost same as build_Internet_interface

Références cleanInterface(), et ni.

Voici le graphe d'appel pour cette fonction :



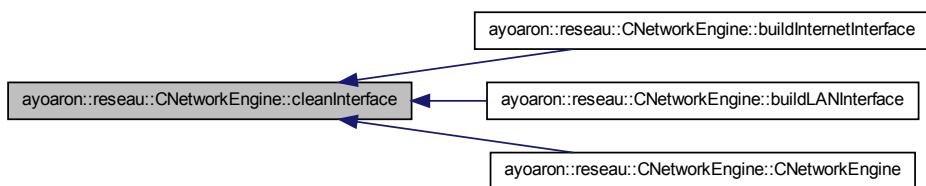
7.56.4.9 void ayoaron ::reseau ::CNetworkEngine ::cleanInterface ()

reconstruct the sockets to be called each time the network status is changed (server/client/lan/internet)

Références ni, s_multicast, s_tcp, et s_udp_in.

Référencé par buildInternetInterface(), buildLANInterface(), et CNetworkEngine().

Voici le graphe d'appel pour cette fonction :



7.56.4.10 void ayoaron ::reseau ::CNetworkEngine ::clientConnectsToServer (std ::string & host, std ::string & nick, std ::string & pass)

client connects to the server

Paramètres

host	hostname to connect
nick	players's nickname
pass	password to join the game

Références getNextMachineId(), et l_machines.

Voici le graphe d'appel pour cette fonction :



7.56.4.11 void ayoaron ::reseau ::CNetworkEngine ::frame (double dt) [virtual]

main network function. Called on each frame.

Implémente [ayoaron ::core ::CEngine](#).

Références ayoaron ::physic ::CGalaxie ::secteurcourant.

Référencé par ayoaron ::core ::CGame ::run().

Voici le graphe d'appel pour cette fonction :



7.56.4.12 void ayoaron ::reseau ::CNetworkEngine ::getConfigData() [private, virtual]

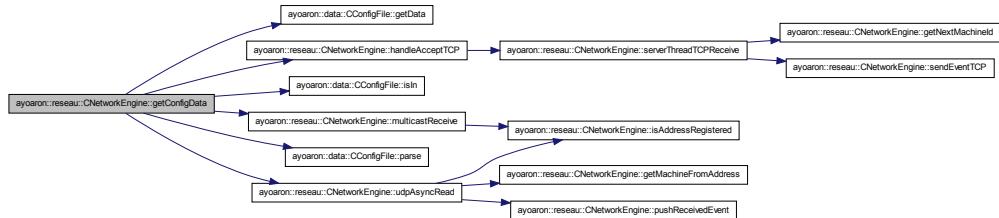
récupération de la configuration actuelle

Implémente [ayoaron ::core ::CEngine](#).

Références BUFFER_SIZE, ayoaron ::core ::CEngine ::cfile, ayoaron ::data ::CConfigFile ::getData(), handleAcceptTCP(), ayoaron ::data ::SConfigFileData ::i_data, io, ayoaron ::data ::CConfigFile ::isIn(), multicast_address, multicastReceive(), network_buffer, ayoaron ::data ::CConfigFile ::parse(), port_multicast, port_tcp_server_connection, port_udp_reception, s_multicast, s_tcp, s_udp_in, tcp_acceptor, udp_remote_endpoint, et udpAsyncRead().

Référencé par CNetworkEngine().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.13 CGameEngine* ayoaron ::core ::CEngine ::getGameEngine() [inline, inherited]

lien vers le moteur de jeu

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de jeu

Références ayoaron ::core ::CEngine ::ge.

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect(), ayoaron ::graphic ::CGuiSystem ::handleProcessCommand(), ayoaron ::graphic ::CGuiSystem ::loadGame(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::event ::CGameEventHandler ::onEvent(), et ayoaron ::graphic ::CGraphicsEngine ::processEvent().

Voici le graphe d'appel pour cette fonction :



7.56.4.14 graphic ::CGraphicsEngine* ayoaron ::core ::CEngine ::getGraphicsEngine() [inline, inherited]

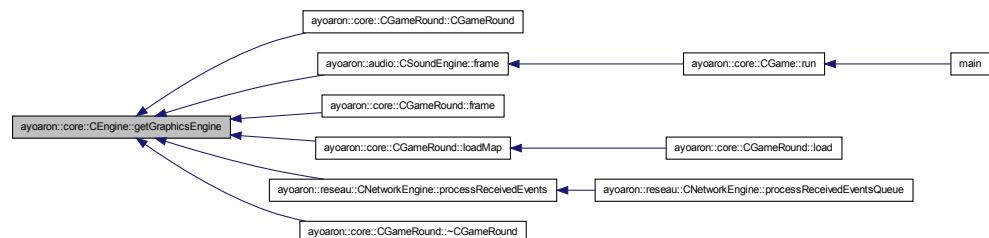
lien vers le moteur de rendu graphique

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gfx.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::core ::CGameRound ::frame(), ayoaron ::core ::CGameRound ::loadMap(), processReceivedEvents(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



**7.56.4.15 boost ::shared_ptr< CNetworkMachine >
ayoaron ::reseau ::CNetworkEngine ::getMachineFromAddress(std ::string s)**

machine accessor

Paramètres

<i>s</i>	: ip address of the machine to find
----------	-------------------------------------

Références l_machines.

Référencé par udpAsyncRead().

Voici le graphe d'appel pour cette fonction :



7.56.4.16 boost ::shared_ptr< CNetworkMachine > ayoaron ::reseau ::CNetworkEngine ::getMachineFromId (int *id*)

machine accessor

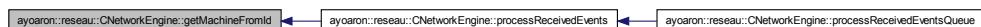
Paramètres

<i>id</i>	: id of the machine to get
-----------	----------------------------

Références l_machines.

Référencé par processReceivedEvents().

Voici le graphe d'appel pour cette fonction :



7.56.4.17 boost ::shared_ptr< CNetworkMachine > ayoaron ::reseau ::CNetworkEngine ::getMachineFromPlayerName (std ::string & *s*)

machine accessor

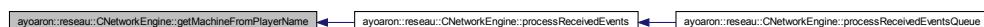
Paramètres

<i>s</i>	: name of the player
----------	----------------------

Références l_machines.

Référencé par processReceivedEvents().

Voici le graphe d'appel pour cette fonction :



7.56.4.18 reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::getNetworkEngine()
 [inline, inherited]

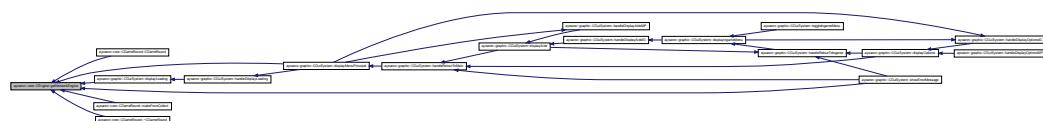
lien vers le moteur de réseau

Cette méthode est utiliser pour obtenir un pointeur vers le moteur de réseau

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::core ::CGameRound ::CGameRound(), ayoaron ::graphic ::CGuiSystem ::displayLoading(), ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal(), ayoaron ::core ::CGameRound ::makeFromCollect(), ayoaron ::graphic ::CGuiSystem ::showErrorMessage(), et ayoaron ::core ::CGameRound ::~CGameRound().

Voici le graphe d'appel pour cette fonction :



7.56.4.19 int ayoaron ::reseau ::CNetworkEngine ::getNextMachineId()

generates a new network_machine_id

Références next_machine_id.

Référencé par clientConnectsToServer(), et serverThreadTCPReceive().

Voici le graphe d'appel pour cette fonction :



7.56.4.20 CGame* ayoaron ::core ::CEngine ::getParent() [inline, inherited]

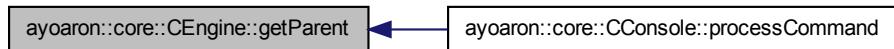
lien vers la classe game parente

Cette méthode est utiliser pour obtenir un pointeur vers le parent

Références ayoaron ::core ::CEngine ::parent.

Référencé par ayoaron ::core ::CConsole ::processCommand().

Voici le graphe d'appel pour cette fonction :



7.56.4.21 audio ::CSoundEngine* ayoaron ::core ::CEngine ::getSoundEngine()
 [inline, inherited]

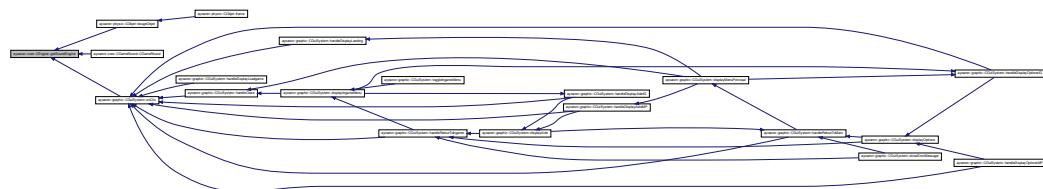
lien vers le moteur de son

Cette méthode est utilisée pour obtenir un pointeur vers le moteur de son

Références ayoaron ::core ::CEngine ::se.

Référencé par ayoaron ::physic ::CObjet ::bougeObjet(), ayoaron ::core ::CGameRound ::CGameRound(), et ayoaron ::graphic ::CGuiSystem ::onClic().

Voici le graphe d'appel pour cette fonction :



7.56.4.22 boost ::asio ::ip ::tcp ::socket* ayoaron ::reseau ::CNetworkEngine ::getTCPSocket() [inline]

Références s_tcp.

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP().

Voici le graphe d'appel pour cette fonction :



```
7.56.4.23 void ayoaron ::reseau ::CNetworkEngine ::handleAcceptTCP ( const
boost ::system ::error_code & e, boost ::asio ::ip ::tcp ::socket * socket )
[private]
```

asynchronous function called each time a tcp connection is accepted

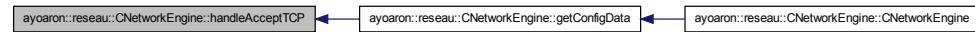
Références io, serverThreadTCPReceive(), ayoaron ::core ::CEngine ::state, et tcp_-acceptor.

Référencé par getConfigData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



```
7.56.4.24 bool ayoaron ::reseau ::CNetworkEngine ::is_running ( ) [inline]
```

Références running.

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect().

Voici le graphe d'appel pour cette fonction :



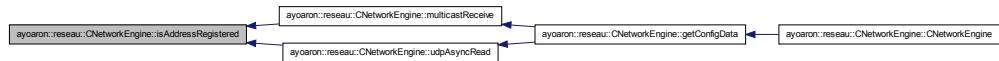
```
7.56.4.25 bool ayoaron ::reseau ::CNetworkEngine ::isAddressRegistered ( std ::string s )
```

check if the address belongs to a registered machine

Références l_machines.

Référencé par multicastReceive(), et udpAsyncRead().

Voici le graphe d'appel pour cette fonction :

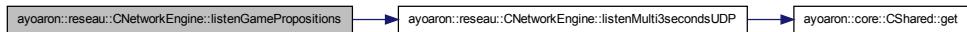


7.56.4.26 void ayoaron ::reseau ::CNetworkEngine ::listenGamePropositions ()

the client listens to the game propositions proposed on the network only used on LAN

Références listenMulti3secondsUDP(), et running.

Voici le graphe d'appel pour cette fonction :



7.56.4.27 void ayoaron ::reseau ::CNetworkEngine ::listenMulti3secondsUDP ()

[private]

thread function. it listens to game propositions and write into collected_propositions each 3 seconds

Références ayoaron ::core ::CShared ::get(), io, propositions_mutex, received_propositions, ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, et ayoaron ::core ::CEngine ::STATE_WAITING_PLAYERS.

Référencé par listenGamePropositions().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.28 void ayoaron ::reseau ::CNetworkEngine ::multicastReceive (const boost ::system ::error_code &, std ::size_t) [private]

asynchronous function called each time we receive something on the multicast socket

Références ayoaron ::event ::CEngineEvent ::GAME_PROP, isAddressRegistered(), network_buffer, propositions_mutex, received_events, received_propositions, ayoaron ::core ::CEngine ::state, et udp_remote_endpoint.

Référencé par getConfigData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.29 int ayoaron ::reseau ::CNetworkEngine ::playAsWhat ()

basic accessor

Références play_as.

Référencé par ayoaron ::core ::CGameRound ::makeFromCollect().

Voici le graphe d'appel pour cette fonction :



7.56.4.30 void ayoaron ::reseau ::CNetworkEngine ::processEvent(event ::CEngineEvent & e) [private, virtual]

process events incoming to the network engine, from the other engines should be very few used. Modify it according to your own needs

Implémente [ayoaron ::core ::CEngine](#).

7.56.4.31 void ayoaron ::core ::CEngine ::processQueue() [inherited]

vidange de l'event_queue

lance pour chaque événement le programme correspondant vide la liste

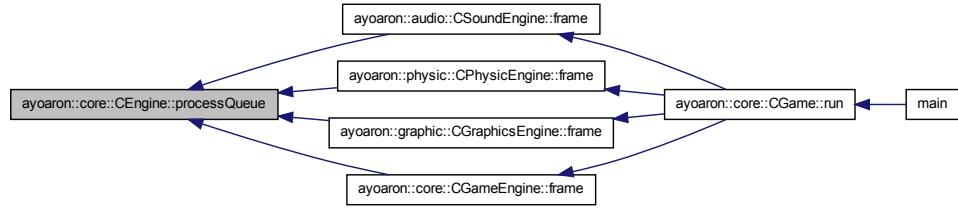
Références ayoaron ::core ::CEngine ::events_queue, ayoaron ::core ::CEngine ::processEvent(), et ayoaron ::core ::CEngine ::queue_mutex.

Référencé par ayoaron ::audio ::CSoundEngine ::frame(), ayoaron ::physic ::CPhysicEngine ::frame(), ayoaron ::graphic ::CGraphicsEngine ::frame(), et ayoaron ::core ::CGameEngine ::frame().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

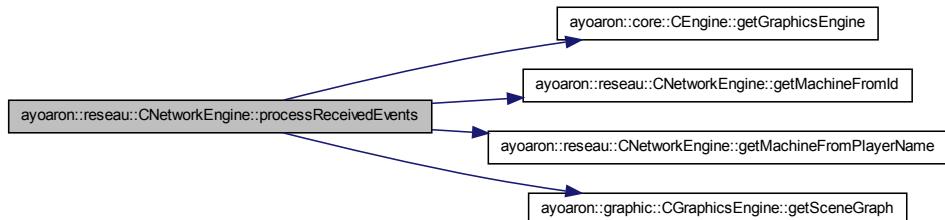


7.56.4.32 void ayoaron ::reseau ::CNetworkEngine ::processReceivedEvents (event ::CEngineEvent & e) [private]

Références ayoaron ::event ::CEngineEvent ::ADD_NODE, ayoaron ::event ::CEngineEvent ::DEL_PLAYER, ayoaron ::core ::CEngine ::getGraphicsEngine(), getMachineFromId(), getMachineFromPlayerName(), ayoaron ::graphic ::CGraphicsEngine ::getSceneGraph(), ayoaron ::event ::CEngineEvent ::i_data, INTERNET_SERVER, I_machines, LAN_SERVER, ayoaron ::event ::CEngineEvent ::MOVE_NODE, ni, play_as, ayoaron ::event ::CEngineEvent ::REM_NODE, ayoaron ::event ::CEngineEvent ::REMOVE_MACHINE, ayoaron ::event ::CEngineEvent ::s_data, ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par processReceivedEventsQueue().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :

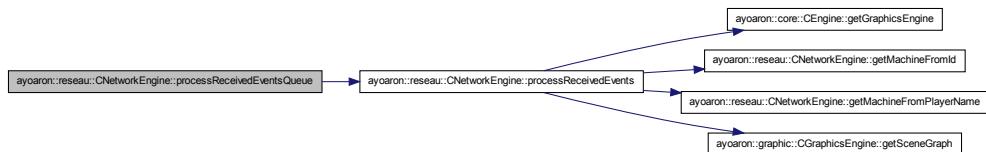


7.56.4.33 void ayoaron ::reseau ::CNetworkEngine ::processReceivedEventsQueue ()
[private]

empty the received events queue thread safe

Références `l_receive_queue`, `processReceivedEvents()`, et `received_events`.

Voici le graphe d'appel pour cette fonction :



7.56.4.34 void ayoaron ::core ::CEngine ::pushEvent (event ::CEngineEvent & e)
[inherited]

ajout d'un événement

Paramètres

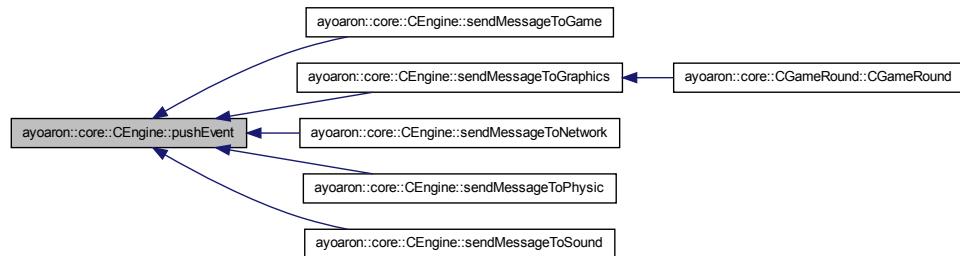
in	e	l'événement à ajouter
----	---	-----------------------

Ajoute l'événement à la liste des à faire

Références `ayoaron ::core ::CEngine ::events_queue`, et `ayoaron ::core ::CEngine ::queue_mutex`.

Référencé par `ayoaron ::core ::CEngine ::sendMessageToGame()`, `ayoaron ::core ::CEngine ::sendMessageToGraphics()`, `ayoaron ::core ::CEngine ::sendMessageToNetwork()`, `ayoaron ::core ::CEngine ::sendMessageToPhysic()`, et `ayoaron ::core ::CEngine ::sendMessageToSound()`.

Voici le graphe d'appel pour cette fonction :



7.56.4.35 void ayoaron ::reseau ::CNetworkEngine ::pushReceivedEvent (event : :CEngineEvent & e)

we have just received an event from a network machine, let's add it to the queue thread safe

Références l_receive_queue, et received_events.

Référencé par ayoaron ::reseau ::CNetworkMachine ::tcpAsyncReceive(), et udpAsyncRead().

Voici le graphe d'appel pour cette fonction :



7.56.4.36 void ayoaron ::reseau ::CNetworkEngine ::queueEventIntoGame (event : :CEngineEvent &) [private]

7.56.4.37 void ayoaron ::reseau ::CNetworkEngine ::sendEventTCP (event : :CEngineEvent & ne, boost ::asio ::ip ::tcp ::socket * s)

massively used function. used to send all the TCP messages similar to send_eventUDP deprecated. remplacé par send_eventTCP(CEngineEvent&, network_machine*)

Références ayoaron ::core ::CEngine ::ne.

Référencé par ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP(), et serverThreadTCPReceive().

Voici le graphe d'appel pour cette fonction :



7.56.4.38 `void ayoaron ::reseau ::CNetworkEngine ::sendEventToAll (event : :CEngineEvent &) [private]`

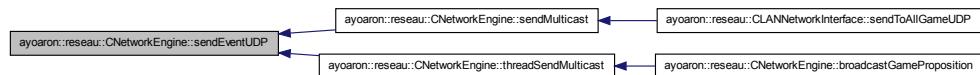
7.56.4.39 `void ayoaron ::reseau ::CNetworkEngine ::sendEventUDP (event : :CEngineEvent & ne, boost ::asio ::ip ::udp ::socket * s)`

massively used function. used to send all the UDP messages similar to send_eventTCP

Références ayoaron ::core ::CEngine ::ne.

Référencé par sendMulticast(), et threadSendMulticast().

Voici le graphe d'appel pour cette fonction :



7.56.4.40 `void ayoaron ::core ::CEngine ::sendMessageToGame (event : :CEngineEvent & e) [inherited]`

envoie un message au moteur de jeu

Paramètres

in	<code>e</code> evenement à envoyer vers le moteur de jeu
----	--

Cette méthode est utiliser pour envoyer un événement vers moteur de jeu

Références ayoaron ::core ::CEngine ::ge, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.56.4.41 void ayoaron ::core ::CEngine ::sendMessageToGraphics (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de rendu graphique

Paramètres

in	e	evenement à envoyer vers le moteur de rendu graphique
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de rendu graphique

Références ayoaron ::core ::CEngine ::gixe, et ayoaron ::core ::CEngine ::pushEvent().

Référencé par ayoaron ::core ::CGameRound ::CGameRound().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.42 void ayoaron ::core ::CEngine ::sendMessageToNetwork (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de réseau

Paramètres

in	e	evenement à envoyer vers le moteur de réseau
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de réseau

Références ayoaron ::core ::CEngine ::ne, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.56.4.43 void ayoaron ::core ::CEngine ::sendMessageToPhysic (event ::CEngineEvent & e) [inherited]

envoie un message au moteur physique

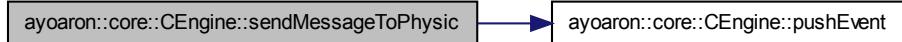
Paramètres

in	e	evenement à envoyer vers le moteur de physique
----	---	--

Cette méthode est utiliser pour envoyer un événement vers moteur de physique

Références ayoaron ::core ::CEngine ::phy, et ayoaron ::core ::CEngine ::pushEvent().

Voici le graphe d'appel pour cette fonction :



7.56.4.44 void ayoaron ::core ::CEngine ::sendMessageToSound (event ::CEngineEvent & e) [inherited]

envoie un message au moteur de son

Paramètres

in	e	evenement à envoyer vers le moteur de son
----	---	---

Cette méthode est utiliser pour envoyer un événement vers moteur de son

Références ayoaron ::core ::CEngine ::pushEvent(), et ayoaron ::core ::CEngine ::se.

Voici le graphe d'appel pour cette fonction :



7.56.4.45 void ayoaron ::reseau ::CNetworkEngine ::sendMulticast (event ::CEngineEvent & e)

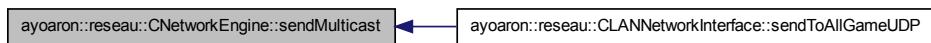
Références s_multicast, et sendEventUDP().

Référencé par ayoaron ::reseau ::CLANNetworkInterface ::sendToAllGameUDP().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.46 void ayoaron ::reseau ::CNetworkEngine ::serverThreadTCPReceive (boost ::asio ::ip ::tcp ::socket * soc) [private]

thread function each player is managed by its own socket and thread this management lasts as long as a game is running

Paramètres

soc	pointor to the socket
-----	-----------------------

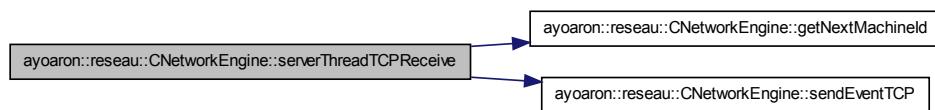
À faire

send password by ssl, so on another socket

Références ayoaron : :core : :CEngine : :ge, getNextMachineId(), l_machines, ayoaron : :event : :CEngineEvent : :MSG_NICK, ayoaron : :core : :CEngine : :ne, network_buffer, ayoaron : :event : :CEngineEvent : :NEW_PLAYER, ayoaron : :event : :CEngineEvent : :REFUSED, ayoaron : :event : :CEngineEvent : :s_data, sendEventTCP(), ayoaron : :event : :CEngineEvent : :type, et ayoaron : :event : :CEngineEvent : :WELCOME.

Référencé par handleAcceptTCP().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.4.47 void ayoaron ::reseau ::CNetworkEngine ::setPlayAs(int *i*) [inline]

Références play_as.

Référencé par ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal().

Voici le graphe d'appel pour cette fonction :



7.56.4.48 void ayoaron : :reseau : :CNetworkEngine : :threadSendMulticast (event : :CEngineEvent *gp*) [private]

thread function sends game proposition as long as the game has not started

Paramètres

gp game proposition to broadcast

Références io, s_multicast, sendEventUDP(), ayoaron ::core ::CEngine ::state, ayoaron ::core ::CEngine ::state_mutex, et ayoaron ::core ::CEngine ::STATE_WAITING_PLAYERS.

Référencé par broadcastGameProposition().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



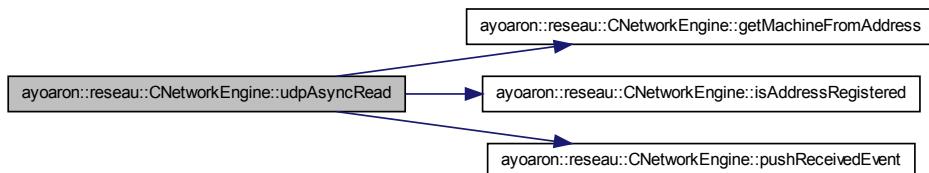
7.56.4.49 void ayoaron ::reseau ::CNetworkEngine ::udpAsyncRead (const boost ::system ::error_code & e, std ::size_t) [private]

main reception function. it receives data from UDP connection we only accept messages from registered machines asynchronous function

Références getMachineFromAddress(), isAddressRegistered(), ayoaron ::core ::CEngine ::ne, network_buffer, pushReceivedEvent(), et udp_remote_endpoint.

Référencé par getConfigData().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.56.5 Documentation des données membres

7.56.5.1 data : :CConfigFile ayoaron ::core ::CEngine ::cfile [protected, inherited]

Fichier de configuration.

Référencé par ayoaron ::audio ::CSoundEngine ::getConfigData(), getConfigData(), et ayoaron ::graphic ::CGraphicsEngine ::getConfigData().

7.56.5.2 std ::string ayoaron ::core ::CEngine ::engine_name [protected, inherited]

nom de ce moteur

Référencé par ayoaron ::core ::CGameEngine ::CGameEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et CNetworkEngine().

7.56.5.3 std ::queue<event ::CEngineEvent> ayoaron ::core ::CEngine ::events_queue [protected, inherited]

liste des événement à traiter

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEngine ::pushEvent().

7.56.5.4 CGameEngine* ayoaron ::core ::CEngine ::ge [protected, inherited]

pointeur vers le moteur de jeu

Référencé par ayoaron ::core ::CEngine ::attachGameEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getGameEngine(), ayoaron ::core ::CEngine ::sendMessageToGame(), et serverThreadTCPReceive().

7.56.5.5 graphic : :CGraphicsEngine* ayoaron ::core ::CEngine ::gffe [protected, inherited]

pointeur vers le moteur de rendu graphique

Référencé par ayoaron ::core ::CEngine ::attachGraphicsEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CGameEngine ::frame(), ayoaron ::core ::CEngine ::getGraphicsEngine(), ayoaron ::core ::CEngine ::sendMessageToGraphics(), et ayoaron ::core ::CGameEngine ::togglePause().

7.56.5.6 **boost ::asio ::io_service ayoaron ::reseau ::CNetworkEngine ::io**
[private]

Référencé par getConfigData(), handleAcceptTCP(), listenMulti3secondsUDP(), threadSendMulticast(), et ~CNetworkEngine().

7.56.5.7 **std ::vector<boost ::shared_ptr<CNetworkMachine> >**
ayoaron ::reseau ::CNetworkEngine ::l_machines

Référencé par clientConnectsToServer(), getMachineFromAddress(), getMachineFromMld(), getMachineFromPlayerName(), isAddressRegistered(), ayoaron ::core ::CGameRound ::makeFromCollect(), processReceivedEvents(), ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP(), ayoaron ::reseau ::CIinternetNetworkInterface ::sendToAllGameUDP(), et serverThreadTCPReceive().

7.56.5.8 **boost ::mutex ayoaron ::reseau ::CNetworkEngine ::l_receive_queue**
[private]

Référencé par processReceivedEventsQueue(), et pushReceivedEvent().

7.56.5.9 **std ::string ayoaron ::reseau ::CNetworkEngine ::multicast_address**
[private]

Référencé par getConfigData().

7.56.5.10 **reseau ::CNetworkEngine* ayoaron ::core ::CEngine ::ne**
[protected, inherited]

pointeur vers le moteur de réseau

Référencé par ayoaron ::core ::CEngine ::attachNetworkEngine(), ayoaron ::core ::CEngine ::CEngine(), ayoaron ::core ::CEngine ::getNetworkEngine(), sendEventTCP(), sendEventUDP(), ayoaron ::core ::CEngine ::sendMessageToNetwork(), serverThreadTCPReceive(), et udpAsyncRead().

7.56.5.11 **boost ::array<char, BUFFER_SIZE> ayoaron ::reseau ::CNetworkEngine ::network_buffer** [private]

Référencé par getConfigData(), multicastReceive(), serverThreadTCPReceive(), et udpAsyncRead().

7.56.5.12 **int ayoaron ::reseau ::CNetworkEngine ::next_machine_id**
[private]

Référencé par CNetworkEngine(), et getNextMachineId().

7.56.5.13 **boost ::shared_ptr<CNetworkInterface>**
ayoaron ::reseau ::CNetworkEngine ::ni

Référencé par buildInternetInterface(), buildLANInterface(), ayoaron ::core ::CGameRound ::CGameRound(), cleanInterface(), CNetworkEngine(), ayoaron ::core ::CGameRound ::makeFromCollect(), processReceivedEvents(), et ayoaron ::core ::CGameRound ::processReceivedEvents().

meRound ::~CGameRound().

7.56.5.14 CGame* ayoaron ::core ::CEngine ::parent [protected, inherited]

pointeur vers le parent

Référencé par ayoaron ::graphic ::CGraphicsEngine ::frame(), ayoaron ::core ::CEngine ::getParent(), et ayoaron ::graphic ::CGraphicsEngine ::requestClose().

7.56.5.15 bool ayoaron ::core ::CEngine ::passive_mode [protected, inherited]

true si mode passif activé

Référencé par ayoaron ::core ::CEngine ::CEngine(), ayoaron ::graphic ::CGraphicsEngine ::CGraphicsEngine(), et ayoaron ::audio ::CSoundEngine ::CSoundEngine().

7.56.5.16 physic ::CPhysicEngine* ayoaron ::core ::CEngine ::phy [protected, inherited]

pointeur vers le moteur de physique

Référencé par ayoaron ::core ::CEngine ::attachPhysicEngine(), ayoaron ::core ::CEngine ::CEngine(), et ayoaron ::core ::CEngine ::sendMessageToPhysic().

7.56.5.17 int ayoaron ::reseau ::CNetworkEngine ::play_as [private]

Référencé par playAsWhat(), processReceivedEvents(), et setPlayAs().

7.56.5.18 int ayoaron ::reseau ::CNetworkEngine ::port_multicast

Référencé par getConfigData().

7.56.5.19 int ayoaron ::reseau ::CNetworkEngine ::port_tcp_server_connection

Référencé par ayoaron ::reseau ::CNetworkMachine ::connect(), et getConfigData().

7.56.5.20 int ayoaron ::reseau ::CNetworkEngine ::port_udp_reception

Référencé par ayoaron ::reseau ::CNetworkMachine ::CNetworkMachine(), ayoaron ::reseau ::CNetworkMachine ::connect(), et getConfigData().

7.56.5.21 boost ::mutex ayoaron ::reseau ::CNetworkEngine ::propositions_mutex [private]

Référencé par listenMulti3secondsUDP(), et multicastReceive().

7.56.5.22 boost ::mutex ayoaron ::core ::CEngine ::queue_mutex [protected, inherited]

Mutex de protection de la liste d'événement event_queue.

Référencé par ayoaron ::core ::CEngine ::processQueue(), et ayoaron ::core ::CEn-

gine ::pushEvent().

7.56.5.23 `std ::queue<event ::CEngineEvent> ayoaron ::reseau ::CNetworkEngine ::received_events [private]`

Référencé par `multicastReceive()`, `processReceivedEventsQueue()`, et `pushReceivedEvent()`.

7.56.5.24 `std ::vector<event ::CEngineEvent> ayoaron ::reseau ::CNetworkEngine ::received_propositions [private]`

Référencé par `listenMulti3secondsUDP()`, et `multicastReceive()`.

7.56.5.25 `bool ayoaron ::reseau ::CNetworkEngine ::running [private]`

Référencé par `CNetworkEngine()`, `is_running()`, et `listenGamePropositions()`.

7.56.5.26 `boost ::asio ::ip ::udp ::socket* ayoaron ::reseau ::CNetworkEngine ::s_multicast [private]`

Référencé par `cleanInterface()`, `getConfigData()`, `sendMulticast()`, et `threadSendMulticast()`.

7.56.5.27 `boost ::asio ::ip ::tcp ::socket* ayoaron ::reseau ::CNetworkEngine ::s_tcp [private]`

Référencé par `cleanInterface()`, `getConfigData()`, et `getTCPSocket()`.

7.56.5.28 `boost ::asio ::ip ::udp ::socket* ayoaron ::reseau ::CNetworkEngine ::s_udp_in [private]`

Référencé par `cleanInterface()`, et `getConfigData()`.

7.56.5.29 `audio ::CSoundEngine* ayoaron ::core ::CEngine ::se [protected, inherited]`

pointeur vers le moteur de son

Référencé par `ayoaron ::core ::CEngine ::attachSoundEngine()`, `ayoaron ::core ::CEngine ::CEngine()`, `ayoaron ::core ::CEngine ::getSoundEngine()`, et `ayoaron ::core ::CEngine ::sendMessageToSound()`.

7.56.5.30 `int ayoaron ::core ::CEngine ::state [inherited]`

l'état actuel

Référencé par `broadcastGameProposition()`, `CNetworkEngine()`, `ayoaron ::graphic ::CGuiSystem ::displayLoading()`, `ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal()`, `ayoaron ::physic ::CSecteur ::frame()`, `ayoaron ::core ::CGameEngine ::frame()`, `handleAcceptTCP()`, `listenMulti3secondsUDP()`, `multicastReceive()`, `ayoaron ::graphic ::CGraphicsEngine ::setState()`, `ayoaron ::graphic ::CGuiSystem ::showErrorMessage()`, `threadSendMulticast()`, et `~CNetworkEngine()`.

7.56.5.31 `boost ::mutex ayoaron ::core ::CEngine ::state_mutex` [inherited]

Mutex de protection de l'état.

Référencé par `broadcastGameProposition()`, `ayoaron ::graphic ::CGuiSystem ::displayLoading()`, `ayoaron ::graphic ::CGuiSystem ::displayMenuPrincipal()`, `listenMulti3secondsUDP()`, `ayoaron ::graphic ::CGuiSystem ::showErrorMessage()`, `threadSendMulticast()`, et `~CNetworkEngine()`.

7.56.5.32 `boost ::asio ::ip ::tcp ::acceptor* ayoaron ::reseau ::CNetworkEngine ::tcp_acceptor` [private]

Référencé par `CNetworkEngine()`, `getConfigData()`, et `handleAcceptTCP()`.

7.56.5.33 `boost ::asio ::ip ::udp ::endpoint ayoaron ::reseau ::CNetworkEngine ::udp_remote_endpoint` [private]

Référencé par `getConfigData()`, `multicastReceive()`, et `udpAsyncRead()`.

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

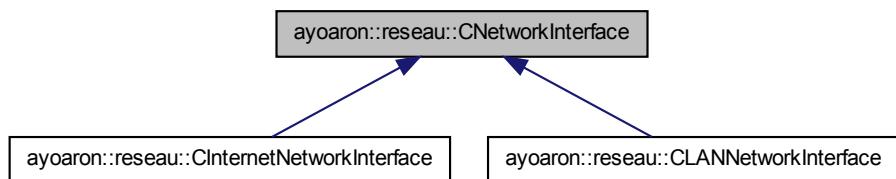
- [CNetworkEngine.h](#)
- [CGalaxie.cpp](#)
- [CNetworkEngine.cpp](#)

7.57 Référence de la classe ayoaron ::reseau ::CNetworkInterface

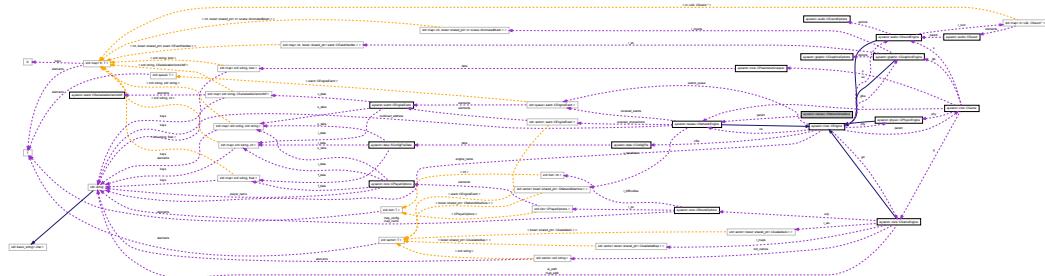
interface réseau

```
#include <CNetworkInterface.h>
```

Graphe d'héritage de `ayoaron ::reseau ::CNetworkInterface` :



Graphe de collaboration de ayoaron ::reseau ::CNetworkInterface :



Types publics

- enum { **LAN, INTERNET** }
type d'interface

Fonctions membres publiques

- **CNetworkInterface (CNetworkEngine *n)**
constructeur
- int **getType ()**
revoie le type d'interface
- void **sendAdd (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendMove (int, event : :CSerializableVector3df, event : :CSerializableVector3df, int)**
- void **sendRem (int, int)**
- void **sendToAllGameTCP (event : :CEngineEvent &, int)**
envoie du serveur vers tous les clients
- virtual void **sendToAllGameUDP (event : :CEngineEvent &e, int exp)=0**
envoie à tout le monde
- void **sendToServerTCP (event : :CEngineEvent &)**
envoie du client vers le serveur

Attributs protégés

- **CNetworkEngine * parent**
le moteur réseau parent
- int **type**
type d'interface

7.57.1 Description détaillée

interface réseau

Classe abstraite traitant les envoies vers le serveur ou les autres joueurs cette classe est instantiable en choisissant LAN ou internet dans le menu

7.57.2 Documentation des énumérations membres

7.57.2.1 anonymous enum

type d'interface

Valeurs énumérées :

LAN

INTERNET

7.57.3 Documentation des constructeurs et destructeur

7.57.3.1 ayoaron ::reseau ::CNetworkInterface ::CNetworkInterface (**CNetworkEngine** * *n*) [inline]

constructeur

Paramètres

in	<i>n</i>	le moteur réseau parent
----	----------	-------------------------

7.57.4 Documentation des fonctions membres

7.57.4.1 int ayoaron ::reseau ::CNetworkInterface ::getType () [inline]

revoie le type d'interface

Renvoie

le type d'interface

Références type.

7.57.4.2 void ayoaron ::reseau ::CNetworkInterface ::sendAdd (int *id*, event : :CSerializableVector3df *pos*, event : :CSerializableVector3df *rot*, int *expeditor*)

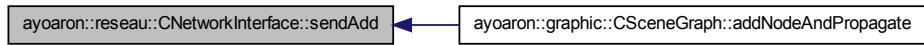
Références ayoaron ::event ::CEngineEvent ::ADD_NODE, ayoaron ::event ::CEngineEvent ::i_data, sendToAllGameTCP(), ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::addNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.57.4.3 `void ayoaron ::reseau ::CNetworkInterface ::sendMove (int id,
event : :CSerializableVector3df pos, event : :CSerializableVector3df rot,
int expeditor)`

Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::MOVE_NODE, sendToAllGameUDP(), ayoaron ::event ::CEngineEvent ::type, et ayoaron ::event ::CEngineEvent ::v_data.

Référencé par ayoaron ::graphic ::CSceneGraph ::moveNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.57.4.4 `void ayoaron ::reseau ::CNetworkInterface ::sendRem (int id, int expeditor)`

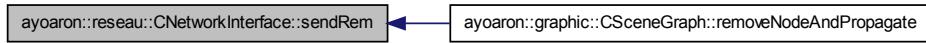
Références ayoaron ::event ::CEngineEvent ::i_data, ayoaron ::event ::CEngineEvent ::REM_NODE, sendToAllGameTCP(), et ayoaron ::event ::CEngineEvent ::type.

Référencé par ayoaron ::graphic ::CSceneGraph ::removeNodeAndPropagate().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.57.4.5 void ayoaron ::reseau ::CNetworkInterface ::sendToAllGameTCP (event : :CEngineEvent & e, int exp)

envoie du serveur vers tous les clients

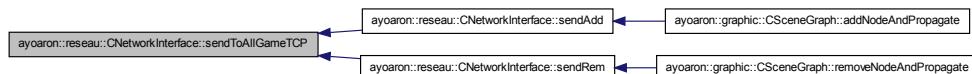
Paramètres

in	e	l'événement à envoyer
in	exp	la machine à qui il ne faut pas envoyer

Références ayoaron ::reseau ::CNetworkEngine ::l_machines, et parent.

Référencé par sendAdd(), et sendRem().

Voici le graphe d'appel pour cette fonction :



7.57.4.6 virtual void ayoaron ::reseau ::CNetworkInterface ::sendToAllGameUDP (event : :CEngineEvent & e, int exp) [pure virtual]

envoie à tout le monde

Paramètres

in	e	l'événement à envoyer
in	exp	la machine à qui il ne faut pas envoyer

Implémenté dans [ayoaron ::reseau ::CIinternetNetworkInterface](#), et [ayoaron ::reseau ::CLANNetworkInterface](#).

Référencé par `sendMove()`.

Voici le graphe d'appel pour cette fonction :



7.57.4.7 void ayoaron ::reseau ::CNetworkInterface ::sendToServerTCP (event ::CEngineEvent & e)

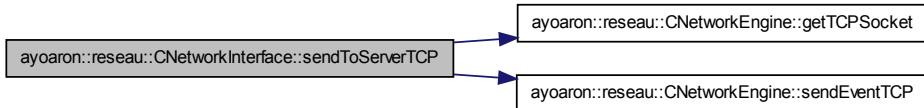
envoie du client vers le serveur

Paramètres

in	e	l'événement à envoyer
----	---	-----------------------

Références `ayoaron ::reseau ::CNetworkEngine ::getTCPSocket()`, parent, et `ayoaron ::reseau ::CNetworkEngine ::sendEventTCP()`.

Voici le graphe d'appel pour cette fonction :



7.57.5 Documentation des données membres

7.57.5.1 CNetworkEngine* ayoaron ::reseau ::CNetworkInterface ::parent [protected]

le moteur réseau parent

Référencé par `sendToAllGameTCP()`, `ayoaron ::reseau ::CLANNetworkInterface ::sendToAllGameUDP()`, `ayoaron ::reseau ::CIinternetNetworkInterface ::sendToAllGameUDP()`, et `sendToServerTCP()`.

7.57.5.2 int ayoaron ::reseau ::CNetworkInterface ::type [protected]

type d'interface

Référencé par ayoaron ::reseau ::ClinternetNetworkInterface ::ClinternetNetworkInterface(), ayoaron ::reseau ::CLANNetworkInterface ::CLANNetworkInterface(), et getType().

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

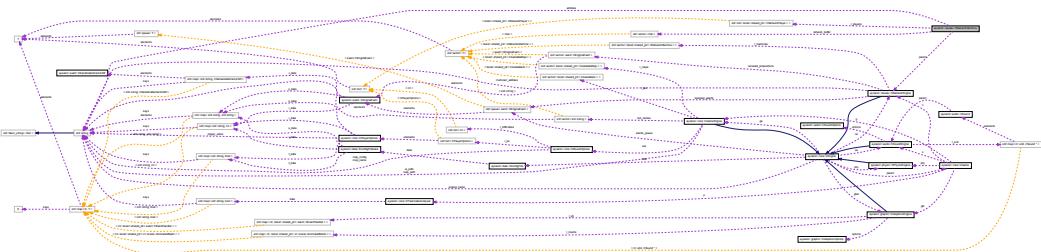
- [CNetworkInterface.h](#)
- [CNetworkInterface.cpp](#)

7.58 Référence de la classe ayoaron ::reseau ::CNetworkMachine

une machine réseau

```
#include <CNetworkMachine.h>
```

Graphe de collaboration de ayoaron ::reseau ::CNetworkMachine :



Fonctions membres publiques

- void [attachPlayer](#) (boost ::shared_ptr< CNetworkPlayer >)

associe un joueur à la machine
- [CNetworkMachine \(\)](#)

constructeur
- [CNetworkMachine \(CNetworkEngine *, boost ::asio ::ip ::tcp ::socket *, int\)](#)

constructeur
- [CNetworkMachine \(CNetworkEngine *, int\)](#)

constructeur
- int [connect](#) (std ::string &, std ::string &, std ::string &)

connect à la machine
- std ::string & [getAddress](#) ()

revoie l'adresse de la machine
- int [getId](#) ()

revoie le numéro de la machine
- bool [hasNick](#) (std ::string &)

test la présence joueur par son nom
- void [removePlayer](#) (std ::string &)

supprime un joueur de la machine
- void [removePlayer](#) (boost ::shared_ptr< CNetworkPlayer >)

supprime un joueur de la machine
- void [tcpAsyncReceive](#) (const boost ::system ::error_code &, size_t)

fonction de réception TCP asynchrone
- void [tcpSend](#) (event ::CEngineEvent &)

envoie un événement par TCP
- void [tcpThreadSend](#) (event ::CEngineEvent &)
- void [udpSend](#) (event ::CEngineEvent &)

- `envoie un événement par UDPP`
- `~CNetworkMachine ()`
destructeur

Attributs publics

- `std ::list< boost ::shared_ptr< CNetworkPlayer > > l_players`
liste des joueur, Ai ou humain
- `boost ::asio ::ip ::tcp ::socket * s_tcp`
socket tcp
- `boost ::asio ::ip ::udp ::socket * s_udp`
socket udp

Attributs privés

- `std ::string address`
adresse de la machine
- `int id`
numéro de la machine
- `boost ::asio ::io_service io`
service d'entrée/sortie
- `std ::vector< char > network_buffer`
buffer de transfert réseau
- `CNetworkEngine * parent`
moteur réseau parent
- `boost ::asio ::ip ::udp ::endpoint * udp_ep`
point d'arrêt udp

7.58.1 Description détaillée

une machine réseau

toutes les machines prennant part au jeu est référencée dans une [CNetworkMachine](#)

7.58.2 Documentation des constructeurs et destructeur

7.58.2.1 ayoaron ::reseau ::CNetworkMachine ::CNetworkMachine ()

constructeur

Constructeur par defaut ne fait rien de spéciale

Références address, s_tcp, s_udp, et udp_ep.

7.58.2.2 ayoaron ::reseau ::CNetworkMachine ::CNetworkMachine (CNetworkEngine * p, boost ::asio ::ip ::tcp ::socket * so, int i)

constructeur

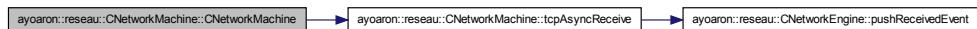
Paramètres

<code>p</code>	pointeur vers le moteur réseau parent
<code>so</code>	le socket tcp de la machine
<code>i</code>	le numéro de la machine

constructeur utilisé massivement

Références address, io, network_buffer, parent, ayoaron ::reseau ::CNetworkEngine ::port_udp_reception, s_udp, tcpAsyncReceive(), et udp_ep.

Voici le graphe d'appel pour cette fonction :



7.58.2.3 ayoaron ::reseau ::CNetworkMachine ::CNetworkMachine (CNetworkEngine * p, int i)

constructeur

Paramètres

<i>p</i>	pointeur vers le moteur réseau parent
<i>i</i>	le numéro de la machine

Références parent.

7.58.2.4 ayoaron ::reseau ::CNetworkMachine ::~CNetworkMachine ()

destructeur

Références io.

7.58.3 Documentation des fonctions membres

7.58.3.1 void ayoaron ::reseau ::CNetworkMachine ::attachPlayer (boost ::shared_ptr< CNetworkPlayer > p)

associe un joueur à la machine

Paramètres

in	<i>p</i>	le joueur à associer
----	----------	----------------------

Ajoute un joueur réseau à la machine

Références I_players.

7.58.3.2 int ayoaron ::reseau ::CNetworkMachine ::connect (std ::string & host, std ::string & nick, std ::string & pass)

connect à la machine

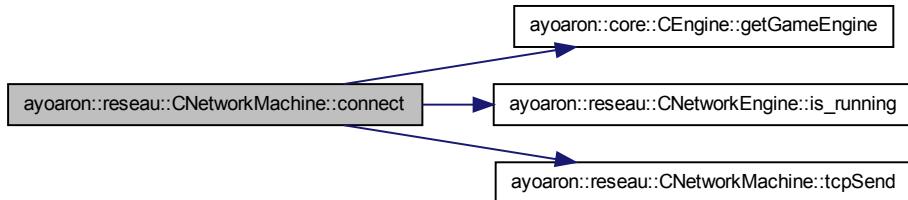
Paramètres

in	<i>host</i>	l'adresse de la machine à connecter
----	-------------	-------------------------------------

in	<i>nick</i>	le nom d'utilisateur
in	<i>pass</i>	le mot de passe

Références ayoaron ::event ::CEngineEvent ::CHOSEN_MAP, ayoaron ::core ::CEngine ::getGameEngine(), io, ayoaron ::reseau ::CNetworkEngine ::is_running(), ayoaron ::event ::CEngineEvent ::MSG_NICK, network_buffer, parent, ayoaron ::reseau ::CNetworkEngine ::port_tcp_server_connection, ayoaron ::reseau ::CNetworkEngine ::port_udp_reception, ayoaron ::event ::CEngineEvent ::REFUSED, ayoaron ::event ::CEngineEvent ::s_data, s_tcp, s_udp, tcpSend(), ayoaron ::event ::CEngineEvent ::type, udp_ep, et ayoaron ::event ::WELCOME.

Voici le graphe d'appel pour cette fonction :



7.58.3.3 std ::string& ayoaron ::reseau ::CNetworkMachine ::getAddress() [inline]

revoie l'adresse de la machine

Références address.

7.58.3.4 int ayoaron ::reseau ::CNetworkMachine ::getId() [inline]

revoie le numéro de la machine

Références id.

7.58.3.5 bool ayoaron ::reseau ::CNetworkMachine ::hasNick(std ::string & n)

test la présence joueur par son nom

Paramètres

in	<i>n</i>	nom du joueur à trouver
----	----------	-------------------------

renvoie vrai si il y a un joueur de nom n

Références l_players.

7.58.3.6 void ayoaron ::reseau ::CNetworkMachine ::removePlayer (boost ::shared_ptr< CNetworkPlayer > *p*)

supprime un joueur de la machine

Paramètres

in	<i>p</i>	le joueur à supprimer
----	----------	-----------------------

supprime un joueur réseau de la machine

Références *l_players*.

7.58.3.7 void ayoaron ::reseau ::CNetworkMachine ::removePlayer (std ::string & *s*)

supprime un joueur de la machine

Paramètres

in	<i>s</i>	le joueur à supprimer
----	----------	-----------------------

Références *l_players*.

7.58.3.8 void ayoaron ::reseau ::CNetworkMachine ::tcpAsyncReceive (const boost ::system ::error_code & *e*, size_t *bytes_received*)

fonction de réception TCP asynchrone

Paramètres

in	<i>e</i>	
in	<i>bytes_received</i>	fonction de réception TCP asynchrone tourne seul et renvoie l'événement reçu à la machine réseau

Références ayoaron ::event ::CEngineEvent ::i_data, id, network_buffer, parent, ayoaron ::reseau ::CNetworkEngine ::pushReceivedEvent(), ayoaron ::event ::CEngineEvent ::REMOVE_MACHINE, et ayoaron ::event ::CEngineEvent ::type.

Référencé par CNetworkMachine().

Voici le graphe d'appel pour cette fonction :



Voici le graphe d'appel pour cette fonction :



7.58.3.9 void ayoaron ::reseau ::CNetworkMachine ::tcpSend (event ::CEngineEvent & e)

envoie un événement par TCP

Paramètres

in	e	l'événement à envoyer
----	---	-----------------------

Références s_tcp.

Référencé par connect().

Voici le graphe d'appel pour cette fonction :



7.58.3.10 void ayoaron ::reseau ::CNetworkMachine ::tcpThreadSend (event ::CEngineEvent &)

7.58.3.11 void ayoaron ::reseau ::CNetworkMachine ::udpSend (event ::CEngineEvent & e)

envoie un événement par UDPP

Paramètres

in	e	l'événement à envoyer
----	---	-----------------------

Références s_udp.

7.58.4 Documentation des données membres

7.58.4.1 **std ::string ayoaron ::reseau ::CNetworkMachine ::address**
[private]

adresse de la machine

Référencé par CNetworkMachine(), et getAddress().

7.58.4.2 **int ayoaron ::reseau ::CNetworkMachine ::id** [private]

numéro de la machine

Référencé par getId(), et tcpAsyncReceive().

7.58.4.3 **boost ::asio ::io_service ayoaron ::reseau ::CNetworkMachine ::io**
[private]

service d'entrée/sortie

Référencé par CNetworkMachine(), connect(), et ~CNetworkMachine().

7.58.4.4 **std ::list<boost ::shared_ptr<CNetworkPlayer>> ayoaron ::reseau ::CNetworkMachine ::l_players**

liste des joueur, Ai ou humain

Référencé par attachPlayer(), hasNick(), et removePlayer().

7.58.4.5 **std ::vector<char> ayoaron ::reseau ::CNetworkMachine ::network_buffer** [private]

buffer de transfert réseau

Référencé par CNetworkMachine(), connect(), et tcpAsyncReceive().

7.58.4.6 **CNetworkEngine* ayoaron ::reseau ::CNetworkMachine ::parent**
[private]

moteur réseau parent

Référencé par CNetworkMachine(), connect(), et tcpAsyncReceive().

7.58.4.7 **boost ::asio ::ip ::tcp ::socket* ayoaron ::reseau ::CNetworkMachine ::s_tcp**

socket tcp

Référencé par CNetworkMachine(), connect(), et tcpSend().

7.58.4.8 **boost ::asio ::ip ::udp ::socket* ayoaron ::reseau ::CNetworkMachine ::s_udp**

socket udp

Référencé par CNetworkMachine(), connect(), et udpSend().

7.58.4.9 boost ::asio ::ip ::udp ::endpoint* ayoaron ::reseau ::CNetworkMachine ::udp_ep [private]

point d'arrêt udp

Référencé par CNetworkMachine(), et connect().

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

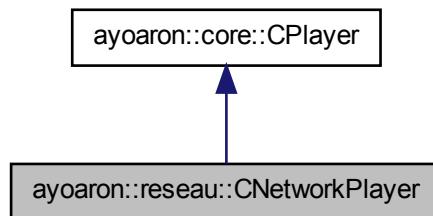
- [CNetworkMachine.h](#)
- [CNetworkMachine.cpp](#)

7.59 Référence de la classe ayoaron ::reseau ::CNetworkPlayer

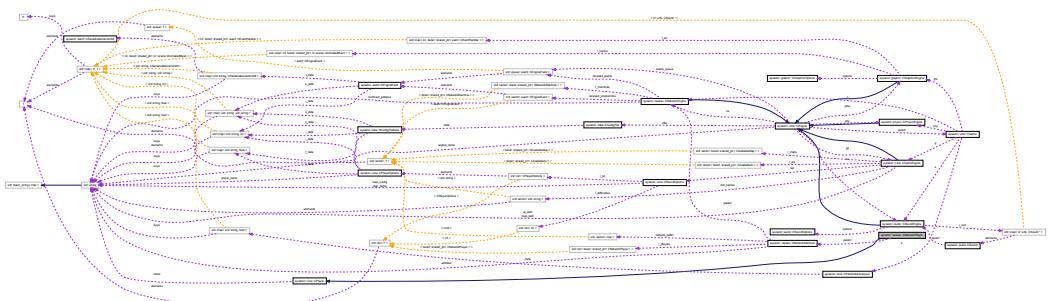
l'objet joueur réseau

```
#include <CNetworkPlayer.h>
```

Graphe d'héritage de ayoaron ::reseau ::CNetworkPlayer :



Graphe de collaboration de ayoaron ::reseau ::CNetworkPlayer :



Fonctions membres publiques

- **CNetworkPlayer ()**
constructeur
- **CNetworkPlayer (core ::CPlayerOptions *po)**
constructeur ai
- **CNetworkPlayer (std ::string &s)**
constructeur humain
- **std ::string & getName ()**
renvoie le nom du joueur
- **void setParent (CNetworkMachine *p)**
definit la machine réseau associée

Attributs privés

- **CNetworkMachine * parent**
la machine réseau parente
- **boost ::asio ::ip ::tcp ::socket * s_tcp**
socket tcp

7.59.1 Description détaillée

l'objet joueur réseau

chaque joueur réseau est géré par une instance de **CNetworkPlayer** les joueur réseau ne peuvent exister sans une machine réseau associée

7.59.2 Documentation des constructeurs et destructeur**7.59.2.1 ayoaron ::reseau ::CNetworkPlayer ::CNetworkPlayer ()**

constructeur

7.59.2.2 ayoaron ::reseau ::CNetworkPlayer ::CNetworkPlayer (core ::CPlayerOptions * po)

constructeur ai

Paramètres

in	po	les option de joueur
----	----	----------------------

7.59.2.3 ayoaron ::reseau ::CNetworkPlayer ::CNetworkPlayer (std ::string & s)

constructeur humain

Paramètres

in	s	nom du joueur
----	---	---------------

7.59.3 Documentation des fonctions membres

7.59.3.1 `std::string& ayoaron::core::CPlayer::getName()` [inline, inherited]

renvoie le nom du joueur

Références `ayoaron::core::CPlayer::name`.

7.59.3.2 `void ayoaron::reseau::CNetworkPlayer::setParent(CNetworkMachine* p)`

definit la machine réseau associée

Paramètres

in	<code>p</code>	la machine réseau associée
----	----------------	----------------------------

Références parent.

7.59.4 Documentation des données membres

7.59.4.1 `CNetworkMachine* ayoaron::reseau::CNetworkPlayer::parent` [private]

la machine réseau parente

Référencé par `setParent()`.

7.59.4.2 `boost::asio::ip::tcp::socket* ayoaron::reseau::CNetworkPlayer::s_tcp` [private]

socket tcp

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

- [CNetworkPlayer.h](#)
- [CNetworkPlayer.cpp](#)

7.60 Référence de la classe game_round

classe de gestion de la partie

```
#include <CGameRound.h>
```

7.60.1 Description détaillée

classe de gestion de la partie

La documentation de cette classe a été générée à partir du fichier suivant :

- [CGameRound.h](#)

7.61 Référence de la classe round_options

rassemblement d'option de partie

```
#include <CRoundOptions.h>
```

7.61.1 Description détaillée

rassemblement d'option de partie

La documentation de cette classe a été générée à partir du fichier suivant :

- [CRoundOptions.h](#)

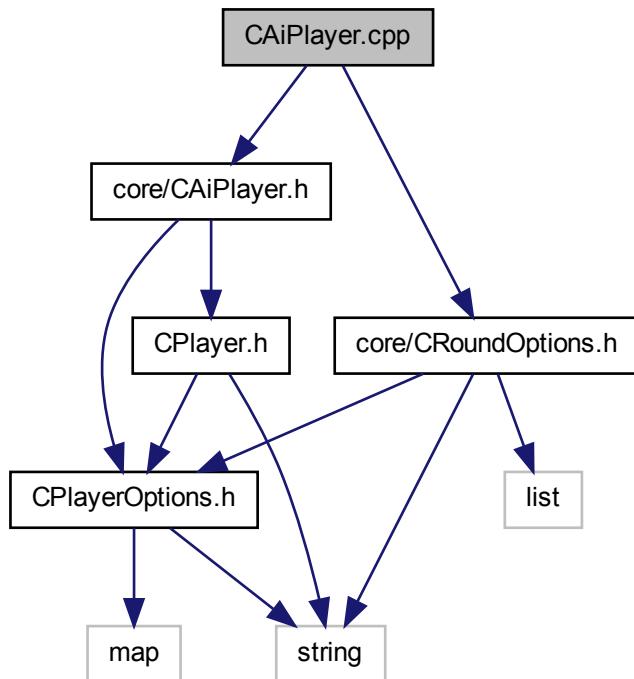
8 Documentation des fichiers

8.1 Référence du fichier CAiPlayer.cpp

Implémentation de la classe CAiPlayer.

```
#include "core/CAiPlayer.h"  
#include "core/CRoundOptions.h"
```

Graphe des dépendances par inclusion de CAiPlayer.cpp :



Espaces de nommage

- namespace `ayoaron`
l'espace de nommage de base du projet ayoaron
- namespace `ayoaron::core`
espace de nommage contenant tous les objets de base

8.1.1 Description détaillée

Implémentation de la classe CAiPlayer.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

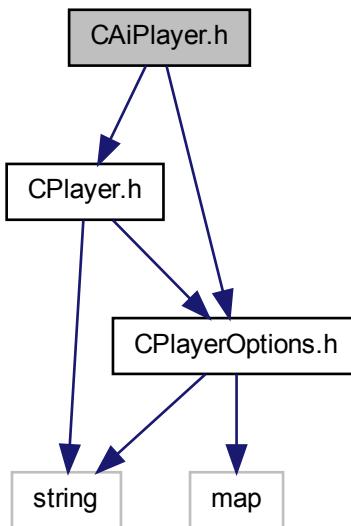
Implémentation de la classe CAiPlayer

8.2 Référence du fichier CAiPlayer.h

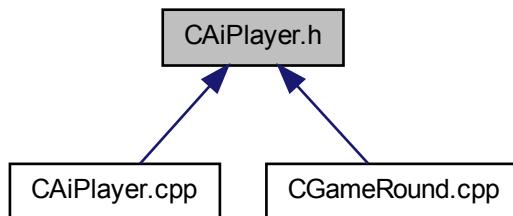
Déclaration de la classe CAiPlayer.

```
#include "CPlayer.h"  
#include "CPlayerOptions.h"
```

Graphe des dépendances par inclusion de CAiPlayer.h :



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



Classes

- class [ayoaron](#) : [:core](#) : [CAiPlayer](#)
les pnj

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:core](#)
espace de nomage contenant tous les objets de base

8.2.1 Description détaillée

Déclaration de la classe CAiPlayer.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

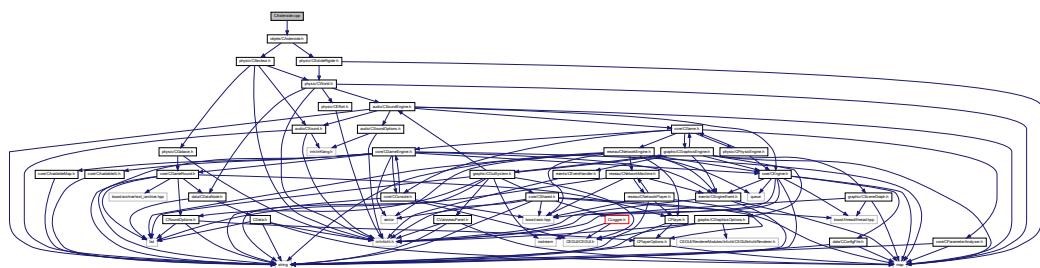
Déclaration de la classe CAiPlayer

8.3 Référence du fichier CAsteroide.cpp

fichier de description d'un astéroïde

```
#include "objets/CAsteroide.h"
```

Graphe des dépendances par inclusion de CAsteroide.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::objets**

8.3.1 Description détaillée

fichier de description d'un astéroïde

Date

4 juin 2011

Auteur

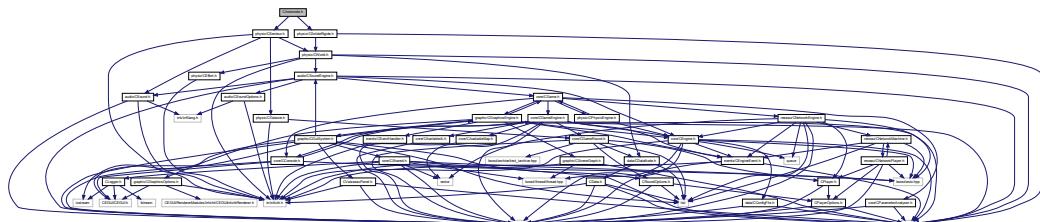
dla

8.4 Référence du fichier CAsteroide.h

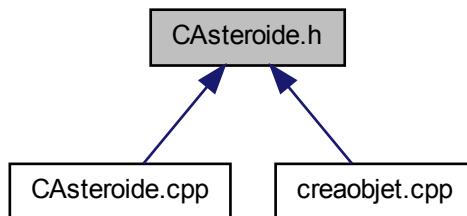
fichier de description d'un astéroïde

```
#include "physic/CSecteur.h"
#include "physic/CSolideRigide.h"
```

Graphe des dépendances par inclusion de CAsteroide.h :



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



Classes

- class [ayoaron](#) : [objets](#) : [CAsteroide](#)
Une classe de description d'un astéroïde.

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) : [objets](#)

8.4.1 Description détaillée

fichier de description d'un astéroïde

Date

4 juin 2011

Auteur

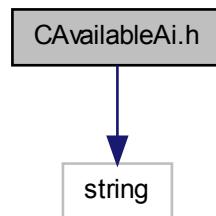
dla

8.5 Référence du fichier CAvailableAi.h

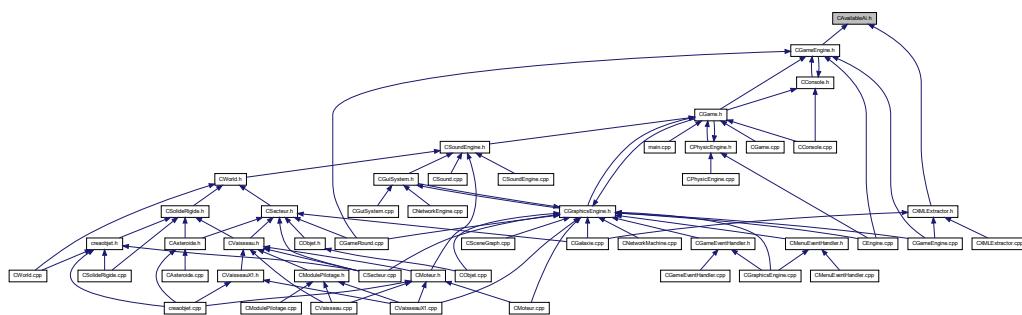
Déclaration de la classe CAvailableAi.

```
#include <string>
```

Graphe des dépendances par inclusion de CAvailableAi.h :



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



Classes

- class `ayoaron` : `:core :CAvailableAi`
contient le nom du script

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :core**
espace de nomage contenant tous les objets de base

8.5.1 Description détaillée

Déclaration de la classe CAvailableAi.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

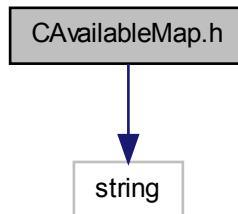
Déclaration de la classe CAvailableAi

8.6 Référence du fichier CAvailableMap.h

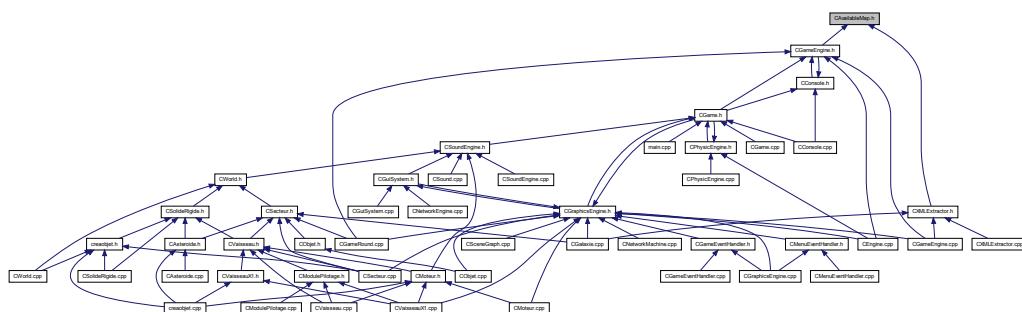
Déclaration de la classe CAvailableMap.

```
#include <string>
```

Graphe des dépendances par inclusion de CAvailableMap.h :



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



Classes

- class [ayoaron](#) : [:core](#) : [CAvailableMap](#)
La liste des maps.

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:core](#)
espace de nomage contenant tous les objets de base

8.6.1 Description détaillée

Déclaration de la classe CAvailableMap.

Auteur

Argawaen

Version

0.1

Date

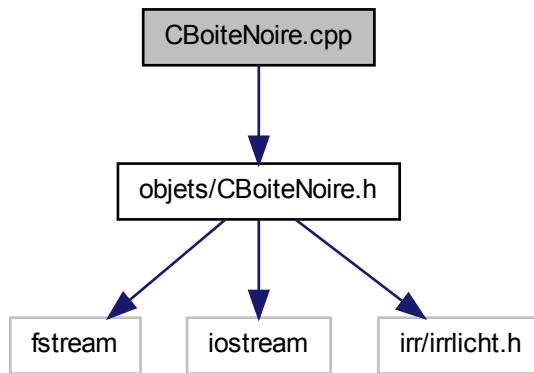
16 février 2011

Déclaration de la classe CAvailableMap

8.7 Référence du fichier CBoiteNoire.cpp

```
#include "objets/CBoiteNoire.h"
```

Graphe des dépendances par inclusion de CBoiteNoire.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:objets](#)

8.7.1 Description détaillée

Date

19 juil. 2011

Auteur

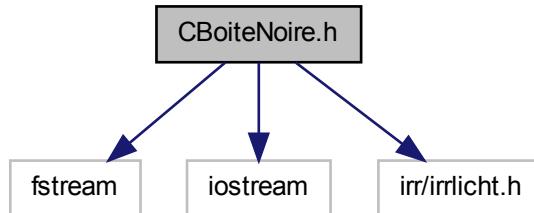
argawaen

8.8 Référence du fichier CBoiteNoire.h

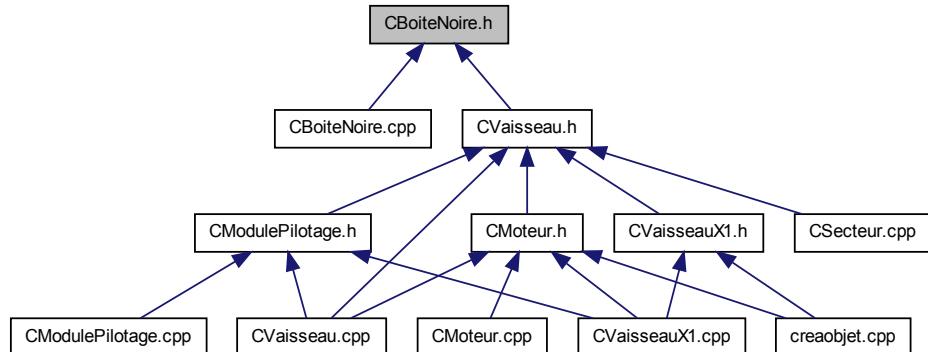
un élément essentiel de vaisseau : la boîte noire

```
#include <fstream>
#include <iostream>
#include <irr/irrlicht.h>
```

Graphe des dépendances par inclusion de CBoiteNoire.h :



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



Classes

- class `ayoaron` : `:objets` : `:CBoiteNoire`
l'élément de boite-noire

Espaces de nommage

- namespace `ayoaron`
l'espace de nomage de base du projet ayoaron
- namespace `ayoaron` : `:objets`

8.8.1 Description détaillée

un élément essentiel de vaisseau : la boîte noire

Date

19 juil. 2011

Auteur

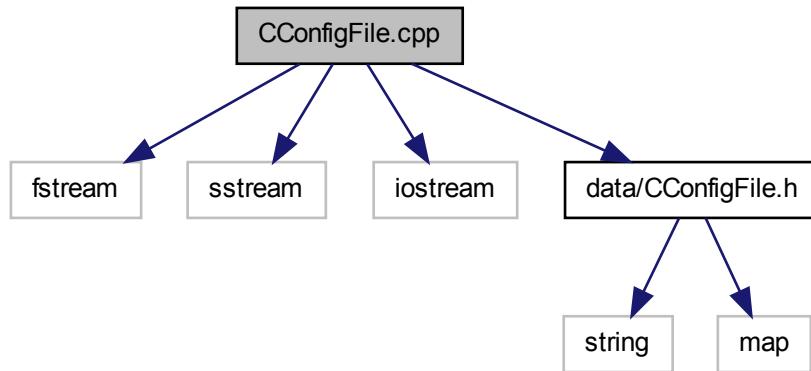
argawaen

8.9 Référence du fichier CConfigFile.cpp

Implémentation de la classe CConfigFile.

```
#include <fstream>
#include <sstream>
#include <iostream>
#include "data/CConfigFile.h"
```

Graphe des dépendances par inclusion de CConfigFile.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) : [:data](#)

8.9.1 Description détaillée

Implémentation de la classe CConfigFile.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

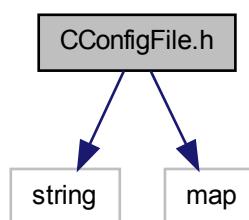
Implémentation de la classe CConfigFile et de la structure SConfigFileData

8.10 Référence du fichier CConfigFile.h

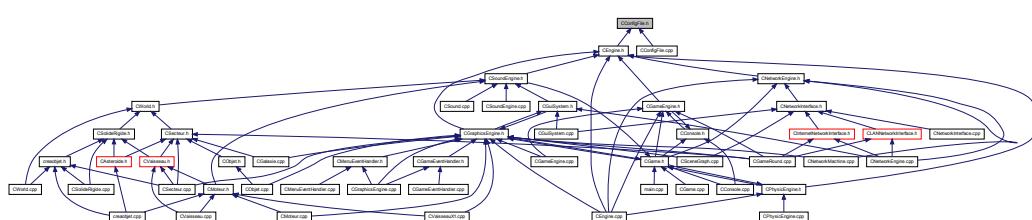
Déclaration de la classe CConfigFile.

```
#include <string>
```

```
#include <map>
```



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



Classes

- class **ayoaron** : :data : :CConfigFile

Traitement des fichiers de config.
- struct **ayoaron** : :data : :SConfigFileData

conteneur de donnée fichier config

Espaces de nommage

- namespace **ayoaron**

l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : :data

8.10.1 Description détaillée

Déclaration de la classe CConfigFile.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

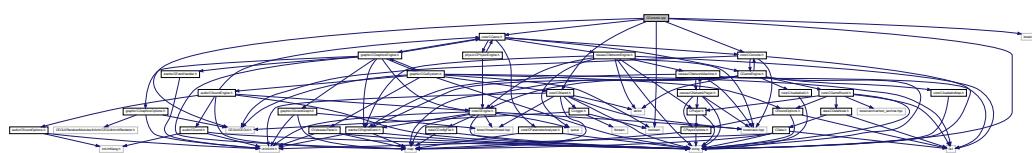
Déclaration de la classe CConfigFile et de la structure SConfigFileData

8.11 Référence du fichier CConsole.cpp

Implémentation de la classe CConsole.

```
#include <CEGUI/CEGUI.h>
#include <boost/bind.hpp>
#include <iostream>
#include <string>
#include "core/CConsole.h"
#include "core/CShared.h"
#include "core/CGame.h"
```

Graphe des dépendances par inclusion de CConsole.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base

8.11.1 Description détaillée

Implémentation de la classe CConsole.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

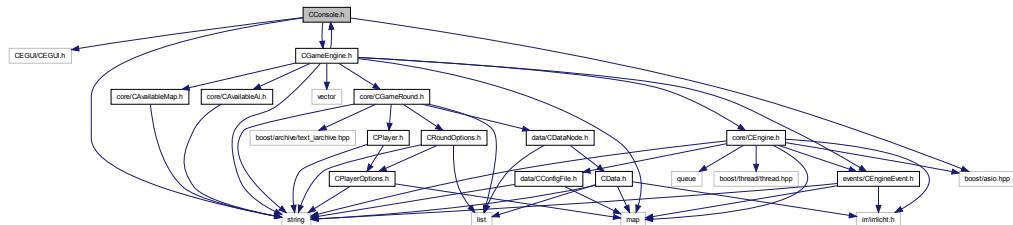
Implémentation de la classe CConsole

8.12 Référence du fichier CConsole.h

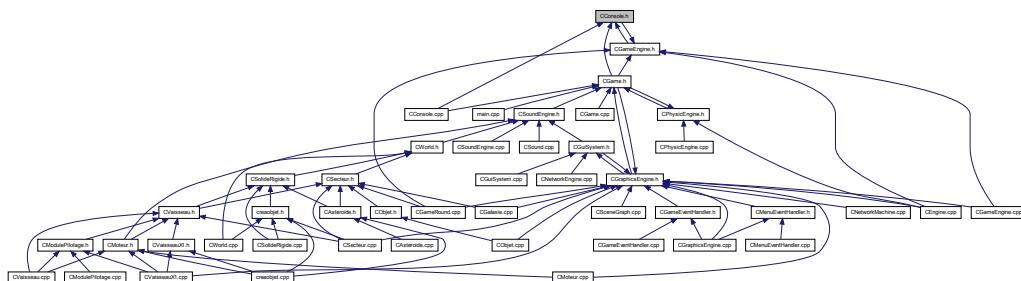
Déclaration de la classe CConsole.

```
#include <CEGUI/CEGUI.h>
#include <string>
#include <boost/asio.hpp>
#include "CGameEngine.h"
```

Graphe des dépendances par inclusion de CConsole.h :



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



Classes

- class ayoaron ::core ::CConsole
classe contenant la *CConsole* à faire apparaître en jeu

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron :core**
espace de nomage contenant tous les objets de base

8.12.1 Description détaillée

Déclaration de la classe CConsole.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

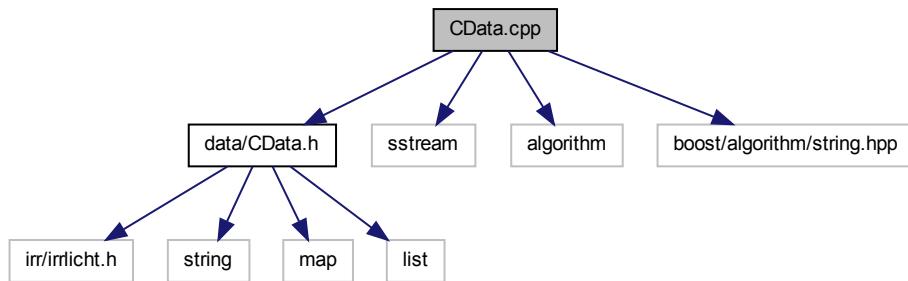
Déclaration de la classe CConsole

8.13 Référence du fichier CData.cpp

```
#include "data/CData.h"  
#include <sstream>  
#include <algorithm>
```

```
#include <boost/algorithm/string.hpp>
```

Graphe des dépendances par inclusion de CData.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:data](#)

Fonctions

- DATATYPE [ayoaron](#) : [:data](#) : [:str2dtyp](#) (std : [:string](#) str)
- std : [:wstring](#) [ayoaron](#) : [:data](#) : [:strtowchar](#) (const std : [:string](#))
*fonction de conversion d'un std : [:string](#) en wchar_t**
- const std : [:string](#) [ayoaron](#) : [:data](#) : [:wchartostr](#) (const wchar_t *)
fonction de conversion entre un wchar_t et un std : [:string](#)*

8.13.1 Description détaillée

Date

24 juil. 2011

Auteur

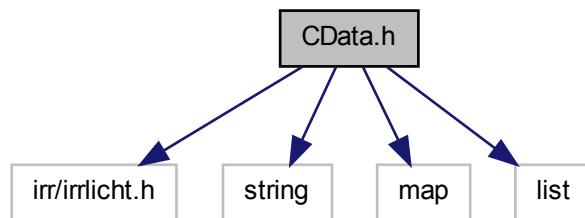
argawaen

8.14 Référence du fichier CData.h

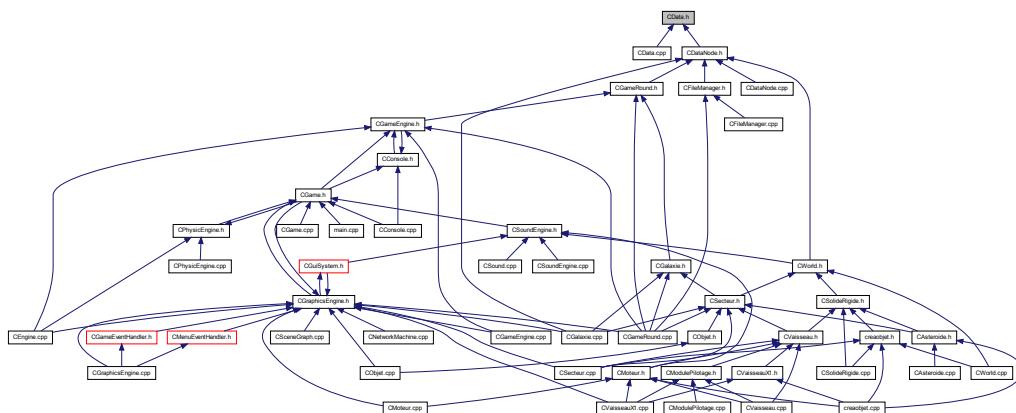
```
#include <irr/irrlicht.h>
#include <string>
#include <map>
```

```
#include <list>
```

Graphe des dépendances par inclusion de CData.h :



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



Classes

- class **ayoaron** : **:data** : **:CData**
classe de manipulation de données
 - struct **ayoaron** : **:data** : **:CDataStructure**
rassemblement des structures de données

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron** :**:data**

Énumérations

- enum ayoaron ::data ::DATATYPE {
 ayoaron ::data ::NOTYPE, ayoaron ::data ::UINT, ayoaron ::data ::SINT, ayoaron ::data ::FLOAT,
 ayoaron ::data ::VECTOR, ayoaron ::data ::MATRIX, ayoaron ::data ::COLOR,
 ayoaron ::data ::BOOL,
 ayoaron ::data ::STRING }

Fonctions

- DATATYPE ayoaron ::data ::str2dtyp (std ::string str)
- std ::wstring ayoaron ::data ::strtowchar (const std ::string)
*fonction de conversion d'un std ::string en wchar_t**
- const std ::string ayoaron ::data ::wchartostr (const wchar_t *)
fonction de conversion entre un wchar_t et un std ::string*

8.14.1 Description détaillée

Date

24 juil. 2011

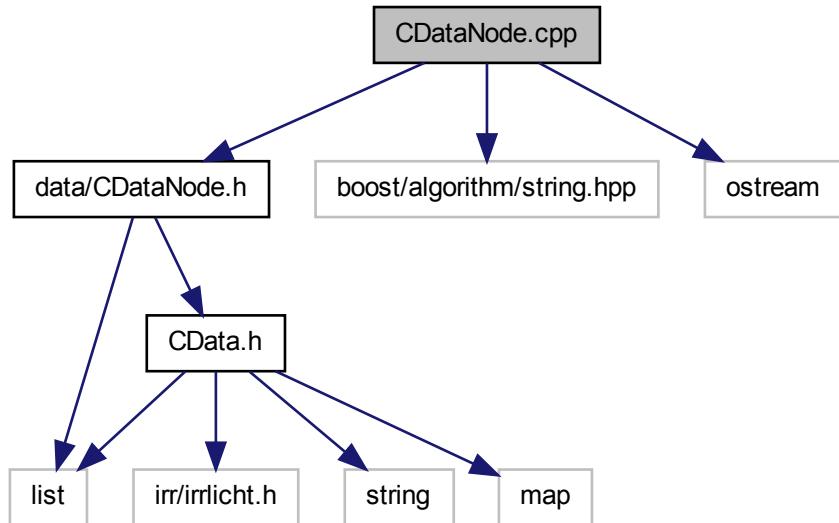
Auteur

argawaen

8.15 Référence du fichier CDataNode.cpp

```
#include "data/CDataNode.h"  
#include <boost/algorithm/string.hpp>  
#include <iostream>
```

Graphe des dépendances par inclusion de CDataNode.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) :: [data](#)

8.15.1 Description détaillée

Date

26 juil. 2011

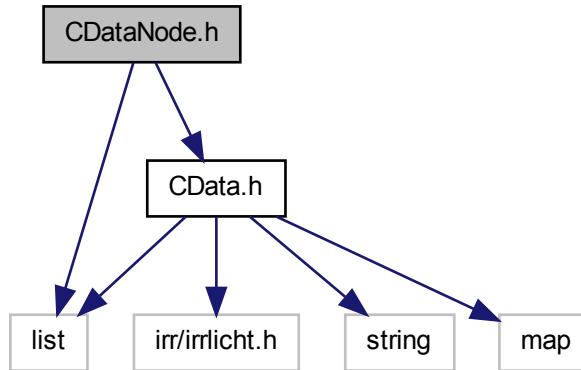
Auteur

dla

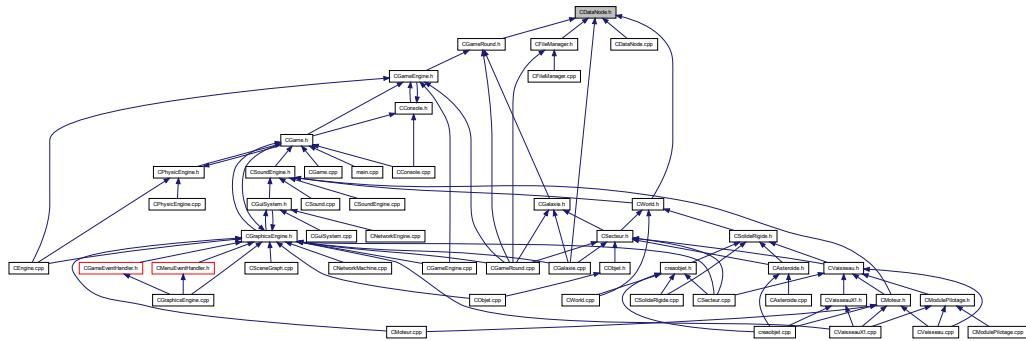
8.16 Référence du fichier CDataNode.h

```
#include <list>
#include "CData.h"
```

Graphe des dépendances par inclusion de CDataNode.h :



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



Classes

- class **ayoaron** : :data : :CDataNode
un bloc de donnée noeud de l'arbre

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : :data

8.16.1 Description détaillée

Date

26 juil. 2011

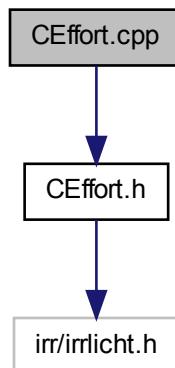
Auteur

dla

8.17 Référence du fichier CEffort.cpp

```
#include "CEffort.h"
```

Graphe des dépendances par inclusion de CEffort.cpp :

**Espaces de nommage**

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:physic](#)

8.17.1 Description détaillée

Date

14 sept. 2011

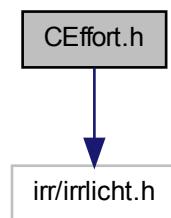
Auteur

dla

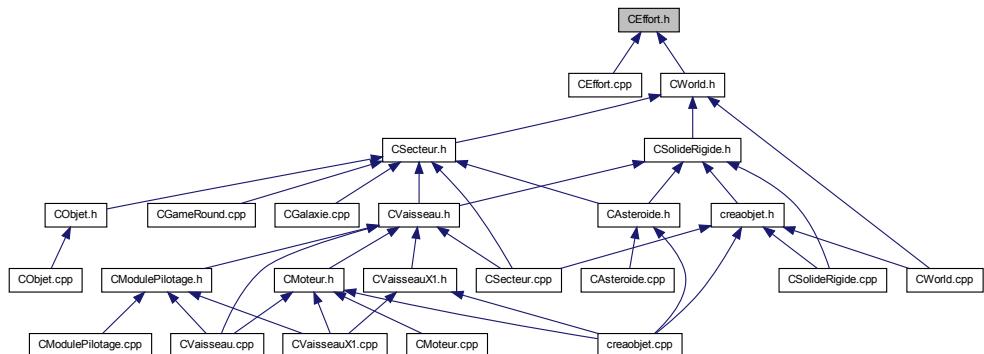
8.18 Référence du fichier CEffort.h

```
#include <irr/irrlicht.h>
```

Graphe des dépendances par inclusion de CEffort.h :



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



Classes

- class ayoaron : `:physic :CEffort`
classe permettant de définir un effort appliqué à un objet

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :physic**

8.18.1 Description détaillée

Date

14 sept. 2011

Auteur

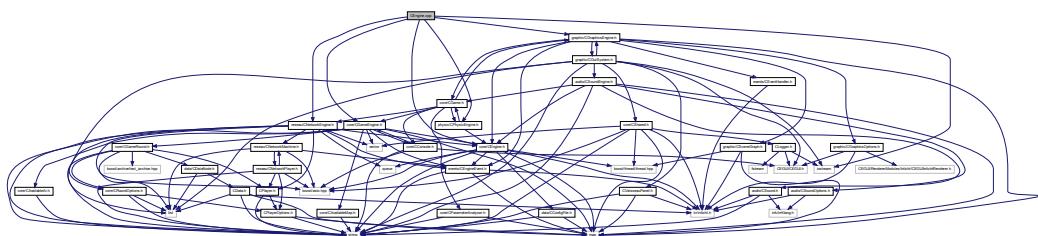
dla

8.19 Référence du fichier CEngine.cpp

Implémentation de la classe CEngine.

```
#include <iostream>
#include "core/CEngine.h"
#include "graphic/CGraphicsEngine.h"
#include "reseau/CNetworkEngine.h"
#include "core/CGameEngine.h"
#include "physic/CPhysicEngine.h"
```

Graphe des dépendances par inclusion de CEngine.cpp :

**Espaces de nommage**

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nommage contenant tous les objets de base

8.19.1 Description détaillée

Implémentation de la classe CEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

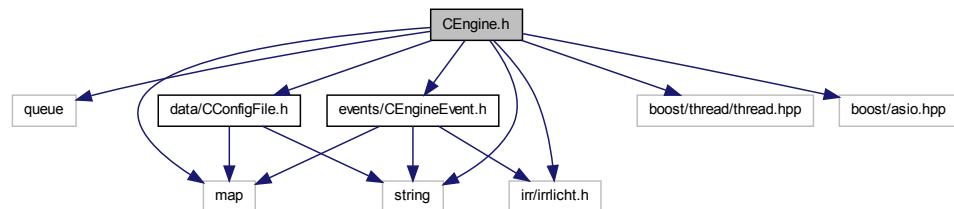
Implementation de la classe CEngine, c'est la classe de base des moteurs

8.20 Référence du fichier CEngine.h

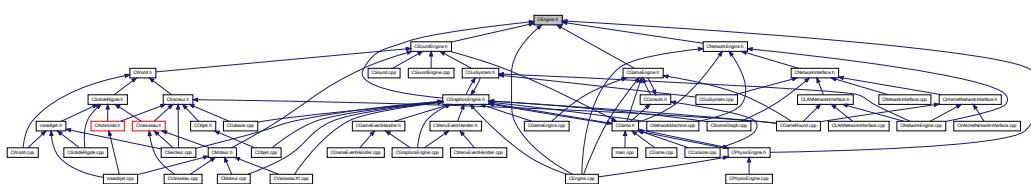
Déclaration de la classe CEngine.

```
#include <queue>
#include <map>
#include <string>
#include <boost/thread/thread.hpp>
#include <boost/asio.hpp>
#include <irr/irrlicht.h>
#include "data/CConfigFile.h"
#include "events/CEngineEvent.h"
```

Graphe des dépendances par inclusion de CEngine.h :



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

**Classes**

- class [ayoaron](#) : [core](#) : [CEngine](#)

Classe de base des moteurs.

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::audio**
espace de nommage contenant tous les objets destiné au moteur de son
- namespace **ayoaron::core**
espace de nommage contenant tous les objets de base
- namespace **ayoaron::graphic**
espace de nommage contenant tous les objets destiné au moteur graphique
- namespace **ayoaron::physic**
- namespace **ayoaron::reseau**
espace de nommage contenant tous les objets destiné au moteur réseau

8.20.1 Description détaillée

Déclaration de la classe CEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

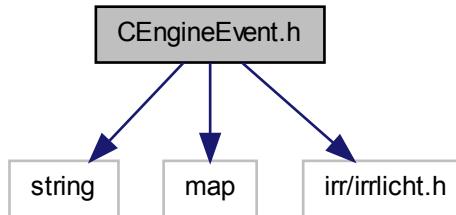
Déclaration de la classe CEngine, c'est la classe de base des moteurs

8.21 Référence du fichier CEngineEvent.h

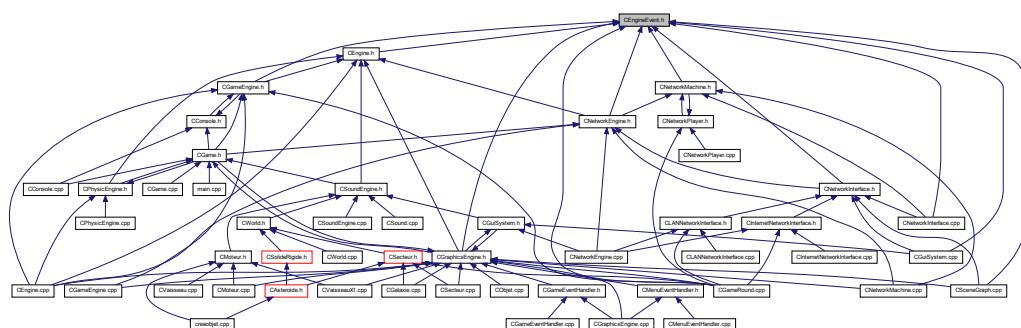
Déclaration de la classe CEngineEvent et CSerializableVector3df.

```
#include <string>
#include <map>
#include <irr/irrlicht.h>
```

Graphe des dépendances par inclusion de CEngineEvent.h :



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



Classes

- class `ayoaron::event::CEngineEvent`
classe générique pour manipuler les événements entre moteurs
- class `ayoaron::event::CSerializableVector3df`
classe vecteur sérialisable

Espaces de nommage

- namespace `ayoaron`
l'espace de nomage de base du projet ayoaron
- namespace `ayoaron::event`
espace de nomage contenant tous les objets destiné à la gestion des événements

8.21.1 Description détaillée

Déclaration de la classe CEngineEvent et CSerializableVector3df.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

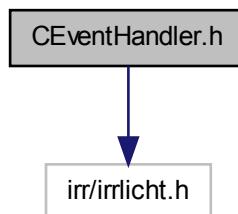
Déclaration de la classe CEngineEvent et CSerializableVector3df

8.22 Référence du fichier CEventHandler.h

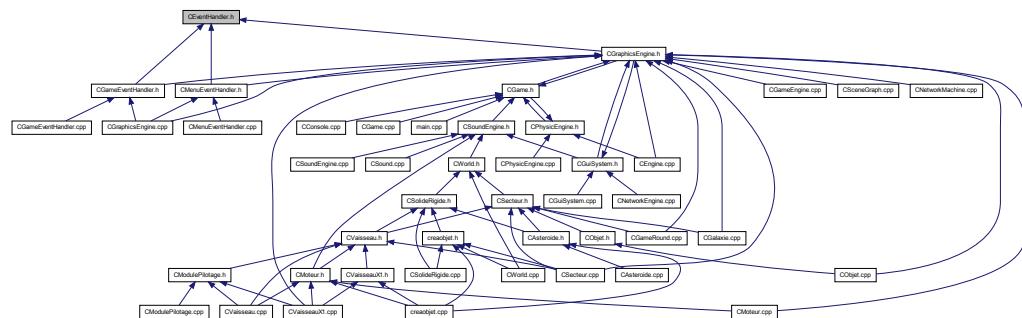
Déclaration de la classe CEventHandler.

```
#include <irr/irrlicht.h>
```

Graphe des dépendances par inclusion de CEventHandler.h :



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



Classes

- class ayoaron : event : CEventHandler
classe générique de gestionnaire d'événement

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron :event**
espace de nomage contenant tous les objets destiné à la gestion des événements
 - namespace **ayoaron :graphic**
espace de nomage contenant tous les objets destiné au moteur graphique

8.22.1 Description détaillée

Déclaration de la classe CEventHandler.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

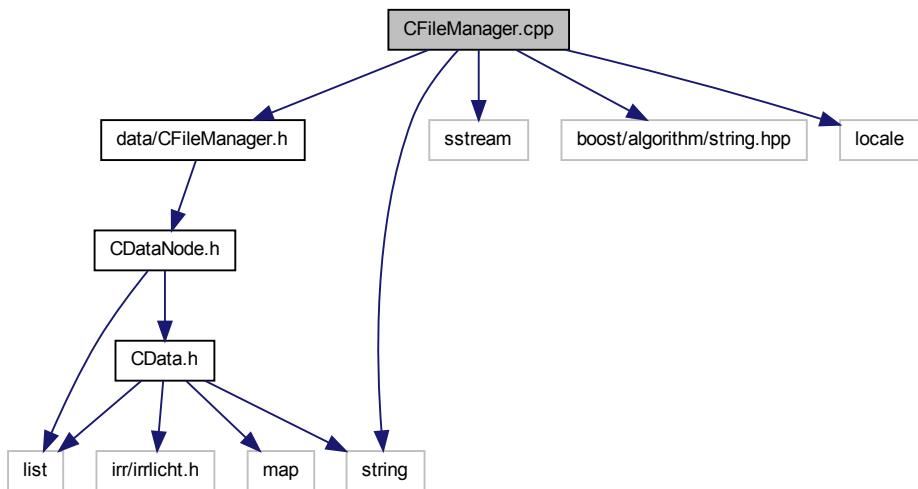
Déclaration de la classe CEventHandler

8.23 Référence du fichier CFileManager.cpp

```
#include "data/CFileManager.h"
```

```
#include <string>
#include <sstream>
#include <boost/algorithm/string.hpp>
#include <locale>
```

Graphe des dépendances par inclusion de CFileManager.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:data](#)

8.23.1 Description détaillée

Date

4 août 2011

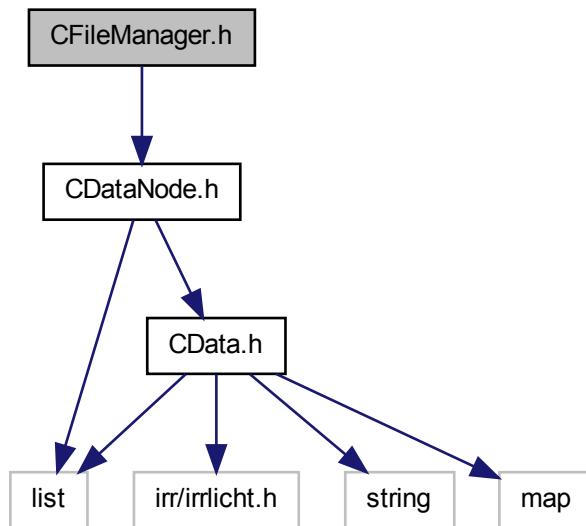
Auteur

argawaen

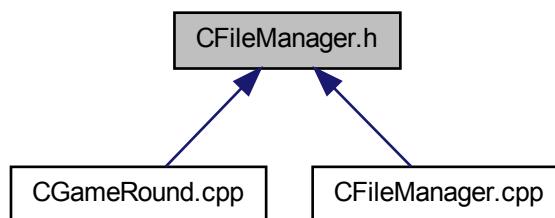
8.24 Référence du fichier CFileManager.h

```
#include "CDataNode.h"
```

Graphe des dépendances par inclusion de CFileManager.h :



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



Classes

- class `ayoaron ::data ::CFileManager`
classe de gestion des fichiers XML notamment

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:data](#)

8.24.1 Description détaillée**Date**

4 août 2011

Auteur

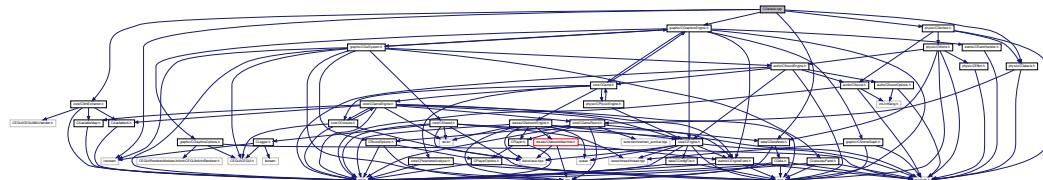
argawaen

8.25 Référence du fichier CGalaxie.cpp

la classe décrivant la galaxie

```
#include "physic/CGalaxie.h"  
#include "core/CXmlExtractor.h"  
#include "graphic/CGraphicsEngine.h"  
#include "physic/CSecteur.h"  
#include <iostream>  
#include "data/CDataNode.h"
```

Graphe des dépendances par inclusion de CGalaxie.cpp :

**Espaces de nommage**

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:physic](#)

8.25.1 Description détaillée

la classe décrivant la galaxie

Date

4 juin 2011

Auteur

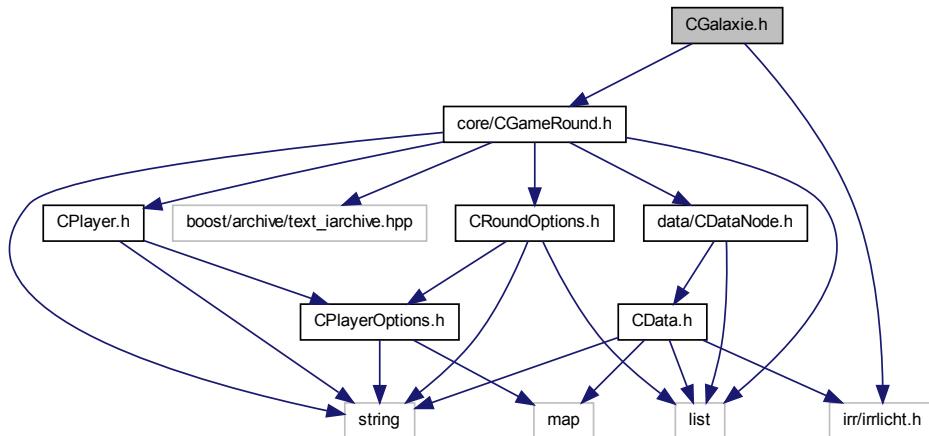
argawaen

8.26 Référence du fichier CGalaxie.h

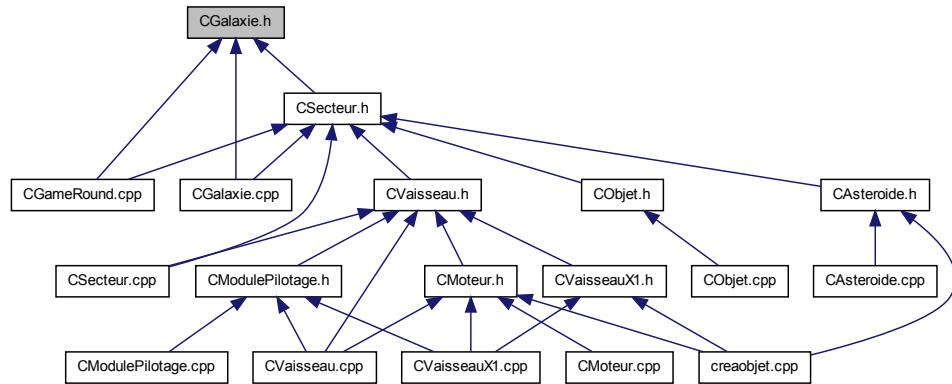
la classe décrivant la galaxie

```
#include "core/CGameRound.h"
#include <irr/irrlicht.h>
```

Graphe des dépendances par inclusion de CGalaxie.h :



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



Classes

- class **ayoaron** : **:physic** : **:CGalaxie**
la classe décrivant la galaxie

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** : **:audio**
espace de nommage contenant tous les objets destiné au moteur de son
- namespace **ayoaron** : **:graphic**
espace de nommage contenant tous les objets destiné au moteur graphique
- namespace **ayoaron** : **:physic**

8.26.1 Description détaillée

la classe décrivant la galaxie

Date

4 juin 2011

Auteur

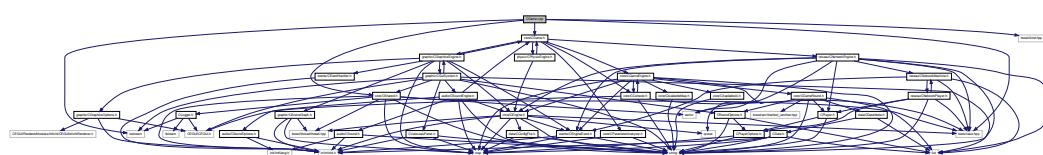
argawaen

8.27 Référence du fichier CGame.cpp

Implémentation de la classe game.

```
#include <iostream>
#include <boost/asio.hpp>
#include <boost/bind.hpp>
#include "core/CGame.h"
#include "core/CShared.h"
```

Graphe des dépendances par inclusion de CGame.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base

8.27.1 Description détaillée

Implémentation de la classe game.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Implémentation de la classe game, c'est la classe de base supportant toute l'architecture du jeu

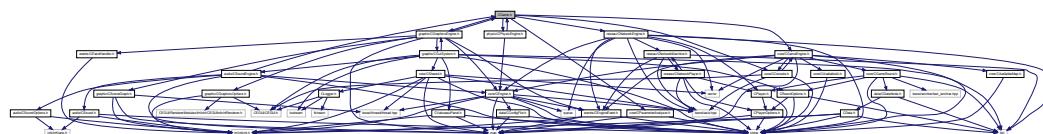
8.28 Référence du fichier CGame.h

Déclaration de la classe CGame.

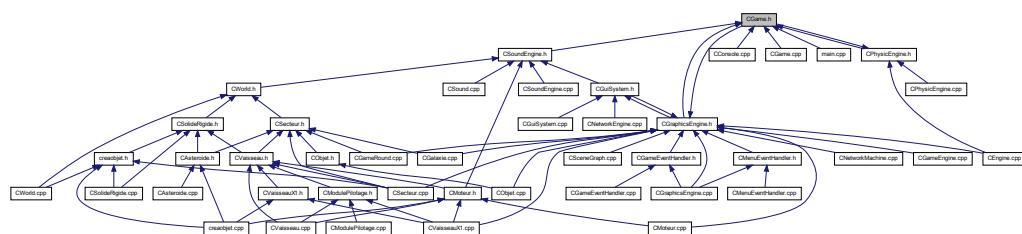
```
#include "core/CGameEngine.h"
#include "graphic/CGraphicsEngine.h"
#include "reseau/CNetworkEngine.h"
```

```
#include "physic/CPhysicsEngine.h"
#include "core/CParameterAnalyser.h"
#include "core/CConsole.h"
```

Graphe des dépendances par inclusion de CGame.h :



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



Classes

- class **ayoaron** : **core** : **CGame**
classe de base du jeu

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **core**
espace de nomage contenant tous les objets de base

8.28.1 Description détaillée

Déclaration de la classe CGame.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

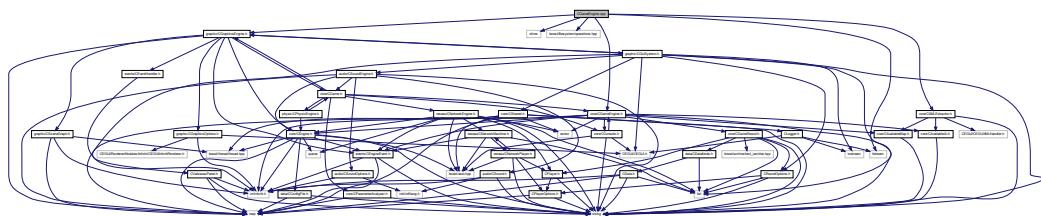
Déclaration de la classe CGame, c'est la classe de base supportant toute l'architecture du jeu

8.29 Référence du fichier CGameEngine.cpp

Implémentation de la classe CGameEngine.

```
#include <fstream>
#include <ctime>
#include <boost/filesystem/operations.hpp>
#include "core/CGameEngine.h"
#include "graphic/CGraphicsEngine.h"
#include "core/CXMLExtractor.h"
```

Graphe des dépendances par inclusion de CGameEngine.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base

8.29.1 Description détaillée

Implémentation de la classe CGameEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

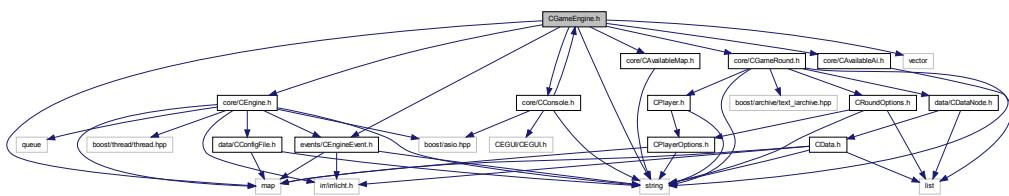
Implémentation de la classe CGameEngine, c'est la classe du moteur de jeu

8.30 Référence du fichier CGameEngine.h

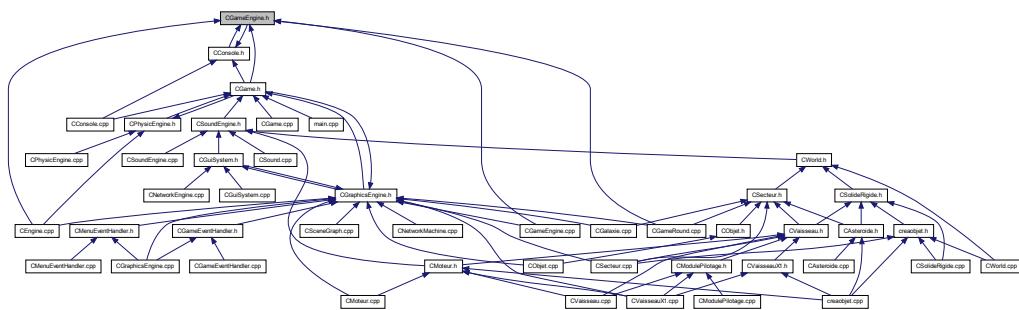
Déclaration de la classe CGameEngine.

```
#include <map>
#include <string>
#include <vector>
#include "core/CEngine.h"
#include "events/CEngineEvent.h"
#include "core/CGameRound.h"
#include "core/CConsole.h"
#include "core/CAvailableMap.h"
#include "core/CAvailableAi.h"
```

Graphe des dépendances par inclusion de CGameEngine.h :



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



Classes

- class ayoaron ::core ::CGameEngine

Le moteur de jeu.

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base
- namespace **ayoaron::graphic**
espace de nomage contenant tous les objets destiné au moteur graphique
- namespace **ayoaron::reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.30.1 Description détaillée

Déclaration de la classe CGameEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

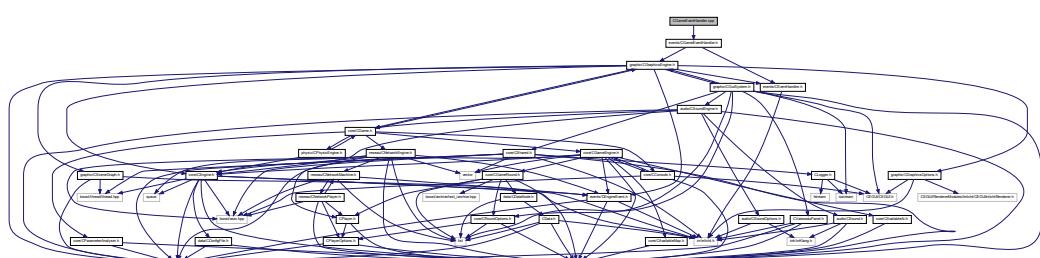
Déclaration de la classe CGameEngine, c'est la classe du moteur de jeu

8.31 Référence du fichier CGameEventHandler.cpp

Déclaration de la classe CGameEventHandler.

```
#include "events/CGameEventHandler.h"
```

Graphe des dépendances par inclusion de CGameEventHandler.cpp :



Espaces de nommage

- namespace **ayoaron**

l'espace de nomage de base du projet ayoaron

- namespace **ayoaron::event**

espace de nomage contenant tous les objets destiné à la gestion des événements

8.31.1 Description détaillée

Déclaration de la classe CGameEventHandler.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

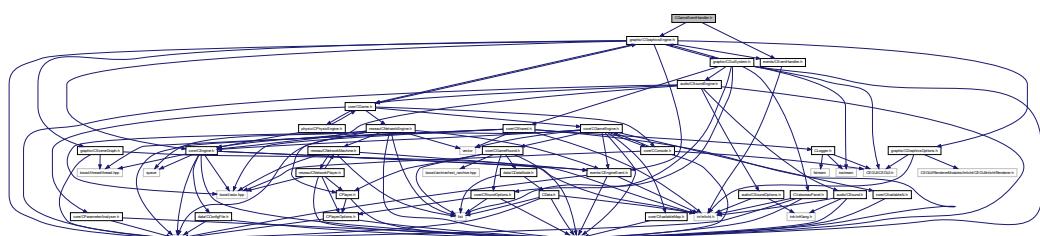
Déclaration de la classe CGameEventHandler

8.32 Référence du fichier CGameEventHandler.h

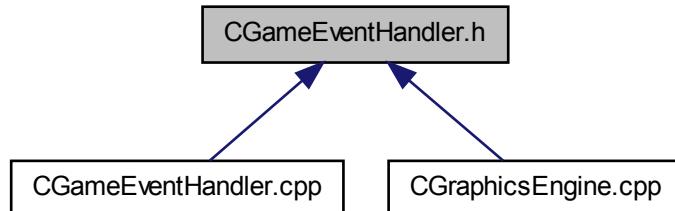
Déclaration de la classe CGameEventHandler.

```
#include "events/CEventHandler.h"  
#include "graphic/CGraphicsEngine.h"
```

Graphe des dépendances par inclusion de CGameEventHandler.h :



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



Classes

- class [ayoaron](#) : [:event](#) : [:CGameEventHandler](#)
gestionnaire d'évenement du jeu

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:event](#)
espace de nomage contenant tous les objets destiné à la gestion des événements

8.32.1 Description détaillée

Déclaration de la classe CGameEventHandler.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

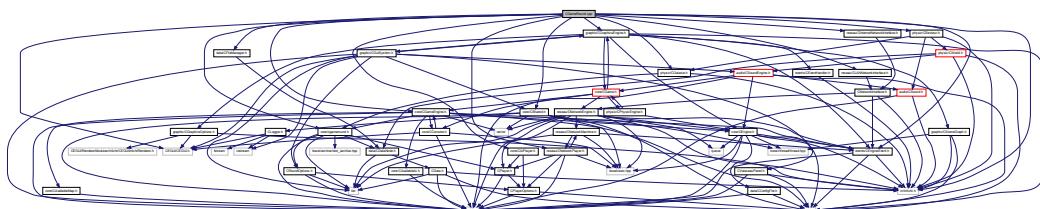
Déclaration de la classe CGameEventHandler

8.33 Référence du fichier CGameRound.cpp

Implémentation de la classe CGameRound.

```
#include <CEGUI/CEGUI.h>
#include <irr/irrlicht.h>
#include "core/cgameround.h"
#include "events/CEngineEvent.h"
#include "core/CGameEngine.h"
#include "core/CShared.h"
#include "reseau/CIInternetNetworkInterface.h"
#include "reseau/CLANNetworkInterface.h"
#include "graphic/CGraphicsEngine.h"
#include "core/CAiPlayer.h"
#include "reseau/CNetworkPlayer.h"
#include "physic/CGalaxie.h"
#include "physic/CSecteur.h"
#include "data/CFileManager.h"
```

Graphe des dépendances par inclusion de CGameRound.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base

8.33.1 Description détaillée

Implémentation de la classe CGameRound.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

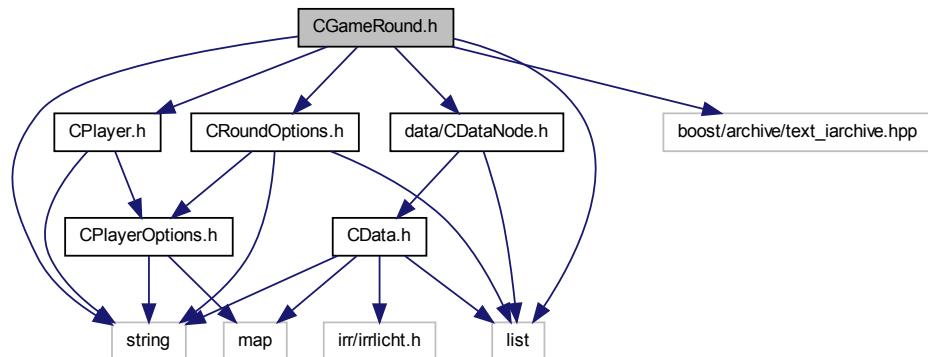
Implémentation de la classe CGameRound

8.34 Référence du fichier CGameRound.h

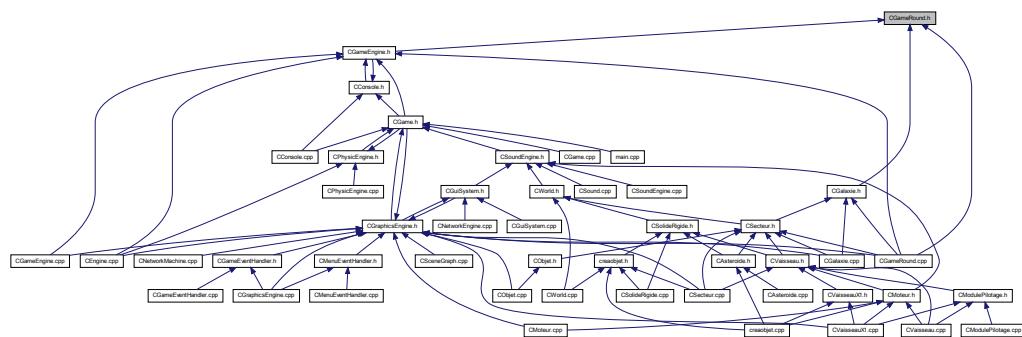
Déclaration de la classe CGameRound.

```
#include <string>
#include <list>
#include <boost/archive/text_iarchive.hpp>
#include "CPlayer.h"
#include "CRoundOptions.h"
#include "data/CDataNode.h"
```

Graphe des dépendances par inclusion de CGameRound.h :



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



Classes

- class ayoaron ::core ::CGameRound

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :core**
espace de nomage contenant tous les objets de base
 - namespace **ayoaron : :physic**

8.34.1 Description détaillée

Déclaration de la classe CGameRound.

Auteur

Argawaen

Version

0.1

Date

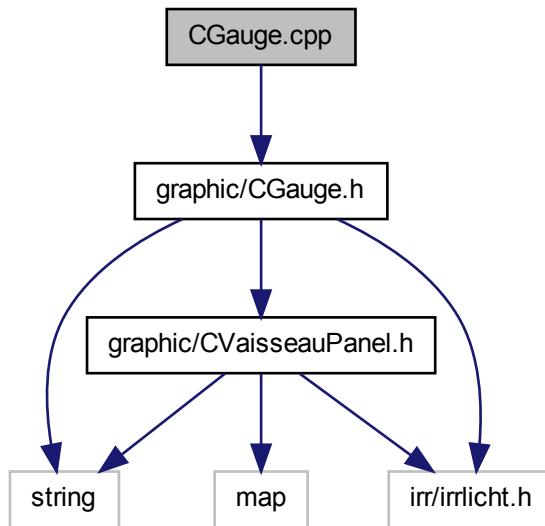
16 février 2011

Déclaration de la classe CGameRound

8.35 Référence du fichier CGauge.cpp

```
#include "graphic/CGauge.h"
```

Graphe des dépendances par inclusion de CGauge.cpp :



Espaces de nommage

- namespace `ayoaron`
l'espace de nommage de base du projet ayoaron
- namespace `ayoaron::graphic`
espace de nommage contenant tous les objets destiné au moteur graphique

8.35.1 Description détaillée

Date

22 août 2011

Auteur

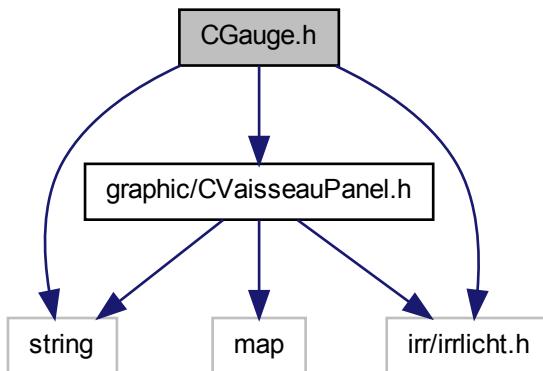
dla

8.36 Référence du fichier CGauge.h

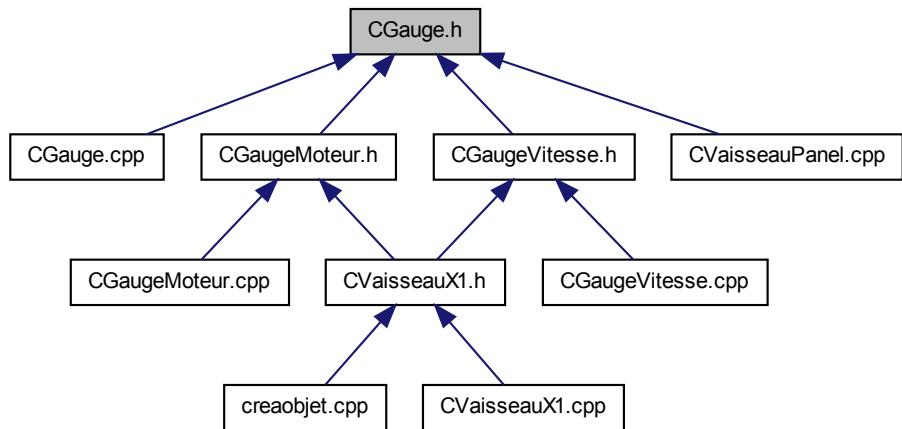
```

#include <string>
#include <irr/irrlicht.h>
#include "graphic/CVaisseauPanel.h"
  
```

Graphe des dépendances par inclusion de CGauge.h :



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



Classes

- class [ayoaron](#) : `:graphic` : `:CGauge`
classe d'encapsulation des compteurs

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.36.1 Description détaillée**Date**

22 août 2011

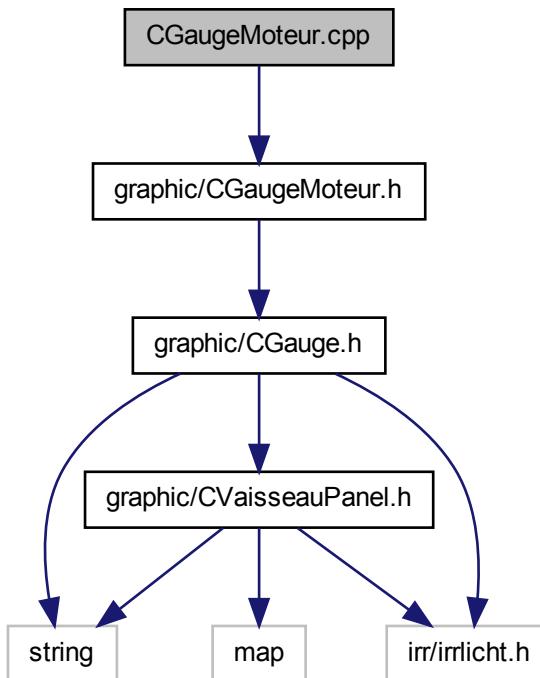
Auteur

dla

8.37 Référence du fichier CGaugeMoteur.cpp

```
#include "graphic/CGaugeMoteur.h"
```

Graphe des dépendances par inclusion de CGaugeMoteur.cpp :



Espaces de nommage

- namespace **ayoaron**

l'espace de nommage de base du projet ayoaron

- namespace **ayoaron::graphic**

espace de nommage contenant tous les objets destiné au moteur graphique

8.37.1 Description détaillée**Date**

23 août 2011

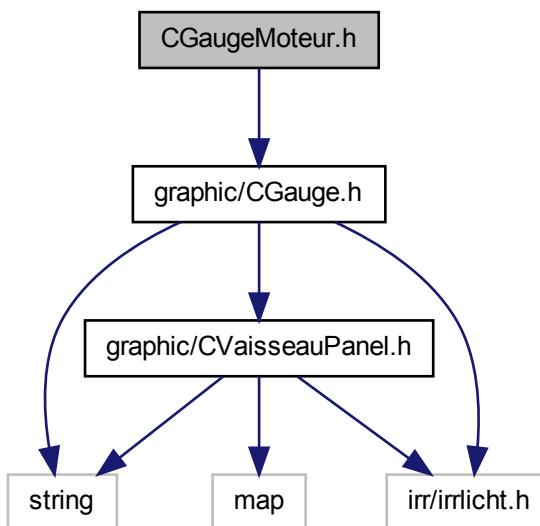
Auteur

argawaen

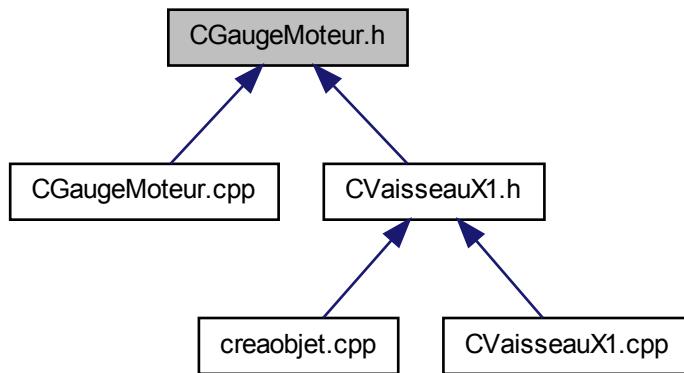
8.38 Référence du fichier CGaugeMoteur.h

```
#include "graphic/CGauge.h"
```

Graphe des dépendances par inclusion de CGaugeMoteur.h :



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



Classes

- class **ayoaron** : **graphic** : **CGaugeMoteur**
classe de dessin de l'écran moteur
- struct **ayoaron** : **graphic** : **datamot**

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** : **graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.38.1 Description détaillée

Date

23 août 2011

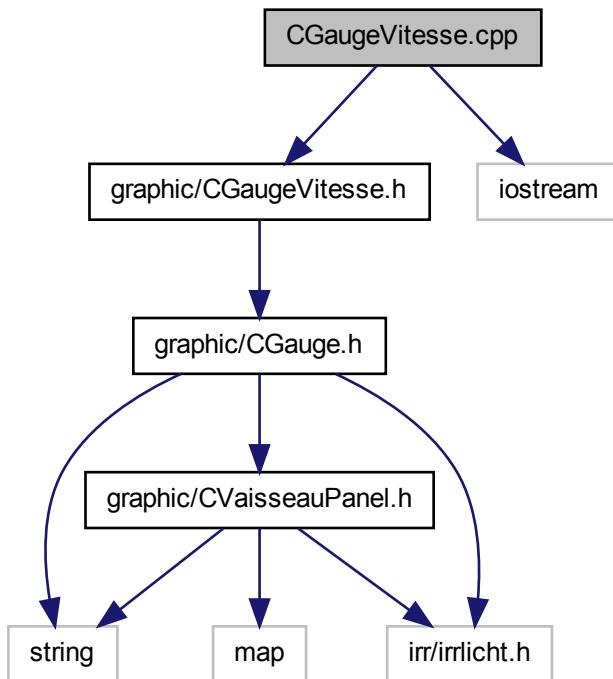
Auteur

argawaen

8.39 Référence du fichier CGaugeVitesse.cpp

```
#include "graphic/CGaugeVitesse.h"
#include <iostream>
```

Graphe des dépendances par inclusion de CGaugeVitesse.cpp :



Espaces de nommage

- namespace `ayoaron`
l'espace de nommage de base du projet ayoaron
- namespace `ayoaron :graphic`
espace de nommage contenant tous les objets destiné au moteur graphique

8.39.1 Description détaillée

Date

23 août 2011

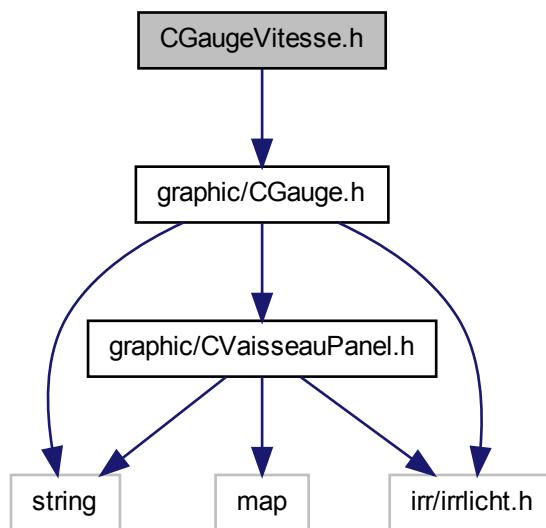
Auteur

dla

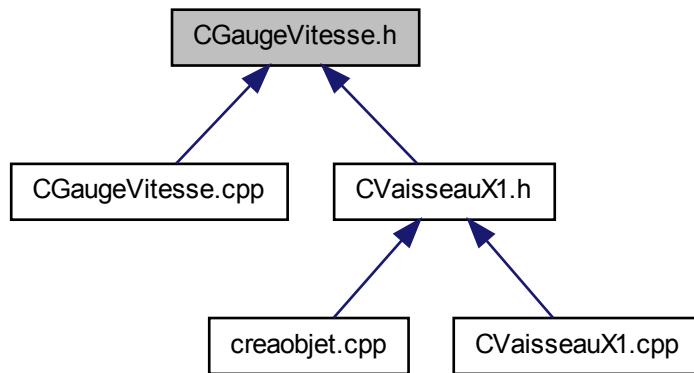
8.40 Référence du fichier CGaugeVitesse.h

```
#include "graphic/CGauge.h"
```

Graphe des dépendances par inclusion de CGaugeVitesse.h :



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



Classes

- class **ayoaron** : **graphic** : **CGaugeVitesse**
la classe de gestion du compteur de vitesse à six axes

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** : **graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.40.1 Description détaillée

Date

23 août 2011

Auteur

dla

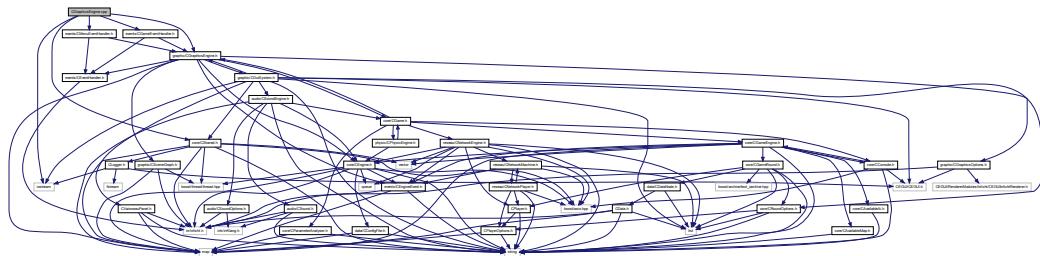
8.41 Référence du fichier CGraphicsEngine.cpp

Implémentation de la classe CGraphicsEngine.

```
#include <iostream>
```

```
#include "graphic/CGraphicsEngine.h"
#include "core/CShared.h"
#include "events/CMuEventHandler.h"
#include "events/CGameEventHandler.h"

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



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron :graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.41.1 Description détaillée

Implémentation de la classe CGraphicsEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Implémentation de la classe CGraphicsEngine

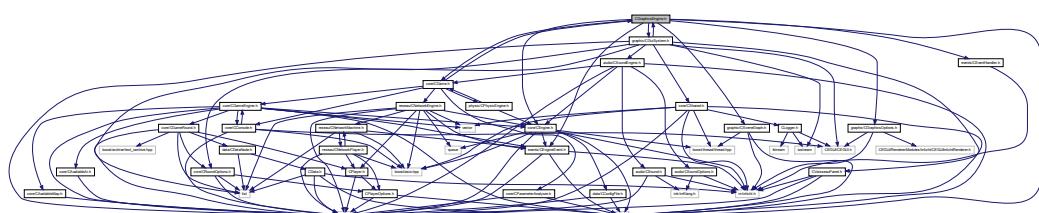
8.42 Référence du fichier CGraphicsEngine.h

Déclaration de la classe CGraphicsEngine.

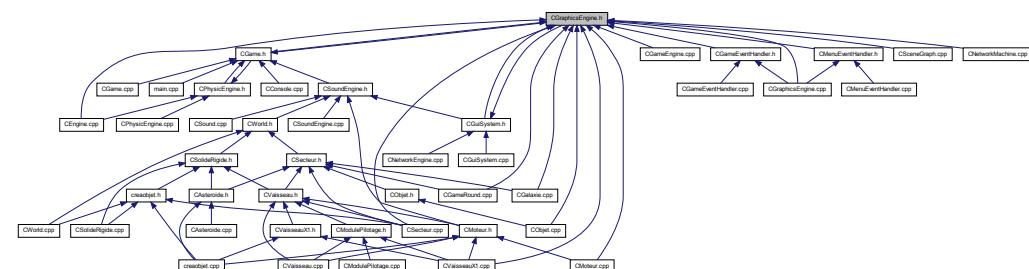
```
#include <map>
#include "graphic/CGraphicsOptions.h"
#include "core/CEngine.h"
```

```
#include "events/CEngineEvent.h"  
#include "graphic/CGuiSystem.h"  
#include "core/CGame.h"  
#include "events/CEventHandler.h"  
#include "graphic/CSceneGraph.h"
```

Graphe des dépendances par inclusion de CGraphicsEngine.h :



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



Classes

- class **ayoaron** : **:graphic** : **:CGraphicsEngine**
le moteur graphique

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :core**
espace de nomage contenant tous les objets de base
 - namespace **ayoaron : :graphic**
espace de nomage contenant tous les objets destiné au moteur graphique

8.42.1 Description détaillée

Déclaration de la classe CGraphicsEngine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Déclaration de la classe CGraphicsEngine

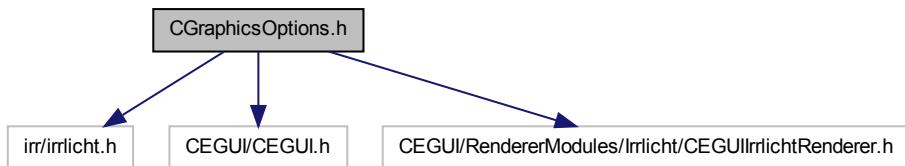
8.43 Référence du fichier CGraphicsOptions.h

Déclaration de la classe CGraphicsOptions.

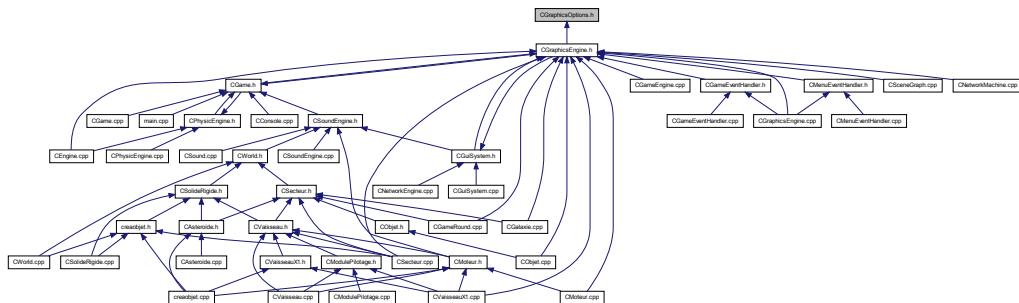
```
#include <irr/irrlicht.h>
#include <CEGUI/CEGUI.h>
```

```
#include <CEGUI/RendererModules/Irrlicht/CEGUIIrrlichtRenderer.h>
```

Graphe des dépendances par inclusion de CGraphicsOptions.h :



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



Classes

- class **ayoaron** : **graphic** : **CGraphicsOptions**
résumé des options graphiques

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **graphic**
espace de nomage contenant tous les objets destiné au moteur graphique

8.43.1 Description détaillée

Déclaration de la classe CGraphicsOptions.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Déclaration de la classe CGraphicsOptions

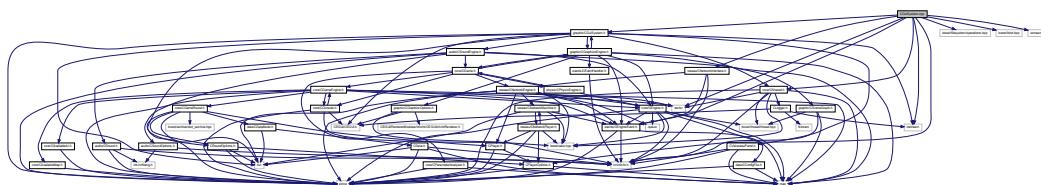
8.44 Référence du fichier CGuiSystem.cpp

Implémentation de la classe CGuiSystem.

```
#include <iostream>
#include <boost/filesystem/operations.hpp>
```

```
#include <boost/bind.hpp>
#include <vector>
#include <sstream>
#include "graphic/CGuiSystem.h"
#include "events/CEngineEvent.h"
#include "core/CShared.h"
#include "core/CPlayerOptions.h"
#include "reseau/CNetworkInterface.h"
```

Graphe des dépendances par inclusion de CGuiSystem.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.44.1 Description détaillée

Implémentation de la classe CGuiSystem.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

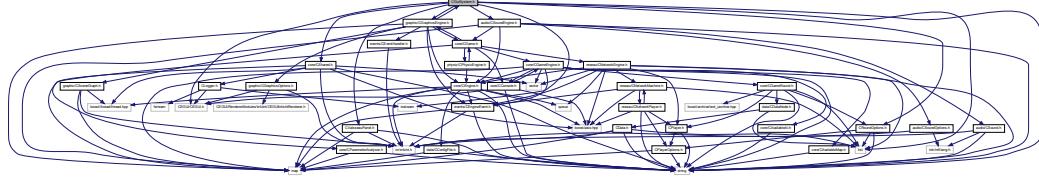
Implémentation de la classe CGuiSystem

8.45 Référence du fichier CGuiSystem.h

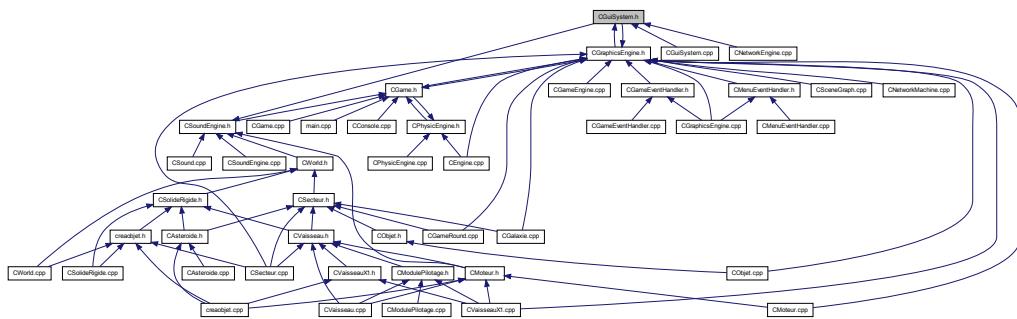
Déclaration de la classe CGuiSystem.

```
#include <CEGUI/CEGUI.h>
#include <iostream>
#include <list>
#include <string>
#include "graphic/CGraphicsEngine.h"
#include "CVaisseauPanel.h"
#include "core/CShared.h"
#include "core/CRoundOptions.h"
#include "audio/CSoundEngine.h"
Graphe des dépendances par inclusion de CGuiSystem
```

Graphe des dépendances par inclusion de CGuiSystem.h :



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



Classes

- class ayoaron : :graphic : :CGuiSystem
class traitant entièrement l'interface utilisateur (avec CEGUI) chaque événement déclenche un handle_XX , appelant une fonction XX

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron

- namespace **ayoaron** : **:graphic**

espace de nomage contenant tous les objets destiné au moteur graphique

8.45.1 Description détaillée

Déclaration de la classe CGuiSystem.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

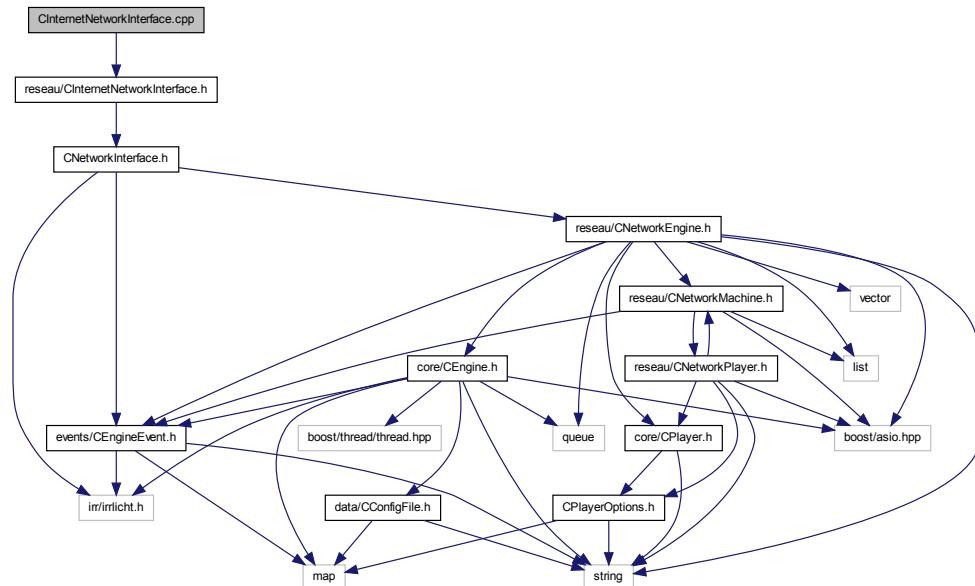
Déclaration de la classe CGuiSystem

8.46 Référence du fichier CInternetNetworkInterface.cpp

Déclaration de la classe CInternetNetworkInterface.

```
#include "reseau/CInternetNetworkInterface.h"
```

Graphe des dépendances par inclusion de CInternetNetworkInterface.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.46.1 Description détaillée

Déclaration de la classe CInternetNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

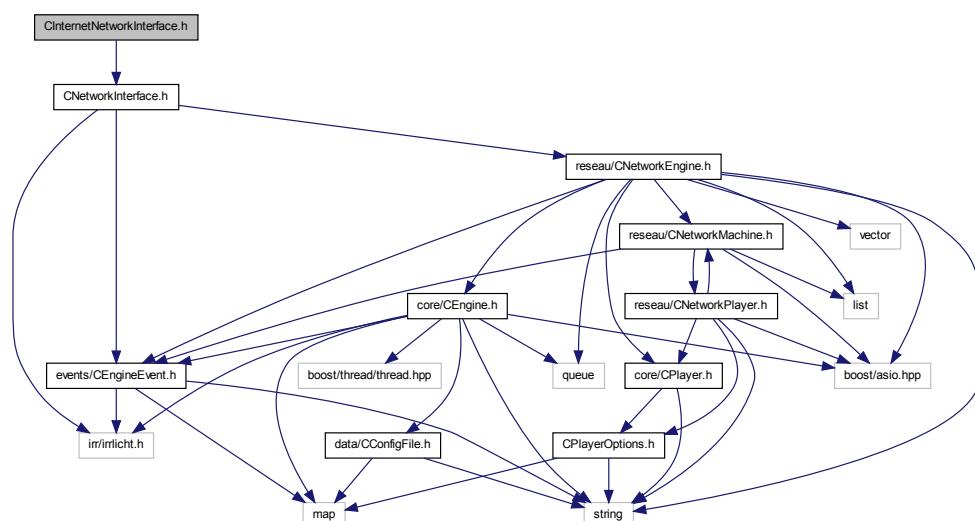
Déclaration de la classe CInternetNetworkInterface

8.47 Référence du fichier CInternetNetworkInterface.h

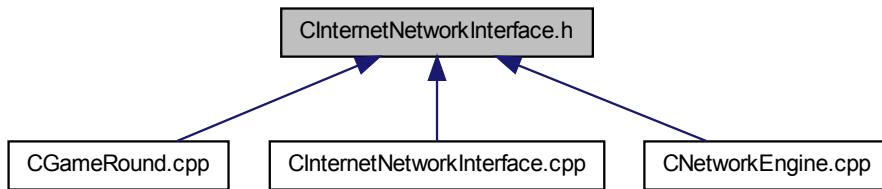
Déclaration de la classe CInternetNetworkInterface.

```
#include "CNetworkInterface.h"
```

Graphe des dépendances par inclusion de CInternetNetworkInterface.h :



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



Classes

- class `ayoaron : :reseau : :CInternetNetworkInterface`
interface réseau internet

Espaces de nommage

- namespace `ayoaron`
l'espace de nomage de base du projet ayoaron
- namespace `ayoaron : :reseau`
espace de nomage contenant tous les objets destiné au moteur réseau

8.47.1 Description détaillée

Déclaration de la classe CInternetNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

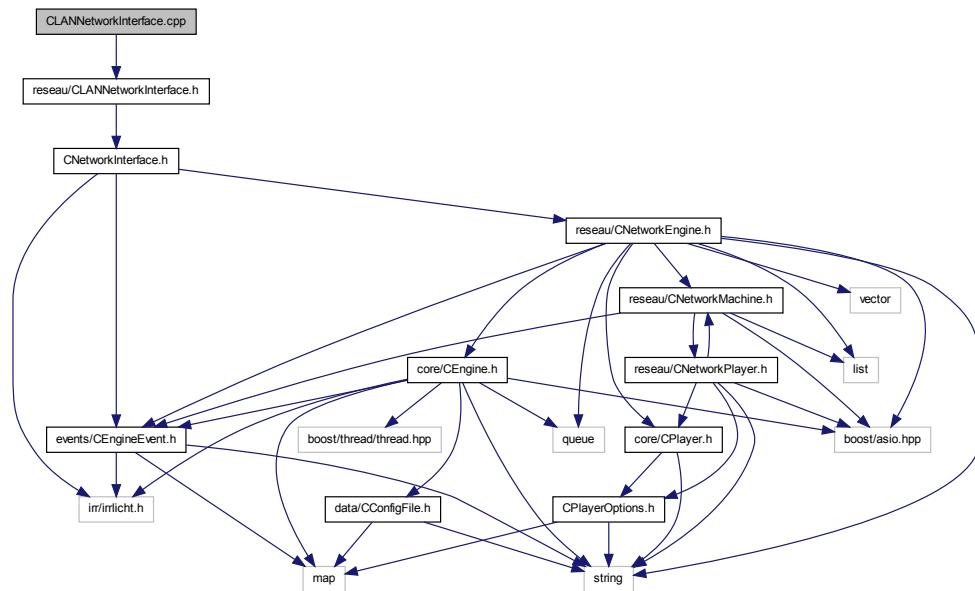
Déclaration de la classe CInternetNetworkInterface

8.48 Référence du fichier CLANNetworkInterface.cpp

Déclaration de la classe CLANNetworkInterface.

```
#include "reseau/CLANNetworkInterface.h"
```

Graphe des dépendances par inclusion de CLANNetworkInterface.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::reseau**
espace de nommage contenant tous les objets destiné au moteur réseau

8.48.1 Description détaillée

Déclaration de la classe CLANNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

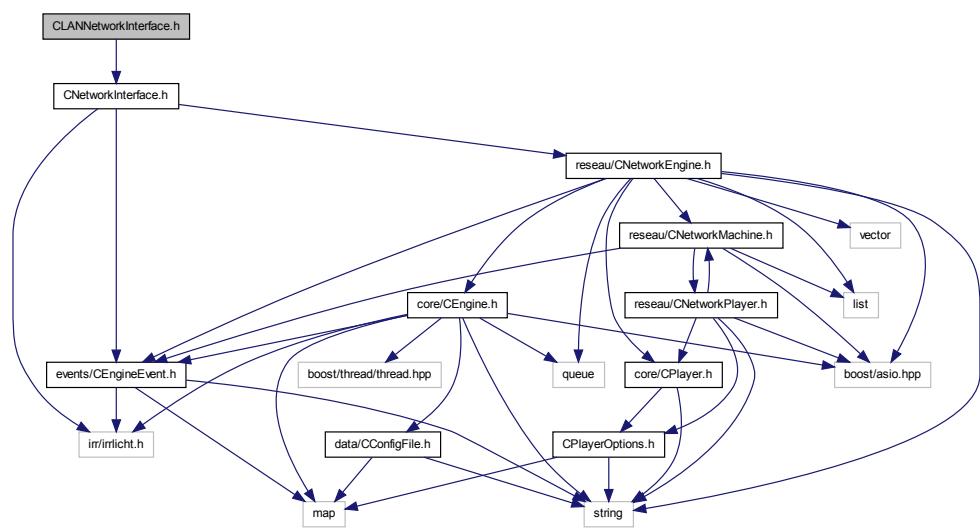
Déclaration de la classe CLANNetworkInterface

8.49 Référence du fichier CLANNetworkInterface.h

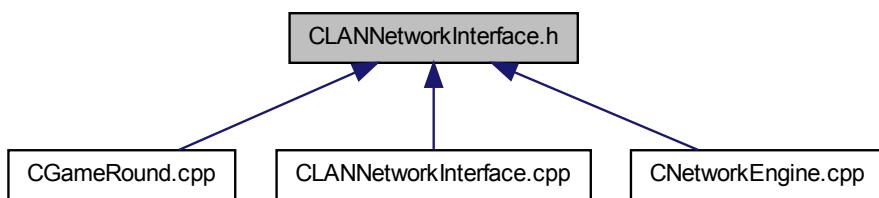
Déclaration de la classe CLANNetworkInterface.

```
#include "CNetworkInterface.h"
```

Graphe des dépendances par inclusion de CLANNetworkInterface.h :



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



Classes

- class [ayoaron](#) : [:reseau](#) : [:CLANNetworkInterface](#)
interface réseau LAN

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron::reseau](#)
espace de nomage contenant tous les objets destiné au moteur réseau

8.49.1 Description détaillée

Déclaration de la classe CLANNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

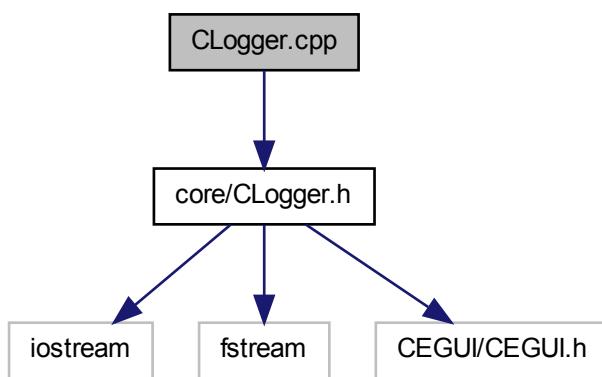
Déclaration de la classe CLANNetworkInterface

8.50 Référence du fichier CLogger.cpp

Implémentation de la classe CLogger.

```
#include "core/CLogger.h"
```

Graphe des dépendances par inclusion de CLogger.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nomage contenant tous les objets de base

8.50.1 Description détaillée

Implémentation de la classe CLogger.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

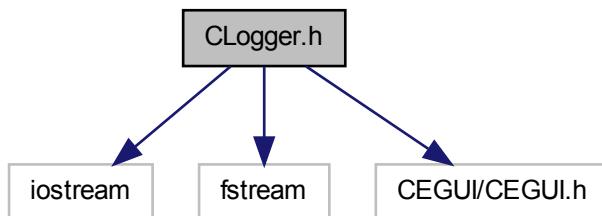
Implémentation de la classe CLogger

8.51 Référence du fichier CLogger.h

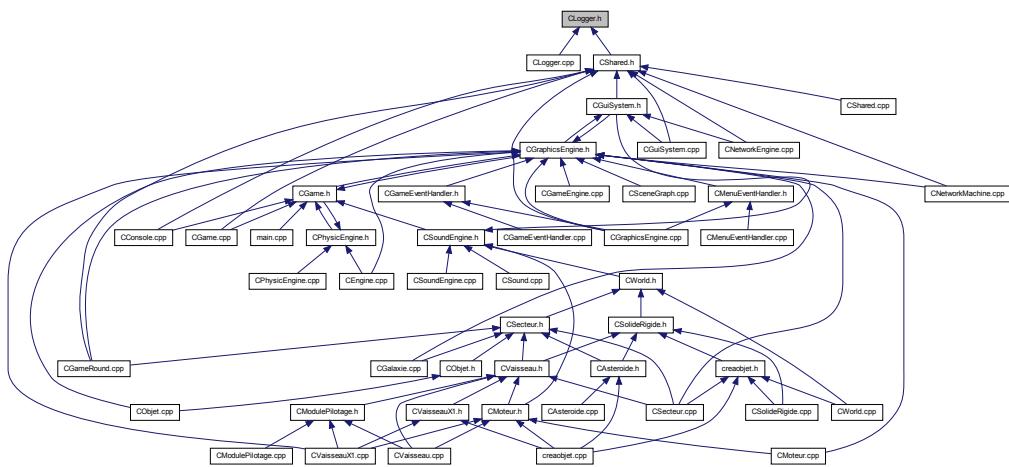
Déclaration de la classe CLogger.

```
#include <iostream>
#include <fstream>
#include <CEGUI/CEGUI.h>
```

Graphe des dépendances par inclusion de CLogger.h :



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



Classes

- class **ayoaron** : **core** : **CLogger**
module de log

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **core**
espace de nomage contenant tous les objets de base

8.51.1 Description détaillée

Déclaration de la classe CLogger.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

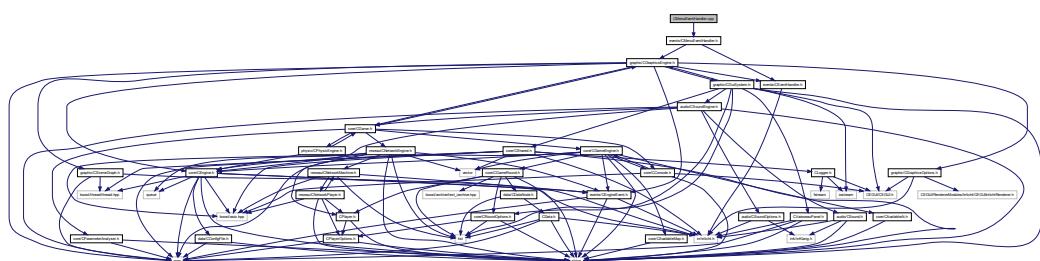
Déclaration de la classe CLogger

8.52 Référence du fichier CMenuEventHandler.cpp

Déclaration de la classe CMenuEventHandler.

```
#include "events/CMenuEventHandler.h"
```

Graphe des dépendances par inclusion de CMenuEventHandler.cpp :



Espaces de nommage

- namespace **ayoaron**

l'espace de nommage de base du projet ayoaron

- namespace **ayoaron::event**

espace de nommage contenant tous les objets destiné à la gestion des événements

8.52.1 Description détaillée

Déclaration de la classe CMenuEventHandler.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Déclaration de la classe CMenuEventHandler

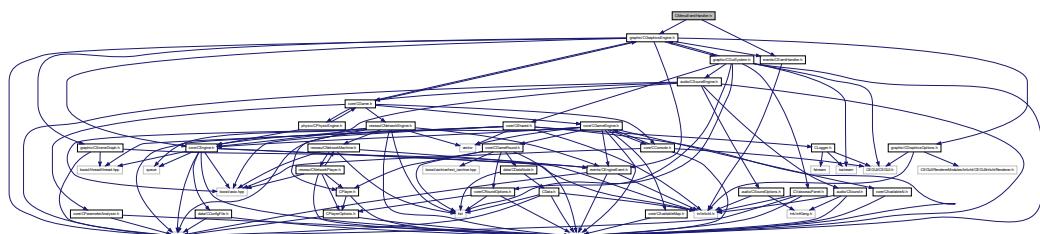
8.53 Référence du fichier CMenuEventHandler.h

Déclaration de la classe CMenuEventHandler.

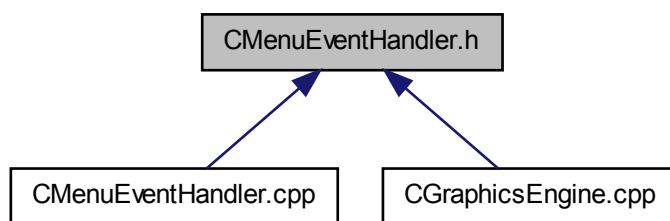
```
#include "events/CEventHandler.h"
```

```
#include "graphic/CGraphicsEngine.h"
```

Graphe des dépendances par inclusion de CMenuEventHandler.h :



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



Classes

- class `ayoaron ::event ::CMenuEventHandler`
gestionnaire d'évenement du menu

Espaces de nommage

- namespace `ayoaron`
l'espace de nomage de base du projet ayoaron
- namespace `ayoaron ::event`
espace de nomage contenant tous les objets destiné à la gestion des événements

8.53.1 Description détaillée

Déclaration de la classe CMenuEventHandler.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

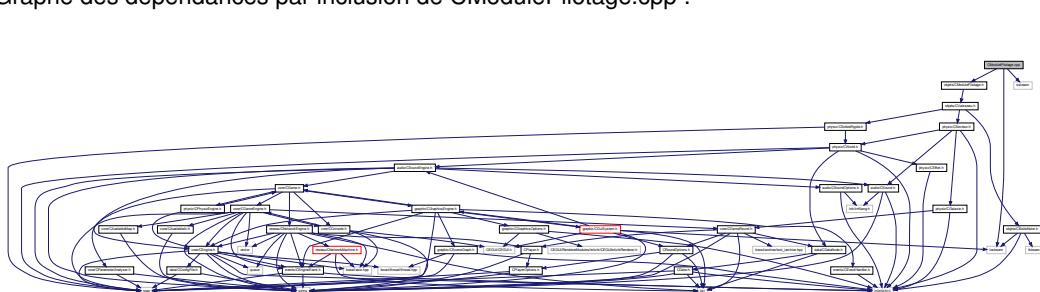
Déclaration de la classe CMenuEventHandler

8.54 Référence du fichier CModulePilotage.cpp

```
#include "objets/CModulePilotage.h"
```

```
#include <sstream>
```

```
#include <iostream>
```



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **avoaron :objets**

8.54.1 Description détaillée

Date

19 iul. 2011

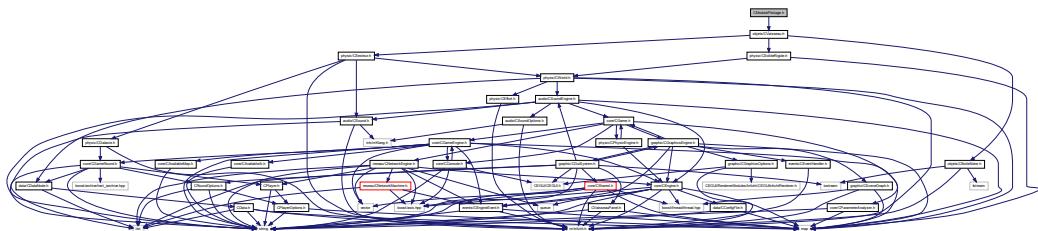
Auteur

argawaen

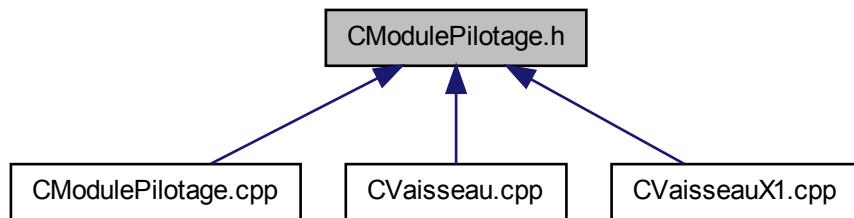
8.55 Référence du fichier CModulePilotage.h

```
#include "objets/Cvaisseau.h"
```

Graphe des dépendances par inclusion de CModulePilotage.h :



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



Classes

- class `ayoaron ::objets ::CModulePilotage`
une classe de module de pilotage

Espaces de nommage

- namespace `ayoaron`
l'espace de nommage de base du projet ayoaron
- namespace `ayoaron ::objets`

8.55.1 Description détaillée

Date

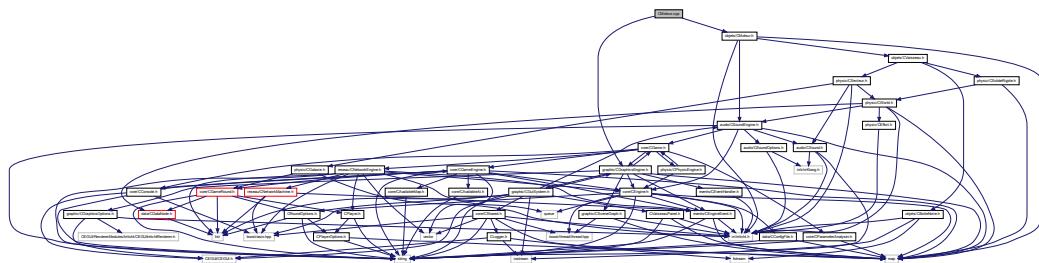
19 juil. 2011

Auteur

argawaen

8.56 Référence du fichier CMoteur.cpp

```
#include "objets/CMoteur.h"
#include "graphic/CGraphicsEngine.h"
Graphe des dépendances par inclusion de CMoteur.cpp :
```



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::objets**

8.56.1 Description détaillée

Date

18 juin 2011

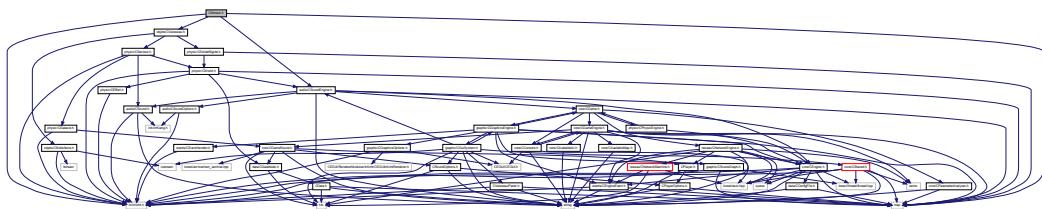
Auteur

argawaen

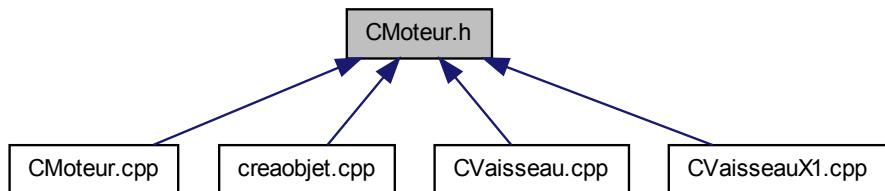
8.57 Référence du fichier CMoteur.h

```
#include <irr/irrlicht.h>
#include "objets/CVaisseau.h"
#include "audio/CSoundEngine.h"
#include <map>
```

Graphe des dépendances par inclusion de CMoteur.h :



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



Classes

- class **ayoaron** : :objets : :CMoteur
classe de description des moteurs
- class **ayoaron** : :objets : :CPousMouv
une classe de vecteur un peu particulière spécifique au pilotage des moteurs

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : :objets

8.57.1 Description détaillée

Date

18 juin 2011

Auteur

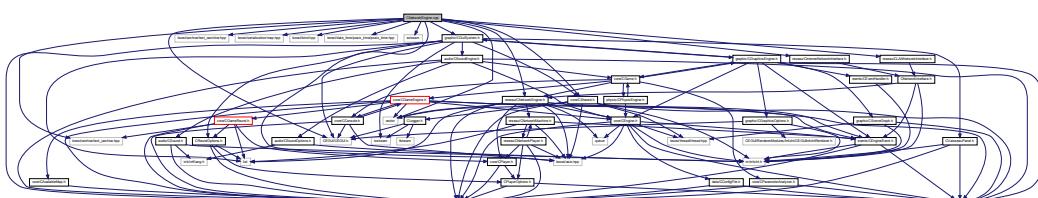
argawaen

8.58 Référence du fichier CNetworkEngine.cpp

Déclaration de la classe `network_engine`.

```
#include <boost/asio.hpp>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/map.hpp>
#include <boost/bind.hpp>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <sstream>
#include <vector>
#include <CEGUI/CEGUI.h>
#include "reseau/CNetworkEngine.h"
#include "core/CShared.h"
#include "graphic/CGuiSystem.h"
#include "reseau/CLANNetworkInterface.h"
#include "reseau/CIternetNetworkInterface.h"
```

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



Espaces de nommage

- namespace `ayoaron`
l'espace de nomage de base du projet ayoaron
- namespace `ayoaron::reseau`
espace de nomage contenant tous les objets destiné au moteur réseau

8.58.1 Description détaillée

Déclaration de la classe `network_engine`.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

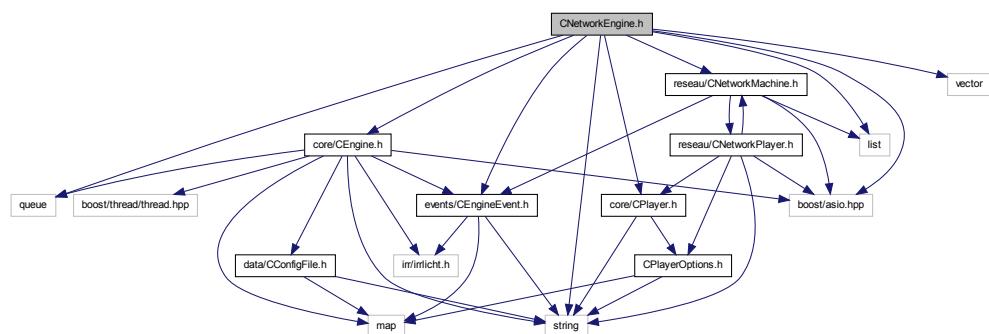
Déclaration de la classe network_engine

8.59 Référence du fichier CNetworkEngine.h

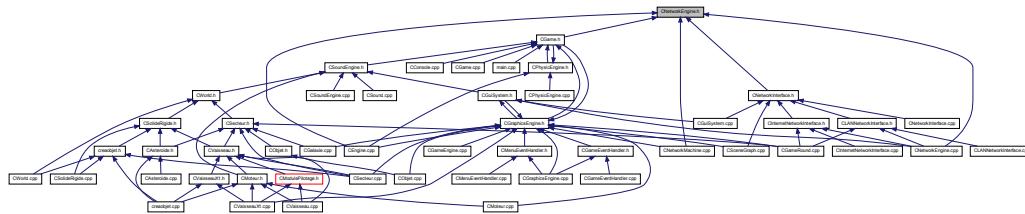
Déclaration de la classe network_engine.

```
#include <queue>
#include <vector>
#include <string>
#include <list>
#include <boost/asio.hpp>
#include "core/CEngine.h"
#include "events/CEngineEvent.h"
#include "core/CPlayer.h"
#include "reseau/CNetworkMachine.h"
```

Graphe des dépendances par inclusion de CNetworkEngine.h :



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



Classes

- class ayoaron ::reseau ::CNetworkEngine
moteur réseau

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :core**
espace de nomage contenant tous les objets de base
 - namespace **ayoaron : :reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

Macros

- `#define BUFFER_SIZE 1024`

8.59.1 Description détaillée

Déclaration de la classe `network_engine`.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Déclaration de la classe network_engine

8.59.2 Documentation des macros

8.59.2.1 #define BUFFER_SIZE 1024

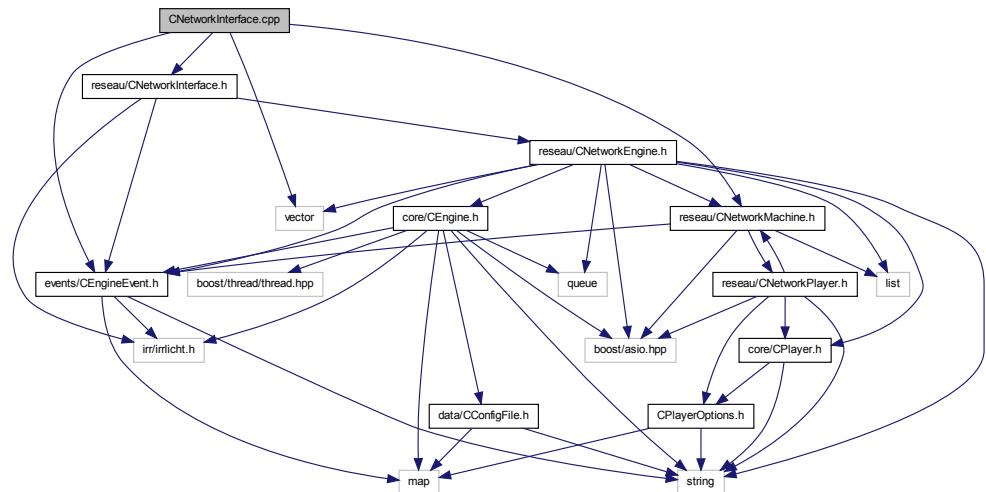
Référencé par ayoaron ::reseau ::CNetworkEngine ::getConfigData().

8.60 Référence du fichier CNetworkInterface.cpp

Implémentation de la classe CNetworkInterface.

```
#include <vector>
#include "reseau/CNetworkInterface.h"
#include "events/CEngineEvent.h"
#include "reseau/CNetworkMachine.h"
```

Graphe des dépendances par inclusion de CNetworkInterface.cpp :



Espaces de nommage

- namespace [ayoaron](#)
 - l'espace de nommage de base du projet ayoaron*
- namespace [ayoaron](#) ::reseau
 - espace de nommage contenant tous les objets destiné au moteur réseau*

8.60.1 Description détaillée

Implémentation de la classe CNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

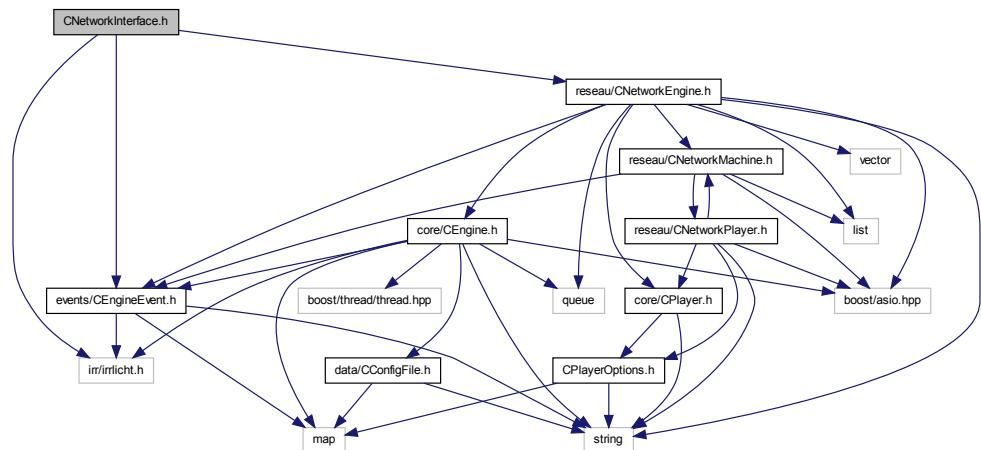
Implémentation de la classe CNetworkInterface

8.61 Référence du fichier CNetworkInterface.h

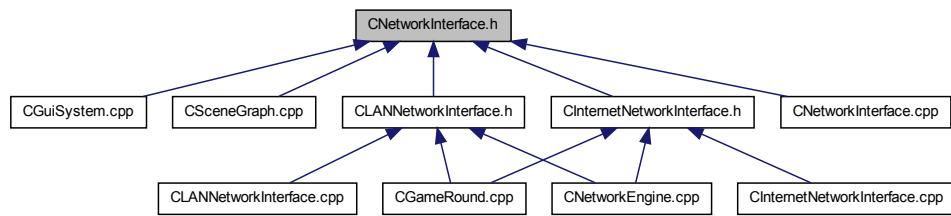
Déclaration de la classe CNetworkInterface.

```
#include <irr/irrlicht.h>
#include "events/CEngineEvent.h"
#include "reseau/CNetworkEngine.h"
```

Graphe des dépendances par inclusion de CNetworkInterface.h :



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



Classes

- class [ayoaron ::reseau ::CNetworkInterface](#)
interface réseau

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron ::reseau](#)
espace de nommage contenant tous les objets destiné au moteur réseau

8.61.1 Description détaillée

Déclaration de la classe CNetworkInterface.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Déclaration de la classe CNetworkInterface

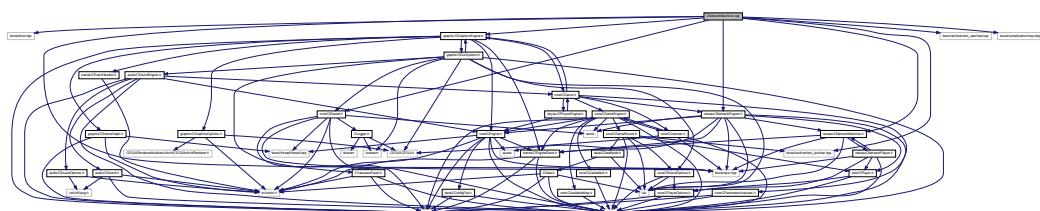
8.62 Référence du fichier CNetworkMachine.cpp

Declaration de la classe CNetworkMachine.

```
#include <boost/bind.hpp>
#include <boost/asio.hpp>
```

```
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/map.hpp>
#include "reseau/CNetworkMachine.h"
#include "reseau/CNetworkEngine.h"
#include "graphic/CGraphicsEngine.h"
#include "core/CShared.h"
```

Graphe des dépendances par inclusion de CNetworkMachine.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.62.1 Description détaillée

Declaration de la classe CNetworkMachine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

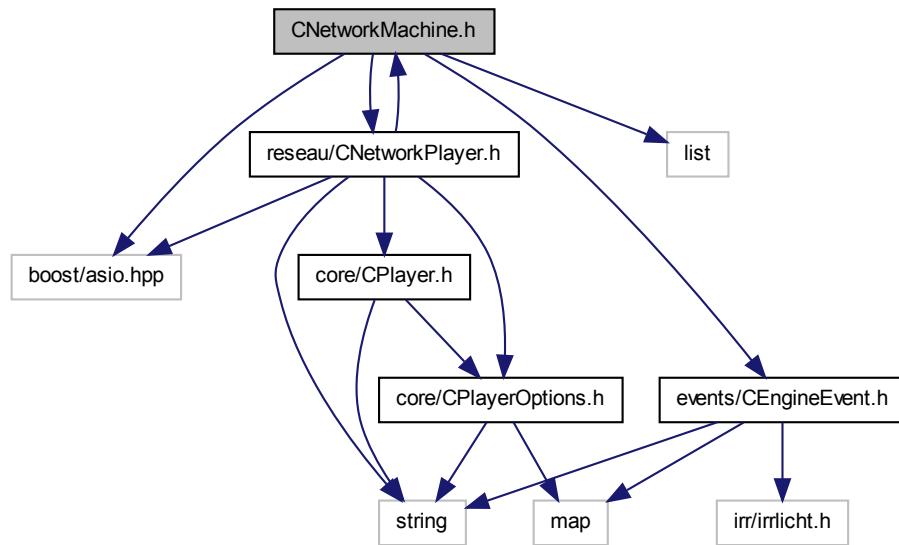
Declaration de la classe CNetworkMachine

8.63 Référence du fichier CNetworkMachine.h

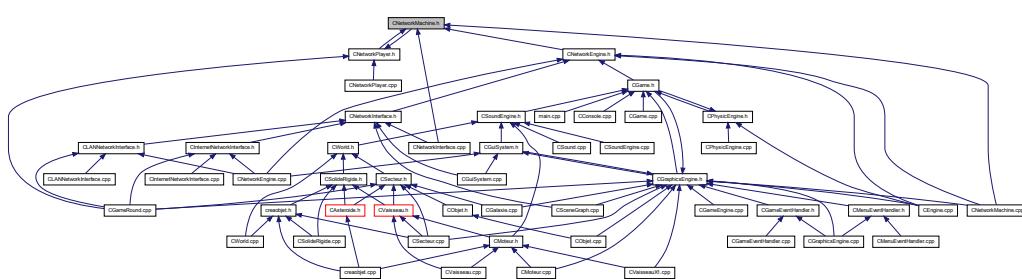
Declaration de la classe CNetworkMachine.

```
#include <boost/asio.hpp>
```

```
#include <list>
#include "reseau/CNetworkPlayer.h"
#include "events/CEngineEvent.h"
Graphe des dépendances par inclusion de CNetworkMachine.h :
```



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



Classes

- class ayoaron : :reseau : :CNetworkMachine
une machine réseau

Espaces de nommage

- namespace [ayoaron](#)
l'espace de nomage de base du projet ayoaron
- namespace [ayoaron](#) : [:reseau](#)
espace de nomage contenant tous les objets destiné au moteur réseau

8.63.1 Description détaillée

Declaration de la classe CNetworkMachine.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

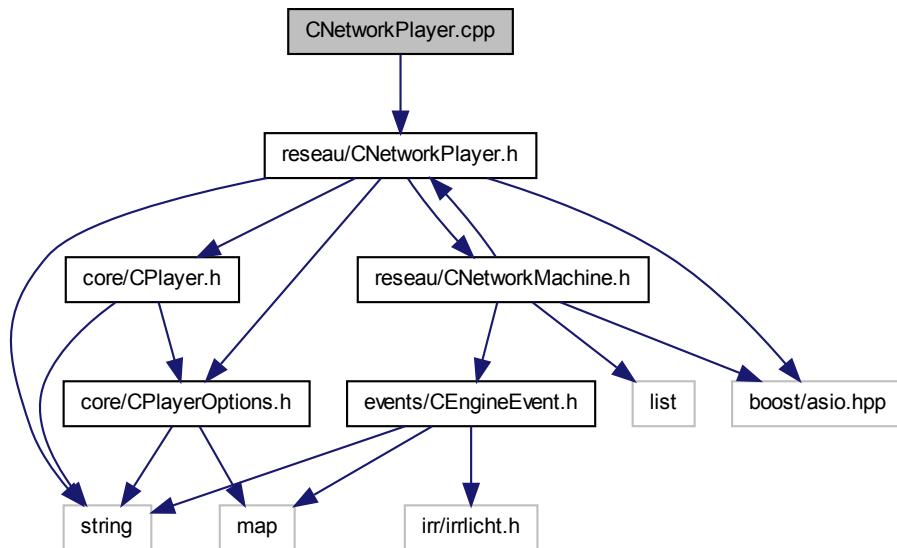
Declaration de la classe CNetworkMachine

8.64 Référence du fichier CNetworkPlayer.cpp

Declaration de la classe CNetworkPlayer.

```
#include "reseau/CNetworkPlayer.h"
```

Graphe des dépendances par inclusion de CNetworkPlayer.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.64.1 Description détaillée

Declaration de la classe CNetworkPlayer.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

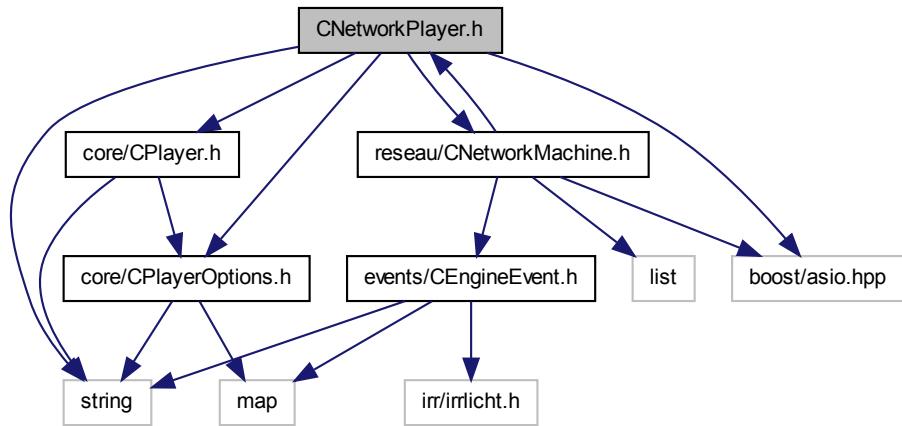
Declaration de la classe CNetworkPlayer

8.65 Référence du fichier CNetworkPlayer.h

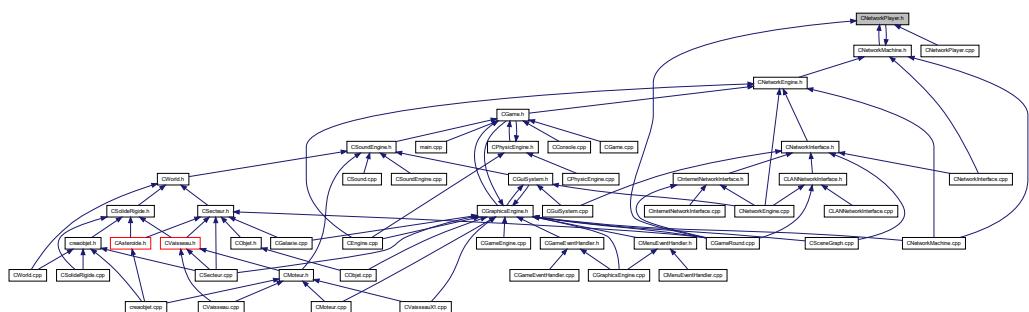
Declaration de la classe CNetworkPlayer.

```
#include <string>
#include <boost/asio.hpp>
#include "core/CPlayerOptions.h"
#include "core/CPlayer.h"
#include "reseau/CNetworkMachine.h"
```

Graphe des dépendances par inclusion de CNetworkPlayer.h :



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



Classes

- class **ayoaron** : **:reseau** : **:CNetworkPlayer**
l'objet joueur réseau

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **:reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.65.1 Description détaillée

Declaration de la classe CNetworkPlayer.

Auteur

Argawaen

Version

0.1

Date

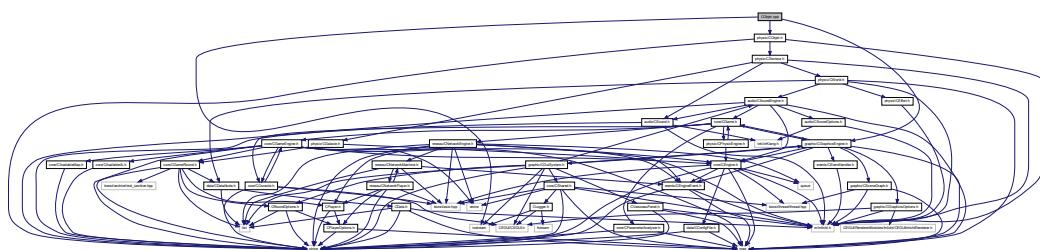
16 février 2011

Declaration de la classe CNetworkPlayer

8.66 Référence du fichier CObjet.cpp

```
#include "physic/CObjet.h"
#include "graphic/CGraphicsEngine.h"
#include <iostream>
```

Graphe des dépendances par inclusion de CObjet.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::physic**

8.66.1 Description détaillée**Date**

9 juin 2011

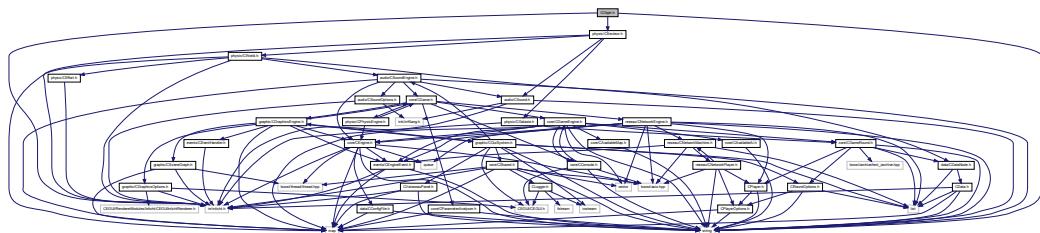
Auteur

argawaen

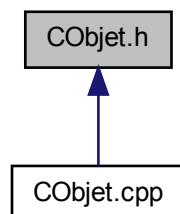
8.67 Référence du fichier CObjet.h

```
#include <irr/irrlicht.h>
#include <string>
#include "physic/CSecteur.h"
```

Graphe des dépendances par inclusion de CObjet.h :



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



Classes

- class **ayoaron** : **:physic** : **:CObjet**
Classe de base de tous les objets.

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **:physic**

8.67.1 Description détaillée**Date**

9 juin 2011

Auteur

argawaen

Date

14 sept. 2011

Auteur

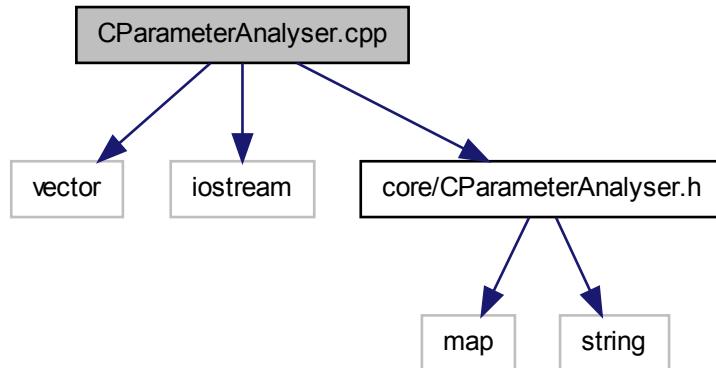
dla

8.68 Référence du fichier CParameterAnalyser.cpp

Implémentation de la classe d'analyse de paramètres.

```
#include <vector>
#include <iostream>
#include "core/CParameterAnalyser.h"
```

Graphe des dépendances par inclusion de CParameterAnalyser.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) : [:core](#)
espace de nommage contenant tous les objets de base

8.68.1 Description détaillée

Implémentation de la classe d'analyse de paramètres.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

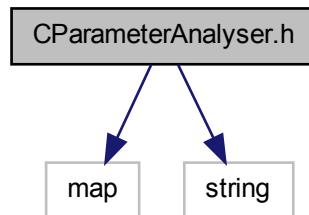
Permet l'analyse des paramètres d'entrée de l'exécutable

8.69 Référence du fichier CParameterAnalyser.h

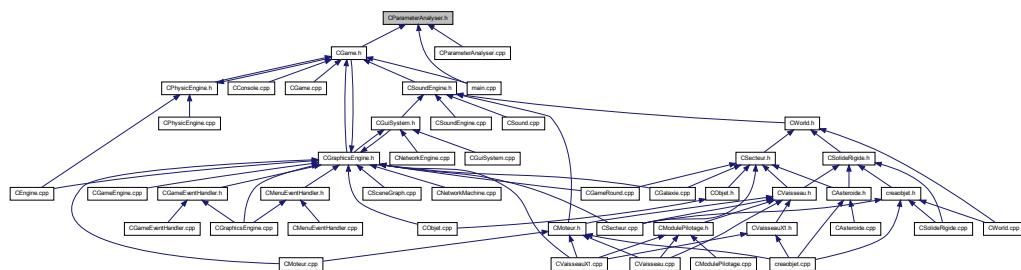
Declaration de la classe d'analyse de paramètres.

```
#include <map>
#include <string>
```

Graphe des dépendances par inclusion de CParameterAnalyser.h :



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



Classes

- class [ayoaron](#) ::core ::CParameterAnalyser
Classe d'analyse des paramètres.

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron :core**
espace de nomage contenant tous les objets de base

8.69.1 Description détaillée

Declaration de la classe d'analyse de paramètres.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

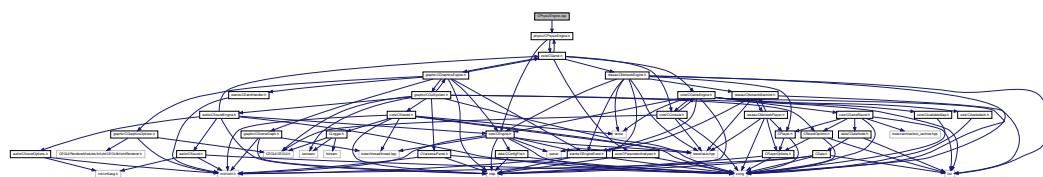
Permet l'analyse des paramètres d'entrée de l'exécutable

8.70 Référence du fichier CPhysicEngine.cpp

fichier de description du poteur de physique

```
#include "physic/CPhysicEngine.h"
```

Graphe des dépendances par inclusion de CPhysicEngine.cpp :



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) : [:physic](#)

8.70.1 Description détaillée

fichier de description du poteur de physique

Date

3 août 2011

Auteur

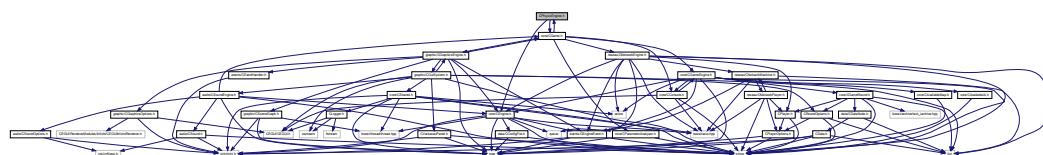
dla

8.71 Référence du fichier CPhysicEngine.h

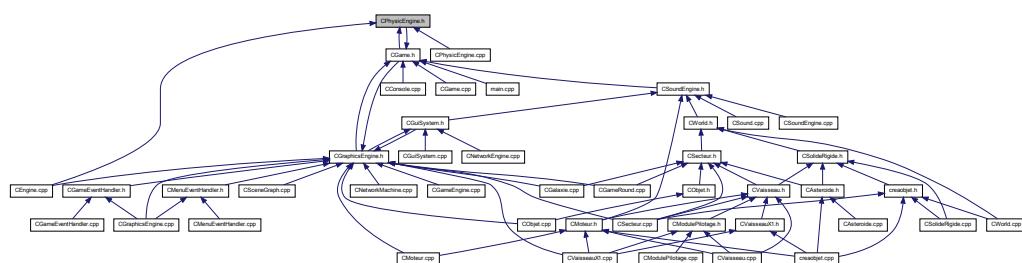
fichier de description du poteur de physique

```
#include "core/CEngine.h"
#include "core/CGame.h"
```

Graphe des dépendances par inclusion de CPhysicEngine.h :



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



Classes

- class **ayoaron** : **:physic** : **:CPhysicEngine**
classe de description du moteur de physique

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **:physic**

8.71.1 Description détaillée

fichier de description du poteur de physique

Date

3 août 2011

Auteur

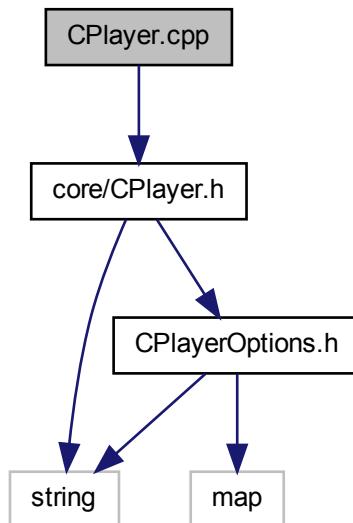
dla

8.72 Référence du fichier CPlayer.cpp

Implementation de la classe CPlayer.

```
#include "core/CPlayer.h"
```

Graphe des dépendances par inclusion de CPlayer.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron ::core**
espace de nommage contenant tous les objets de base

8.72.1 Description détaillée

Implementation de la classe CPlayer.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

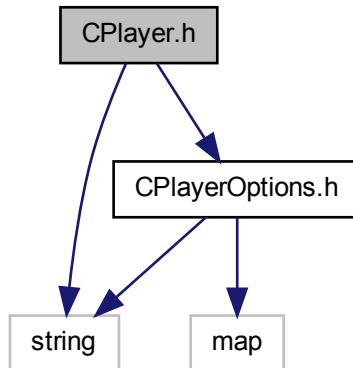
Implementation de la classe CPlayer

8.73 Référence du fichier CPlayer.h

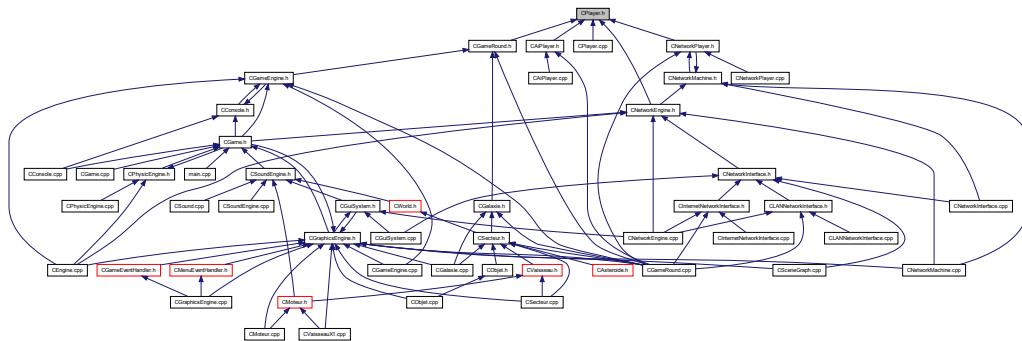
Declaration de la classe player.

```
#include <string>
#include "CPlayerOptions.h"
```

Graphe des dépendances par inclusion de CPlayer.h :



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



Classes

- class **ayoaron** : **:core** : **CPlayer**
l'objet joueur

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** : **:core**
espace de nommage contenant tous les objets de base

8.73.1 Description détaillée

Declaration de la classe player.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

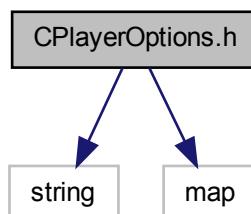
Declaration de la classe player

8.74 Référence du fichier CPlayerOptions.h

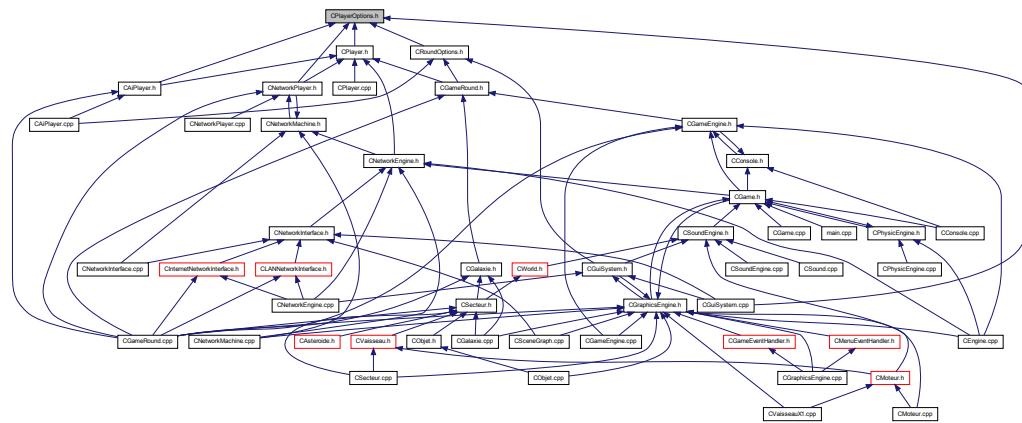
Declaration de la classe CPlayerOptions.

```
#include <string>
#include <map>
```

Graphe des dépendances par inclusion de CPlayerOptions.h :



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



Classes

- class **ayoaron** : **core** : **CPlayerOptions**
rassemblement des options joueur

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **core**
espace de nomage contenant tous les objets de base

8.74.1 Description détaillée

Declaration de la classe CPlayerOptions.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

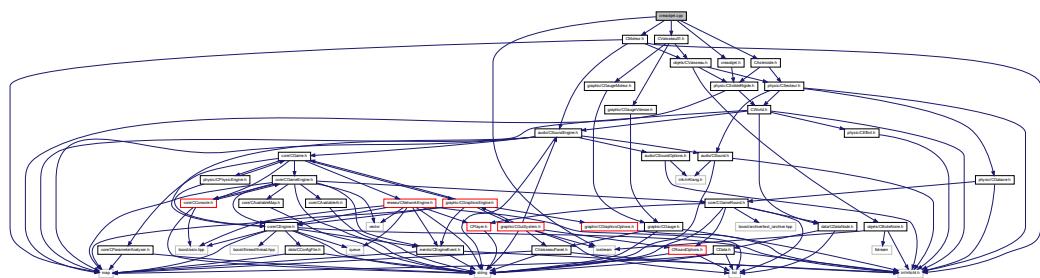
Declaration de la classe CPlayerOptions

8.75 Référence du fichier creaobjet.cpp

```
#include <iostream>
```

```
#include "creaobjet.h"
#include "CAsteroide.h"
#include "CVaisseauX1.h"
#include "CMoteur.h"
```

Graphe des dépendances par inclusion de creaobjet.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::objets**

Fonctions

- **physic** : **CSolideRigide** * **ayoaron::objets::creaobjet** (data : **CDataNode** *donn, physic : **CWorld** *world, physic : **CSolideRigide** *parent)

8.75.1 Description détaillée

Date

15 sept. 2011

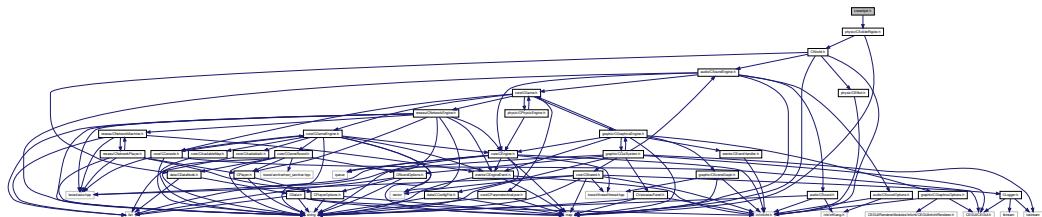
Auteur

dla

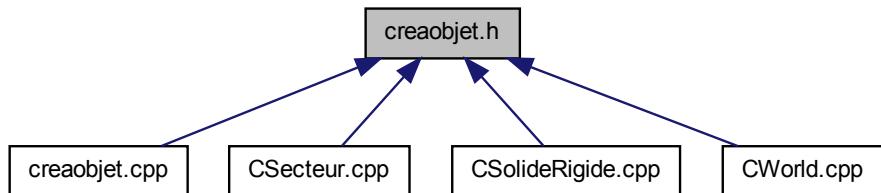
8.76 Référence du fichier creaobjet.h

```
#include "physic/CSolideRigide.h"
```

Graphe des dépendances par inclusion de creaobjet.h :



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



Espaces de nommage

- namespace [ayoaron](#)
l'espace de nommage de base du projet ayoaron
- namespace [ayoaron](#) : [:objets](#)

Fonctions

- physic : [:CSolideRigide](#) * [ayoaron](#) : [:objets](#) : [:creaobjet](#) (data : [:CDataNode](#) *donn, physic : [:CWorld](#) *world, physic : [:CSolideRigide](#) *parent)

8.76.1 Description détaillée

Date

15 sept. 2011

Auteur

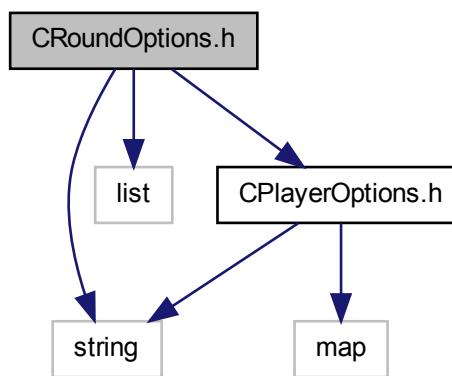
dla

8.77 Référence du fichier CRoundOptions.h

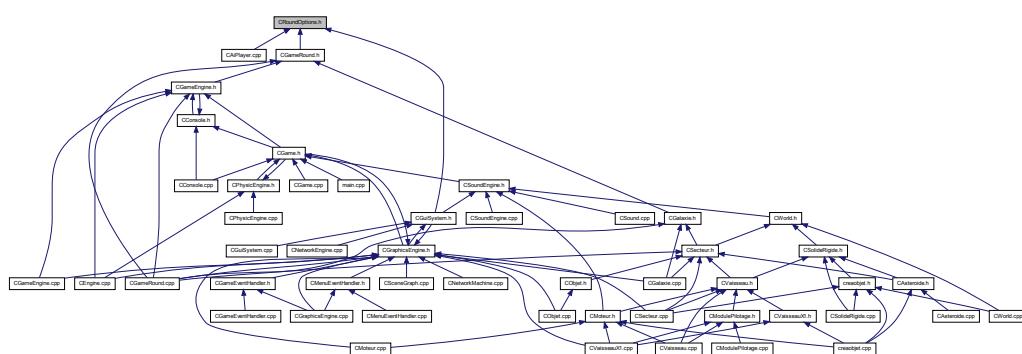
Declaration de la classe CRoundOptions.

```
#include <string>
#include <list>
#include "CPlayerOptions.h"
```

Graphe des dépendances par inclusion de CRoundOptions.h :



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



Classes

- class [ayoaron](#) : [core](#) : [CRoundOptions](#)

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nommage contenant tous les objets de base

8.77.1 Description détaillée

Declaration de la classe CRoundOptions.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

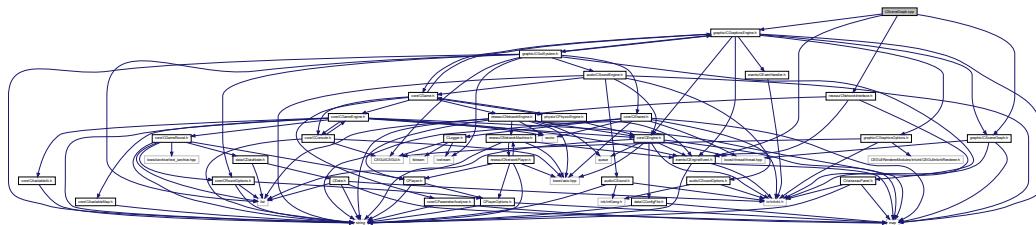
Declaration de la classe CRoundOptions

8.78 Référence du fichier CSceneGraph.cpp

Declaration de la classe CSceneGraph.

```
#include "graphic/CSceneGraph.h"
#include "graphic/CGraphicsEngine.h"
#include "reseau/CNetworkInterface.h"
#include "events/CEngineEvent.h"
```

Graphe des dépendances par inclusion de CSceneGraph.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::graphic**
espace de nommage contenant tous les objets destiné au moteur graphique

8.78.1 Description détaillée

Declaration de la classe CSceneGraph.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

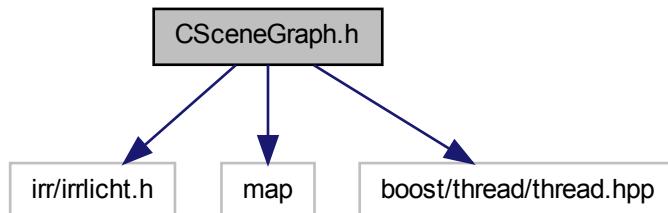
Declaration de la classe CSceneGraph

8.79 Référence du fichier CSceneGraph.h

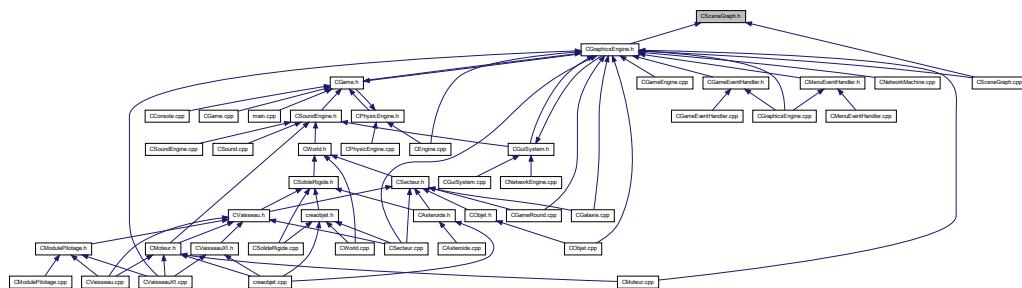
Declaration de la classe CSceneGraph.

```
#include <irr/irrlicht.h>
#include <map>
#include <boost/thread/thread.hpp>
```

Graphe des dépendances par inclusion de CSceneGraph.h :



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



Classes

- class **ayoaron** : **:graphic** : **:CSceneGraph**
le graph de scène

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :graphic**
espace de nomage contenant tous les objets destiné au moteur graphique
 - namespace **ayoaron : :reseau**
espace de nomage contenant tous les objets destiné au moteur réseau

8.79.1 Description détaillée

Declaration de la classe CSceneGraph.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

Declaration de la classe CSceneGraph

8.80 Référence du fichier CSecteur.cpp

la classe décrivant un secteur de la galaxie

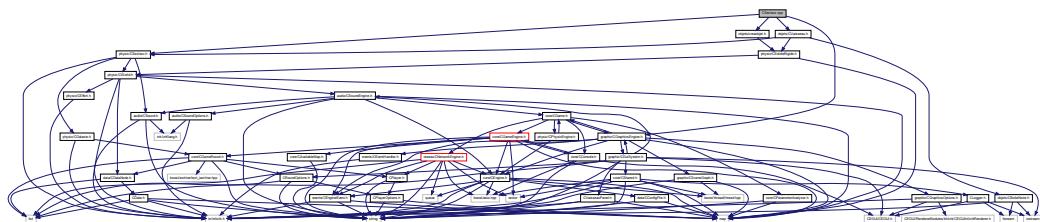
```
#include "physic/CSecteur.h"
```

```
#include "graphic/CGraphicsEngine.h"
```

```
#include "objets/CVaisseau.h"
```

```
#include "objets/creaobjet.h"
```

Graphe des dépendances par inclusion de CSecteur.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::physic**

8.80.1 Description détaillée

la classe décrivant un secteur de la galaxie

Date

4 juin 2011

Auteur

argawaen

8.81 Référence du fichier CSecteur.h

la classe décrivant un secteur de la galaxie

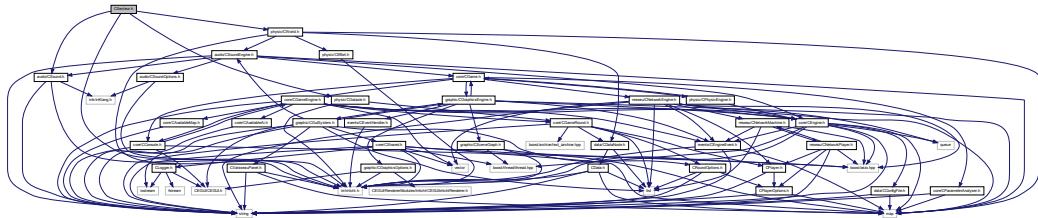
```
#include "physic/CGalaxie.h"
```

```
#include "physic/CWorld.h"
```

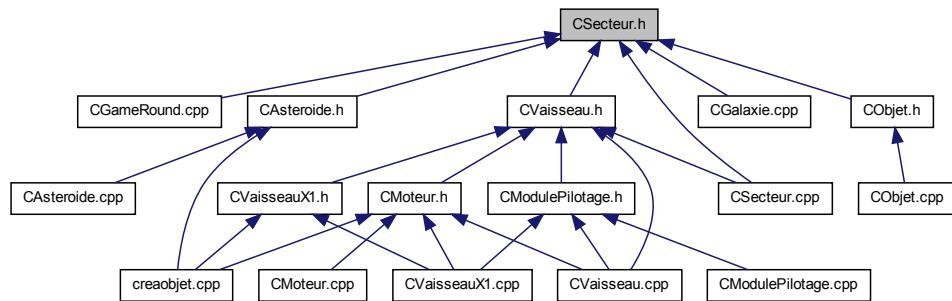
```
#include <irr/irrlicht.h>
```

```
#include "audio/CSound.h"
```

Graphe des dépendances par inclusion de CSecteur.h :



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



Classes

- class **ayoaron** : **:physic** : **:CSecteur**
la classe décrivant un secteur de la galaxie

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron** : **:objets**
- namespace **ayoaron** : **:physic**

8.81.1 Description détaillée

la classe décrivant un secteur de la galaxie

Date

4 juin 2011

Auteur

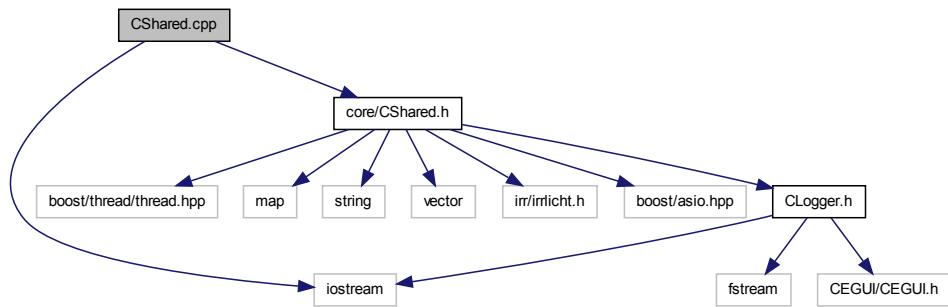
argawaen

8.82 Référence du fichier CShared.cpp

Implémentation de la classe CShared et CSharedInstance.

```
#include <iostream>
#include "core/CShared.h"
```

Graphe des dépendances par inclusion de CShared.cpp :

**Espaces de nommage**

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron::core**
espace de nommage contenant tous les objets de base

8.82.1 Description détaillée

Implémentation de la classe CShared et CSharedInstance.

Auteur

Argawaen

Version

0.1

Date

16 février 2011

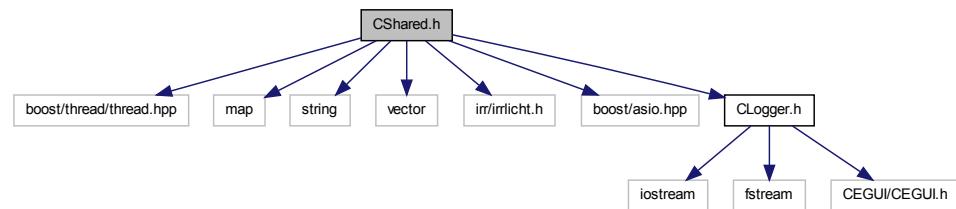
Implémentation de la classe CShared et CShared_Instance

8.83 Référence du fichier CShared.h

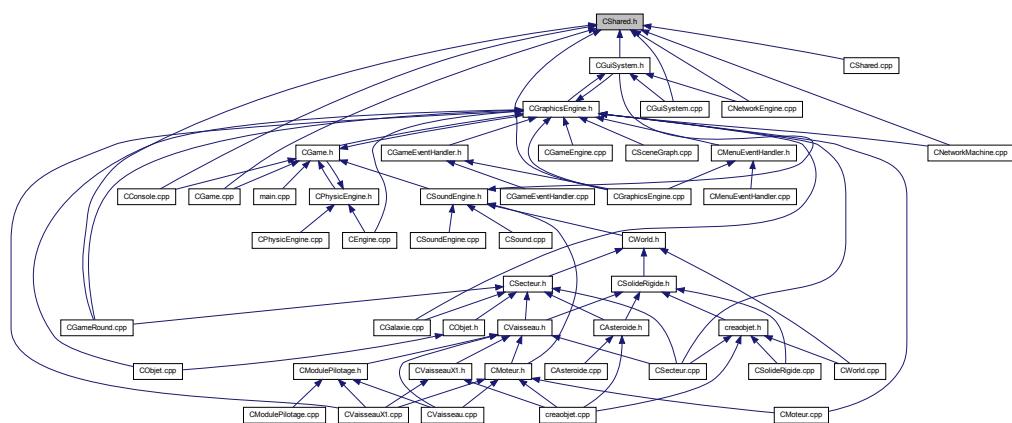
Declaration de la classe CShared et CSharedInstance.

```
#include <boost/thread/thread.hpp>
#include <map>
#include <string>
#include <vector>
#include <irr/irrlicht.h>
#include <boost/asio.hpp>
#include "CLogger.h"
```

Graphe des dépendances par inclusion de CShared.h :



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



Classes

- class **ayoaron** ::core ::CShared
singleton
- class **ayoaron** ::core ::CSharedInstance
donnée stockée dans le singleton

Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** ::core
espace de nommage contenant tous les objets de base

8.83.1 Description détaillée

Declaration de la classe CShared et CSharedInstance.

Auteur

Argawaen

Version

0.1

Date

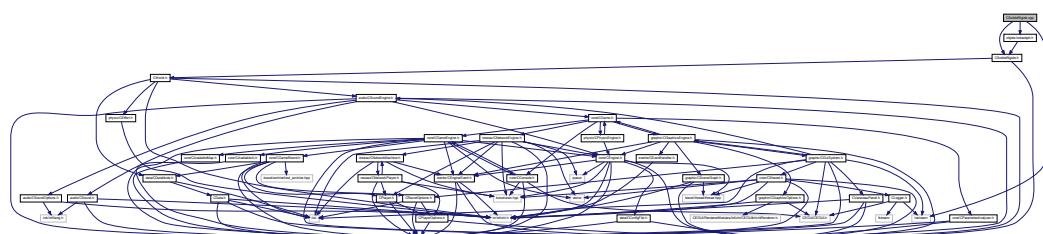
16 février 2011

Declaration de la classe CShared et CSharedInstance

8.84 Référence du fichier CSolideRigide.cpp

```
#include "CSolideRigide.h"  
#include "objets/creaobjet.h"  
#include <iostream>
```

Graphe des dépendances par inclusion de CSolideRigide.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nommage de base du projet ayoaron
- namespace **ayoaron** : **:physic**

8.84.1 Description détaillée

Date

14 sept. 2011

Auteur

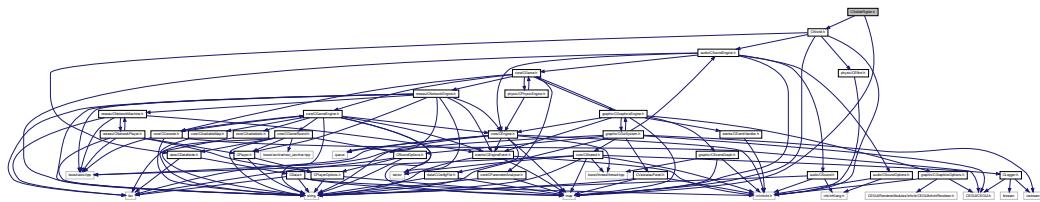
dla

8.85 Référence du fichier CSolideRigide.h

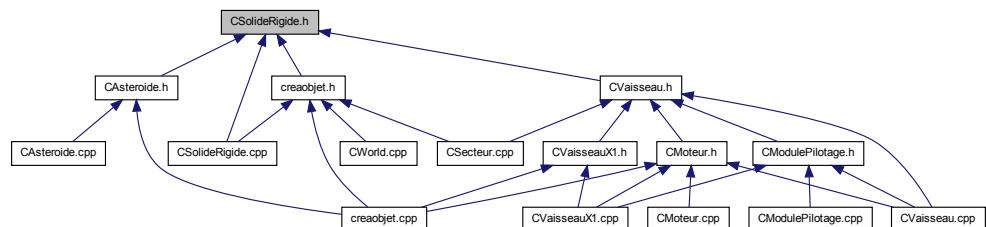
```
#include "CWorld.h"
```

```
#include <map>
```

Graphe des dépendances par inclusion de CSolideRigide.h :



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



Classes

- class **ayoaron** : **:physic** : **:CSolideRigide**
une classe d'objet physique

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::physic**

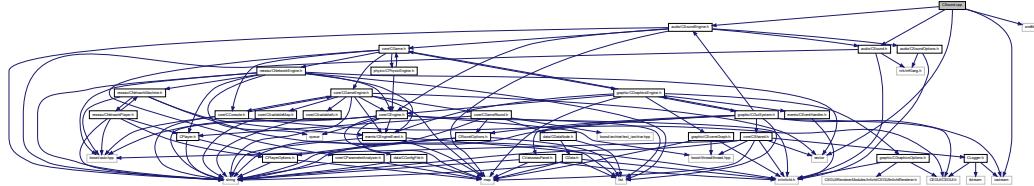
Énumérations

- enum **ayoaron::physic::REPERE** { **ayoaron::physic::LOCAL**, **ayoaron::physic::PARENT**, **ayoaron::physic::GENERAL** }

8.86 Référence du fichier CSound.cpp

```
#include "audio/CSound.h"
#include "audio/CSoundEngine.h"
#include <vector>
#include <sndfile.h>
#include <iostream>
```

Graphe des dépendances par inclusion de CSound.cpp :



Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
- namespace **ayoaron::audio**
espace de nomage contenant tous les objets destiné au moteur de son

8.86.1 Description détaillée

Date

12 août 2011

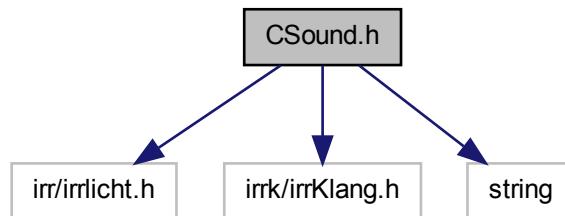
Auteur

dla

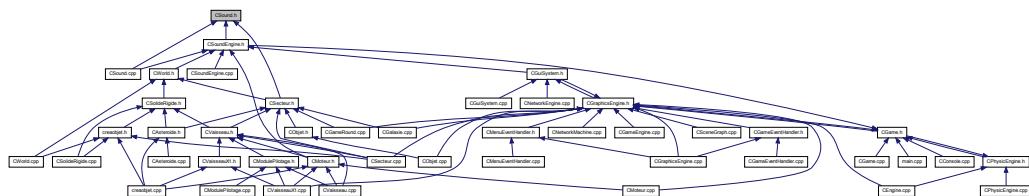
8.87 Référence du fichier CSound.h

```
#include <irr/irrlicht.h>
#include <irrk/irrklang.h>
#include <string>
```

Graphe des dépendances par inclusion de CSound.h :



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



Classes

- class ayoaron ::audio ::CSound

Espaces de nommage

- namespace **ayoaron**
l'espace de nomage de base du projet ayoaron
 - namespace **ayoaron : :audio**
espace de nomage contenant tous les objets destiné au moteur de son

Énumérations

- ```
- enum ayoaron : :audio : :SOUND_TYPE { ayoaron : :audio : :SND_2D_INSTANT,
 ayoaron : :audio : :SND_2D_MUSIC, ayoaron : :audio : :SND_3D_CONTINUOUS,
 ayoaron : :audio : :SND_3D_INSTANT }
```

### 8.87.1 Description détaillée

Date

12 août 2011

## Auteur

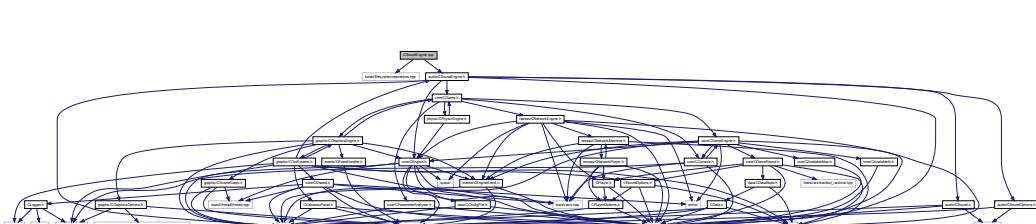
argawaen

## 8.88 Référence du fichier CSoundEngine.cpp

## Declaration de la classe CSoundEngine.

```
#include <boost/filesystem/operations.hpp>
```

```
#include "audio/CSoundEngine.h"
```



## Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
  - namespace **ayoaron : :audio**  
*espace de nomage contenant tous les objets destiné au moteur de son*

### 8.88.1 Description détaillée

## Declaration de la classe CSoundEngine.

## Auteur

## Argawaen

## Version

0.1

**Date**

16 février 2011

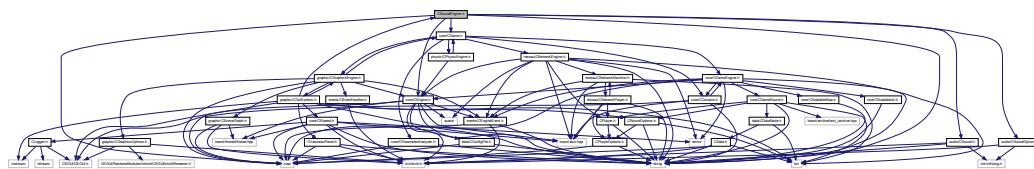
Declaration de la classe CSoundEngine

**8.89 Référence du fichier CSoundEngine.h**

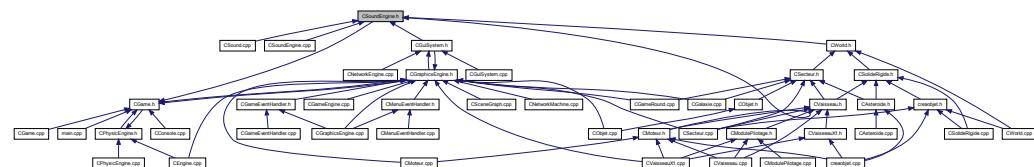
Declaration de la classe CSoundEngine.

```
#include "core/CEngine.h"
#include "core/CGame.h"
#include <map>
#include <string>
#include "audio/CSoundOptions.h"
#include "audio/CSound.h"
```

Graphe des dépendances par inclusion de CSoundEngine.h :



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

**Classes**

- class [ayoaron](#) ::[audio](#) ::[CSoundEngine](#)  
*le moteur de son*

**Espaces de nommage**

- namespace [ayoaron](#)  
*l'espace de nommage de base du projet ayoaron*
- namespace [ayoaron](#) ::[audio](#)  
*espace de nommage contenant tous les objets destiné au moteur de son*

### 8.89.1 Description détaillée

## Declaration de la classe CSoundEngine.

## Auteur

## Argawaen

## Version

0.1

Date

16 février 2011

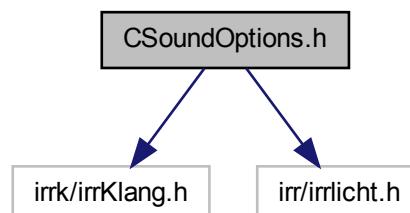
## Declaration de la classe CSoundEngine

## 8.90 Référence du fichier CSoundOptions.h

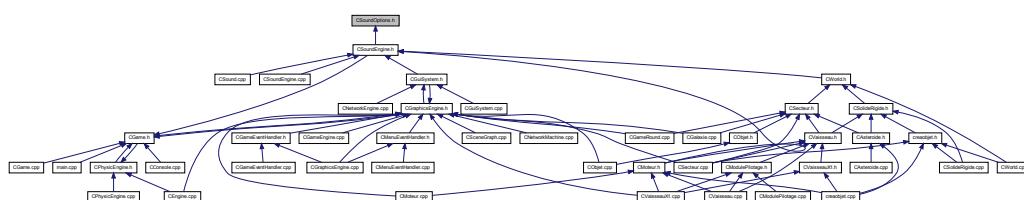
```
#include <irrk/irrkKlang.h>
```

```
#include <irr/irrlicht.h>
```

## Graphe des dépendances par inclusion de CSoundOptions.h :



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



### Classes

- class **ayoaron** : **audio** : **CSoundOptions**  
*classe des options sonore*

### Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
- namespace **ayoaron** : **audio**  
*espace de nomage contenant tous les objets destiné au moteur de son*

#### 8.90.1 Description détaillée

##### Date

15 août 2011

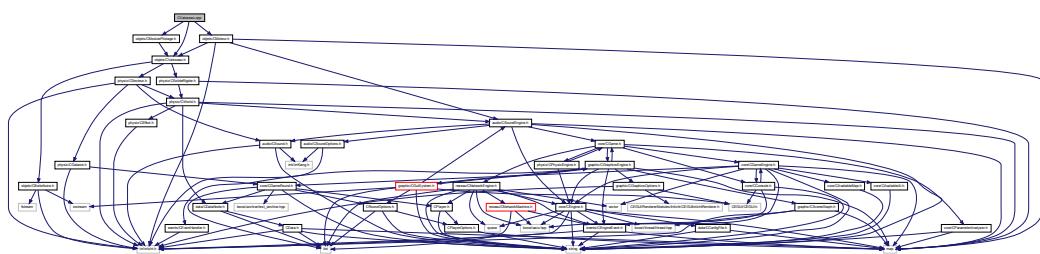
##### Auteur

argawaen

## 8.91 Référence du fichier CVaisseau.cpp

```
#include "objets/CVaisseau.h"
#include "objets/CMoteur.h"
#include "objets/CModulePilotage.h"
```

Graphe des dépendances par inclusion de CVaisseau.cpp :



### Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
- namespace **ayoaron** : **objets**

### Fonctions

- irr : **f32** **ayoaron** : **objets** : **vmin** (irr : **f32** v, irr : **f32** eps, irr : **f32** VM)

## 8.91.1 Description détaillée

**Date**

14 juil. 2011

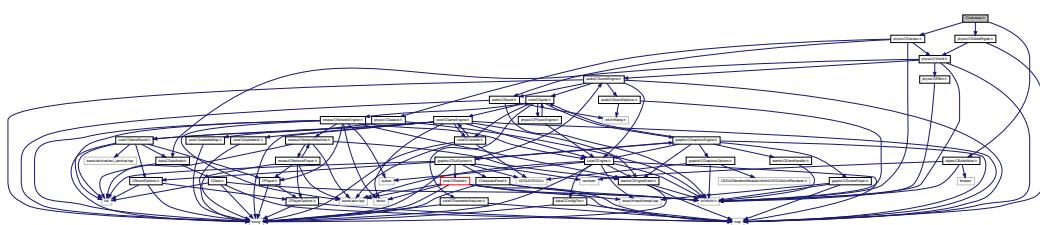
**Auteur**

dla

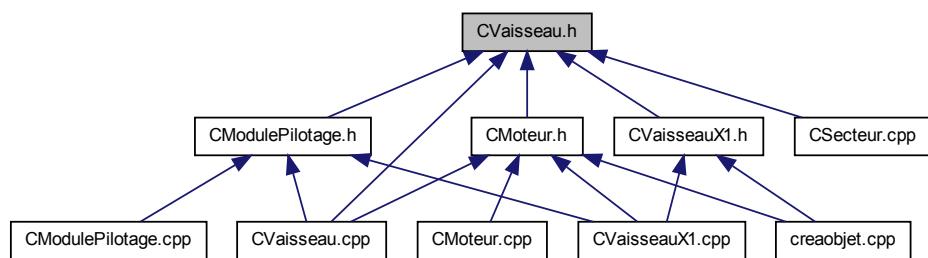
## 8.92 Référence du fichier CVaisseau.h

```
#include "physic/CSecteur.h"
#include "physic/CSolideRigide.h"
#include "objets/CBoiteNoire.h"
```

Graphe des dépendances par inclusion de CVaisseau.h :



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

**Classes**

- class [ayoaron](#) : [objets](#) : [CVaisseau](#)

*Classe de base de description d'un vaisseau quelconque.*

**Espaces de nommage**

- namespace [ayoaron](#)  
*l'espace de nommage de base du projet ayoaron*
- namespace [ayoaron](#) : [:objets](#)

**8.92.1 Description détaillée****Date**

14 juil. 2011

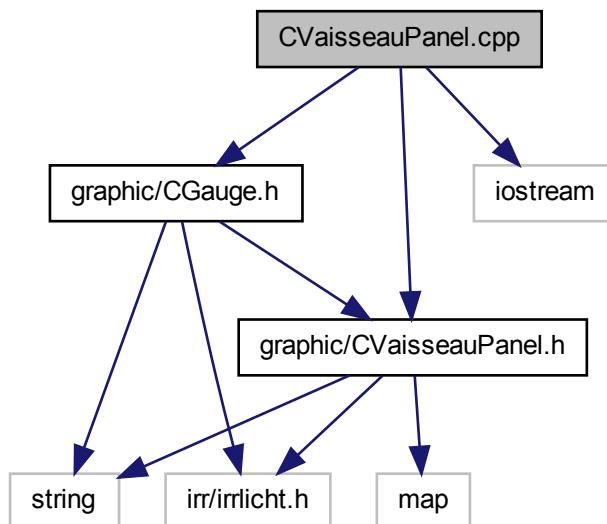
**Auteur**

dla

**8.93 Référence du fichier CVaisseauPanel.cpp**

```
#include "graphic/CVaisseauPanel.h"
#include "graphic/CGauge.h"
#include <iostream>
```

Graphe des dépendances par inclusion de CVaisseauPanel.cpp :



**Espaces de nommage**

- namespace **ayoaron**  
*l'espace de nommage de base du projet ayoaron*
- namespace **ayoaron::graphic**  
*espace de nommage contenant tous les objets destiné au moteur graphique*

**8.93.1 Description détaillée****Date**

23 août 2011

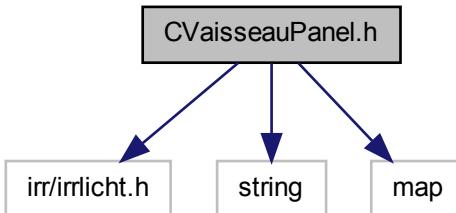
**Auteur**

dla

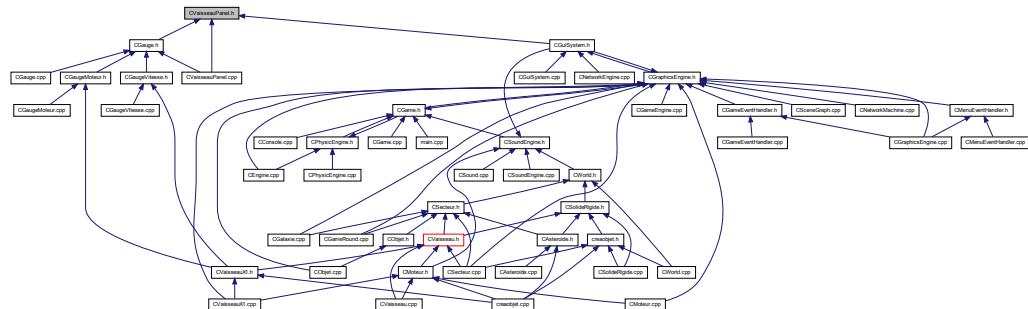
**8.94 Référence du fichier CVaisseauPanel.h**

```
#include <irr/irrlicht.h>
#include <string>
#include <map>
```

Graphe des dépendances par inclusion de CVaisseauPanel.h :



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



## Classes

- class **ayoaron : :graphic : :CVaisseauPanel**  
*une classe englobante permettant le dessin complet du tableau de bord*

## Espaces de nommage

- namespace **ayoaron**  
*l'espace de nommage de base du projet ayoaron*
- namespace **ayoaron : :graphic**  
*espace de nommage contenant tous les objets destiné au moteur graphique*

### 8.94.1 Description détaillée

#### Date

23 août 2011

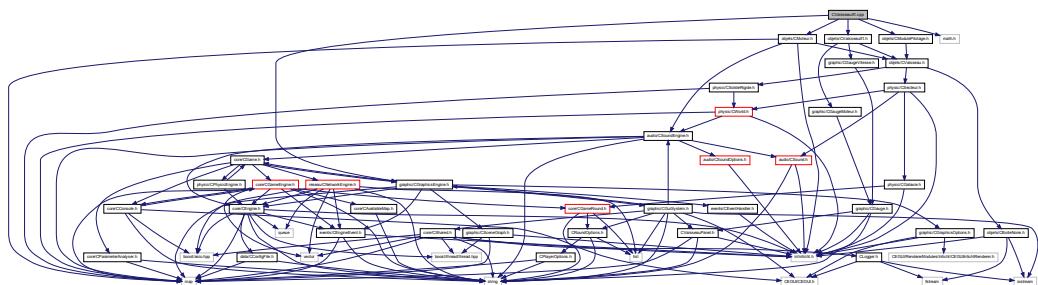
#### Auteur

dla

## 8.95 Référence du fichier CVaisseauX1.cpp

```
#include "objets/CVaisseauX1.h"
#include "objets/CModulePilotage.h"
#include "graphic/CGraphicsEngine.h"
#include "objets/CMoteur.h"
#include <math.h>
```

Graphe des dépendances par inclusion de CVaisseauX1.cpp :



### Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
- namespace **ayoaron::objets**

### Fonctions

- irr : **:f32 ayoaron::objets::mround** (irr : **:f32 a**, irr : **:u32 off**)

#### 8.95.1 Description détaillée

##### Date

18 juin 2011

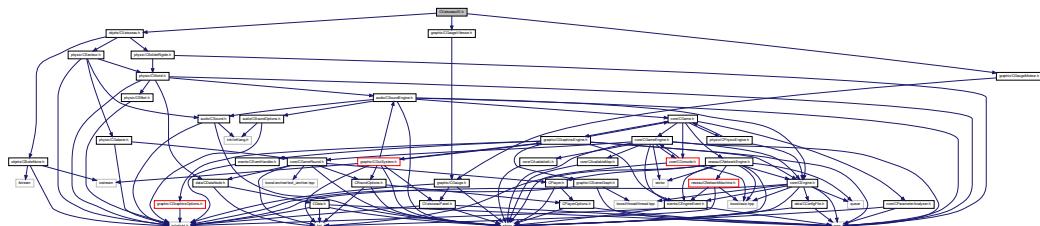
##### Auteur

argawaen

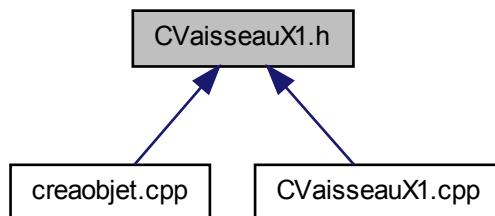
### 8.96 Référence du fichier CVaisseauX1.h

```
#include "objets/CVaisseau.h"
#include "graphic/CGaugeVitesse.h"
#include "graphic/CGaugeMoteur.h"
```

Graphe des dépendances par inclusion de CVaisseauX1.h :



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



### Classes

- class [ayoaron](#) ::objets ::CVaisseauX1

### Espaces de nommage

- namespace [ayoaron](#)  
*l'espace de nommage de base du projet ayoaron*
- namespace [ayoaron](#) ::objets

#### 8.96.1 Description détaillée

##### Date

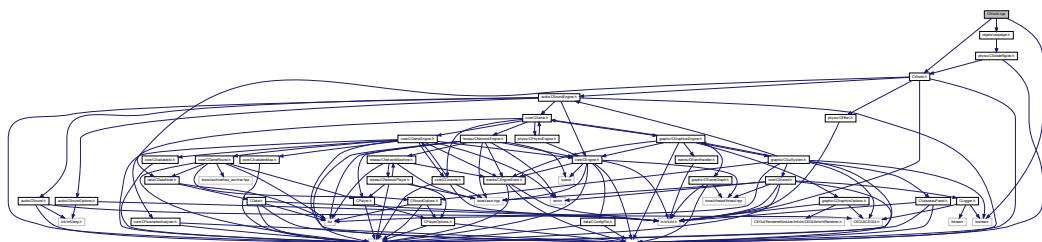
18 juin 2011

##### Auteur

argawaen

## 8.97 Référence du fichier CWorld.cpp

```
#include "objets/creaobjet.h"
#include "CWorld.h"
#include <iostream>
```



## Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
  - namespace **ayoaron :physic**

### 8.97.1 Description détaillée

Date

14 sept. 2011

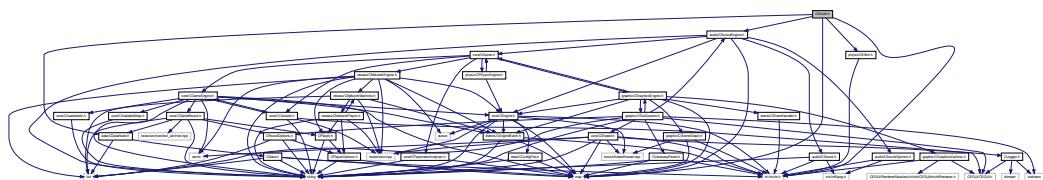
## Auteur

dla

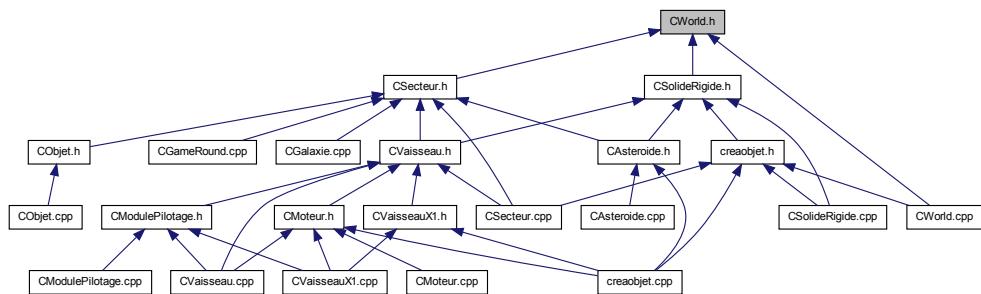
## 8.98 Référence du fichier CWorld.h

```
#include <irr/irrlicht.h>
#include <map>
#include "data/CDataNode.h"
#include "physic/CEffort.h"
#include "audio/CSoundEngine.h"
```

Graphe des dépendances par inclusion de CWorld.h :



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



## Classes

- class **ayoaron** : **:physic** : **:CWorld**  
*Classe de définition d'un monde d'objet.*

## Espaces de nommage

- namespace **ayoaron**  
*l'espace de nommage de base du projet ayoaron*
- namespace **ayoaron** : **:physic**  
*espace de nommage relatif à la physique du jeu*
- namespace **physique**

### 8.98.1 Description détaillée

#### Date

14 sept. 2011

#### Auteur

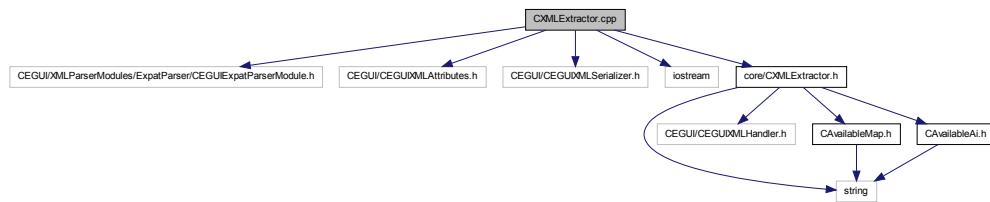
dla

## 8.99 Référence du fichier CXMLExtractor.cpp

Déclaration de la classe CXMLExtractor.

```
#include <CEGUI/XMLParserModules/ExpatParser/CEGUIExpatParserModule.h>
#include <CEGUI/CEGUIXMLAttributes.h>
#include <CEGUI/CEGUIXMLSerializer.h>
#include <iostream>
#include "core/CXMLExtractor.h"
```

Graphe des dépendances par inclusion de CXMLExtractor.cpp :



### Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
- namespace **ayoaron :core**  
*espace de nomage contenant tous les objets de base*

#### 8.99.1 Description détaillée

Déclaration de la classe CXMLExtractor.

##### Auteur

Argawaen

##### Version

0.1

##### Date

16 février 2011

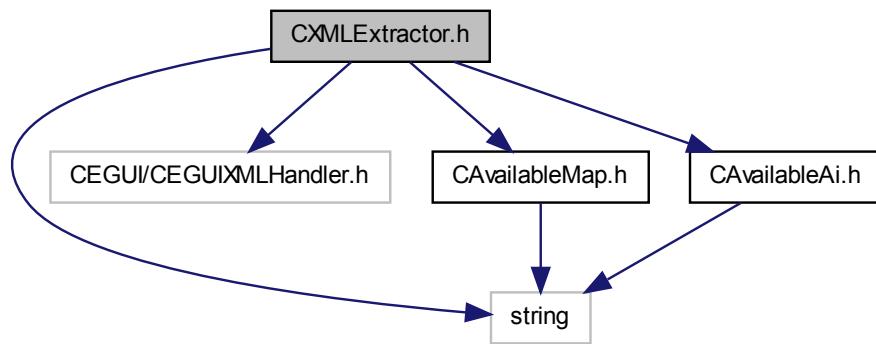
Déclaration de la classe CXMLExtractor

## 8.100 Référence du fichier CXMLExtractor.h

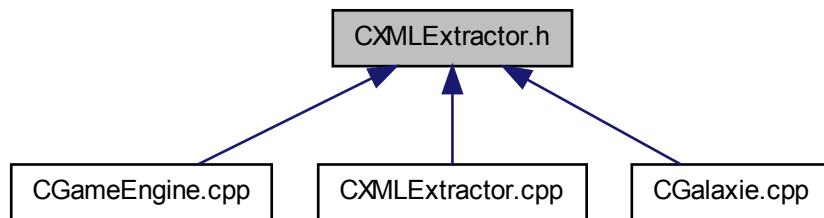
Déclaration de la classe CXMLExtractor.

```
#include <string>
#include <CEGUI/CEGUIXMLHandler.h>
#include "CAvailableMap.h"
#include "CAvailableAi.h"
```

Graphe des dépendances par inclusion de CXMLExtractor.h :



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



### Classes

- class [ayoaron](#) : [core](#) : CXMLExtractor
   
cree les interfaces XML

### Espaces de nommage

- namespace **ayoaron**  
*l'espace de nomage de base du projet ayoaron*
- namespace **ayoaron::core**  
*espace de nomage contenant tous les objets de base*

#### 8.100.1 Description détaillée

Déclaration de la classe CXMLExtractor.

##### Auteur

Argawaen

##### Version

0.1

##### Date

16 février 2011

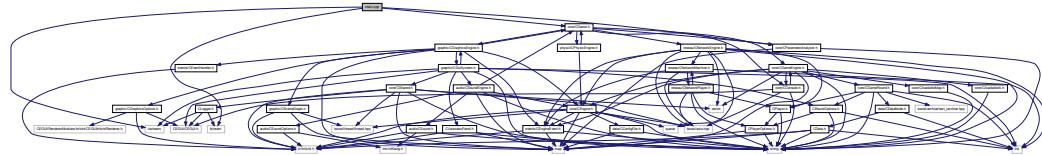
Déclaration de la classe CXMLExtractor

## 8.101 Référence du fichier main.cpp

Partie principale du programme.

```
#include <iostream>
#include <fstream>
#include "core/CGame.h"
#include "core/CParameterAnalyser.h"
```

Graphe des dépendances par inclusion de main.cpp :



### Fonctions

- int **main** (int argc, char \*\*argv)  
*point d'entrée de l'exécutable.*

### 8.101.1 Description détaillée

Partie principale du programme.

#### Auteur

Argawaen

#### Version

0.1

#### Date

16 février 2011

Partie principale du programme crée une classe jeu et la lance

### 8.101.2 Documentation des fonctions

#### 8.101.2.1 int main ( int argc, char \*\* argv )

point d'entrée de l'exécutable.

#### Paramètres

|    |      |                               |
|----|------|-------------------------------|
| in | argc | Nombre de mot dans argv       |
| in | argv | Pointeur sur la liste de mots |

#### Renvoie

Arrêt nomrmal du programme

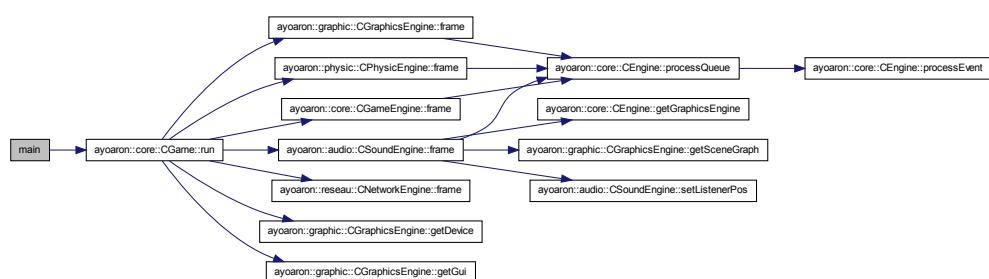
redirection de cout et cerr dans des fichiers

création de la classe jeux et initialisation des moteurs

lancement de la boucle principale

Références ayoaron ::core ::CGame ::run().

Voici le graphe d'appel pour cette fonction :



## Index

- ~CAsteroide
  - ayoaron : :objets : :CAsteroide, 286
- ~CBoiteNoire
  - ayoaron : :objets : :CBoiteNoire, 319
- ~CConfigFile
  - ayoaron : :data : :CConfigFile, 128
- ~CConsole
  - ayoaron : :core : :CConsole, 55
- ~CData
  - ayoaron : :data : :CData, 132
- ~CDataNode
  - ayoaron : :data : :CDataNode, 149
- ~CEffort
  - ayoaron : :physic : :CEffort, 454
- ~CEngine
  - ayoaron : :core : :CEngine, 62
- ~CEventHandler
  - ayoaron : :event : :CEventHandler, 190
- ~CFileManager
  - ayoaron : :data : :CFileManager, 177
- ~CGalaxie
  - ayoaron : :physic : :CGalaxie, 458
- ~CGame
  - ayoaron : :core : :CGame, 76
- ~CGameEngine
  - ayoaron : :core : :CGameEngine, 83
- ~CGameEventHandler
  - ayoaron : :event : :CGameEventHandler, 193
- ~CGameRound
  - ayoaron : :core : :CGameRound, 102
- ~CGauge
  - ayoaron : :graphic : :CGauge, 203
- ~CGaugeMoteur
  - ayoaron : :graphic : :CGaugeMoteur, 209
- ~CGaugeVitesse
  - ayoaron : :graphic : :CGaugeVitesse, 217
- ~CGraphicsEngine
  - ayoaron : :graphic : :CGraphicsEngine, 229
- ~CGraphicsOptions
  - ayoaron : :graphic : :CGraphicsOptions, 249
- ~CGuiSystem
  - ayoaron : :graphic : :CGuiSystem, 252
- ~CIInternetNetworkInterface
  - ayoaron : :reseau : :CIInternetNetworkInterface, 542
- ~CLANNetworkInterface
  - ayoaron : :reseau : :CLANNetworkInterface, 548
- ~CLogger
  - ayoaron : :core : :CLogger, 109
- ~CMenuEventHandler
  - ayoaron : :event : :CMenuEventHandler, 198
- ~CModulePilotage
  - ayoaron : :objets : :CModulePilotage, 322
- ~CMoteur
  - ayoaron : :objets : :CMoteur, 329
- ~CNetworkEngine
  - ayoaron : :reseau : :CNetworkEngine, 557
- ~CNetworkMachine
  - ayoaron : :reseau : :CNetworkMachine, 592
- ~CObjet
  - ayoaron : :physic : :CObjet, 465
- ~CPhysicEngine
  - ayoaron : :physic : :CPhysicEngine, 480
- ~CPlayer
  - ayoaron : :core : :CPlayer, 114
- ~CPousMouv
  - ayoaron : :objets : :CPousMouv, 366
- ~CSceneGraph
  - ayoaron : :graphic : :CSceneGraph, 269
- ~CSecteur
  - ayoaron : :physic : :CSecteur, 495
- ~CSharedInstance
  - ayoaron : :core : :CSharedInstance, 125
- ~CSolideRigide
  - ayoaron : :physic : :CSolideRigide, 506
- ~CSound
  - ayoaron : :audio : :CSound, 22
- ~CSoundEngine
  - ayoaron : :audio : :CSoundEngine, 31

~CSoundOptions  
    ayoaron : :audio : :CSoundOptions,  
        48

~CVaisseau  
    ayoaron : :objets : :CVaisseau, 372

~CVaisseauPanel  
    ayoaron : :graphic : :CVaisseauPanel,  
        275

~CVaisseauX1  
    ayoaron : :objets : :CVaisseauX1, 414

~CWorld  
    ayoaron : :physic : :CWorld, 537

~CXMLExtractor  
    ayoaron : :core : :CXMLExtractor, 126

abs  
    ayoaron : :objets : :CPousMouv, 366

accmax  
    ayoaron : :objets : :CModulePilotage,  
        323

ADD\_NODE  
    ayoaron : :event : :CEngineEvent, 187

addAbsoluteEffort  
    ayoaron : :objets : :CAsteriode, 286  
    ayoaron : :objets : :CMoteur, 329  
    ayoaron : :objets : :CVaisseau, 373  
    ayoaron : :objets : :CVaisseauX1, 414  
    ayoaron : :physic : :CSolideRigide, 506

addAnimator  
    ayoaron : :physic : :CObjet, 465

addEnfant  
    ayoaron : :data : :CDataNode, 150  
    ayoaron : :objets : :CAsteriode, 287  
    ayoaron : :objets : :CMoteur, 330  
    ayoaron : :objets : :CVaisseau, 373  
    ayoaron : :objets : :CVaisseauX1, 415  
    ayoaron : :physic : :CSolideRigide, 507

addGauge  
    ayoaron : :graphic : :CVaisseauPanel,  
        275, 276

addLocalEffort  
    ayoaron : :objets : :CAsteriode, 287  
    ayoaron : :objets : :CMoteur, 330  
    ayoaron : :objets : :CVaisseau, 374  
    ayoaron : :objets : :CVaisseauX1, 415  
    ayoaron : :physic : :CSolideRigide, 507

addNode  
    ayoaron : :graphic : :CSceneGraph,  
        269

addNodeAndPropagate

    ayoaron : :graphic : :CSceneGraph,  
        270

addpoussee  
    ayoaron : :objets : :CVaisseau, 375  
    ayoaron : :objets : :CVaisseauX1, 416

addRelativeEffort  
    ayoaron : :objets : :CAsteriode, 288  
    ayoaron : :objets : :CMoteur, 331  
    ayoaron : :objets : :CVaisseau, 375  
    ayoaron : :objets : :CVaisseauX1, 416  
    ayoaron : :physic : :CSolideRigide, 508

address  
    ayoaron : :reseau : :CNetworkMachine,  
        595

addSound  
    ayoaron : :audio : :CSoundEngine, 31

affiche\_perm  
    ayoaron : :graphic : :CGaugeMoteur,  
        213

AI\_EASY  
    ayoaron : :core : :CRoundOptions, 117

AI\_HARD  
    ayoaron : :core : :CRoundOptions, 118

AI\_MEDIUM  
    ayoaron : :core : :CRoundOptions, 117

ai\_path  
    ayoaron : :core : :CGameEngine, 96

aiguille  
    ayoaron : :graphic : :CGaugeVitesse,  
        223

ALL  
    ayoaron : :core : :CLogger, 109

aromax  
    ayoaron : :objets : :CModulePilotage,  
        324

attachGameEngine  
    ayoaron : :audio : :CSoundEngine, 32  
    ayoaron : :core : :CEngine, 62  
    ayoaron : :core : :CGameEngine, 83  
    ayoaron : :graphic : :CGraphicsEngine,  
        230

    ayoaron : :physic : :CPhysicEngine,  
        480

    ayoaron : :reseau : :CNetworkEngine,  
        557

attachGraphicsEngine  
    ayoaron : :audio : :CSoundEngine, 32  
    ayoaron : :core : :CEngine, 62  
    ayoaron : :core : :CGameEngine, 84

ayoaron ::graphic ::CGraphicsEngine,  
230  
ayoaron :: physic ::CPhysicEngine,  
481  
ayoaron :: reseau ::CNetworkEngine,  
557  
attachNetworkEngine  
    ayoaron :: audio ::CSoundEngine, 33  
    ayoaron :: core ::CEngine, 63  
    ayoaron :: core ::CGameEngine, 84  
    ayoaron :: graphic ::CGraphicsEngine,  
        230  
    ayoaron :: physic ::CPhysicEngine,  
        481  
    ayoaron :: reseau ::CNetworkEngine,  
        558  
attachPhysicEngine  
    ayoaron :: audio ::CSoundEngine, 33  
    ayoaron :: core ::CEngine, 63  
    ayoaron :: core ::CGameEngine, 85  
    ayoaron :: graphic ::CGraphicsEngine,  
        231  
    ayoaron :: physic ::CPhysicEngine,  
        482  
    ayoaron :: reseau ::CNetworkEngine,  
        558  
attachPlayer  
    ayoaron :: reseau ::CNetworkMachine,  
        592  
attachSoundEngine  
    ayoaron :: audio ::CSoundEngine, 34  
    ayoaron :: core ::CEngine, 64  
    ayoaron :: core ::CGameEngine, 85  
    ayoaron :: graphic ::CGraphicsEngine,  
        231  
    ayoaron :: physic ::CPhysicEngine,  
        482  
    ayoaron :: reseau ::CNetworkEngine,  
        559  
ayoaron, 12  
ayoaron :: audio, 12  
    SND\_2D\_INSTANT, 13  
    SND\_2D\_MUSIC, 13  
    SND\_3D\_CONTINUOUS, 13  
    SND\_3D\_INSTANT, 13  
    SOUND\_TYPE, 13  
ayoaron :: audio ::CSound, 20  
    ~CSound, 22  
    CSound, 22  
    destrSound, 22  
            getPitch, 22  
            getPos, 22  
            getType, 22  
            getVit, 23  
            getVolume, 23  
            isDoomed, 23  
            isPlaying, 23  
            isRepeat, 23  
            loadSound, 23  
            parent, 27  
            playSound, 24  
            pos, 27  
            setPitch, 24  
            setPos, 25  
            setType, 25  
            setVit, 26  
            setVolume, 26  
            snd, 27  
            sndsource, 27  
            sndtype, 27  
            stopSound, 26  
            vit, 27  
            ayoaron :: audio ::CSoundEngine, 27  
            ~CSoundEngine, 31  
            addSound, 31  
            attachGameEngine, 32  
            attachGraphicsEngine, 32  
            attachNetworkEngine, 33  
            attachPhysicEngine, 33  
            attachSoundEngine, 34  
            cfile, 45  
            CSoundEngine, 31  
            device, 45  
            dropAllSound, 34  
            dropSound, 35  
            engine\_name, 45  
            events\_queue, 45  
            frame, 35  
            ge, 45  
            getConfigData, 36  
            getDevice, 37  
            getGameEngine, 37  
            getGraphicsEngine, 38  
            getNetworkEngine, 38  
            getParent, 39  
            getSound, 39  
            getSoundEngine, 39  
            gffe, 46  
            l\_snd, 46  
            ne, 46

options, 46  
parent, 46  
passive\_mode, 46  
phy, 46  
processEvent, 40  
processQueue, 40  
pushEvent, 41  
queue\_mutex, 47  
se, 47  
sendMessageToGame, 41  
sendMessageToGraphics, 42  
sendMessageToNetwork, 43  
sendMessageToPhysic, 43  
sendMessageToSound, 44  
setDopplerProperties, 44  
setListenerPos, 44  
smgr, 47  
state, 47  
STATE\_LOADING\_ROUND, 31  
STATE\_NOTHING, 31  
STATE\_PLAYING, 31  
STATE\_PLAYING\_INTRO, 31  
STATE\_QUITTING, 31  
STATE\_WAITING\_PLAYERS, 31  
state\_mutex, 47  
ayoaron : :audio : :CSoundOptions, 48  
  ~CSoundOptions, 48  
  CSoundOptions, 48  
  distfac, 48  
  doplfac, 49  
  maxdist, 49  
  mindist, 49  
  rolloff, 49  
  volume, 49  
ayoaron : :core, 13  
ayoaron : :core : :CAiPlayer, 49  
  CAiPlayer, 51  
  getName, 51  
ayoaron : :core : :CAvailableAi, 51  
  script\_filename, 52  
ayoaron : :core : :CAvailableMap, 52  
  description, 53  
  mesh\_file, 53  
  name, 54  
ayoaron : :core : :CConsole, 54  
  ~CConsole, 55  
  CConsole, 55  
  handleAcceptTCP, 55  
  io, 57  
  parent, 58  
          processCommand, 56  
          s, 58  
          serverThreadTCPReceive, 57  
          tcp\_acceptor, 58  
ayoaron : :core : :CEngine, 58  
  ~CEngine, 62  
  attachGameEngine, 62  
  attachGraphicsEngine, 62  
  attachNetworkEngine, 63  
  attachPhysicEngine, 63  
  attachSoundEngine, 64  
  CEngine, 62  
  cfile, 72  
  engine\_name, 72  
  events\_queue, 72  
  frame, 64  
  ge, 72  
  getConfigData, 64  
  getGameEngine, 65  
  getGraphicsEngine, 65  
  getNetworkEngine, 66  
  getParent, 66  
  getSoundEngine, 67  
  gffe, 72  
  ne, 72  
  parent, 73  
  passive\_mode, 73  
  phy, 73  
  processEvent, 67  
  processQueue, 68  
  pushEvent, 68  
  queue\_mutex, 73  
  se, 73  
  sendMessageToGame, 69  
  sendMessageToGraphics, 70  
  sendMessageToNetwork, 70  
  sendMessageToPhysic, 71  
  sendMessageToSound, 71  
  state, 73  
  STATE\_LOADING\_ROUND, 61  
  STATE\_NOTHING, 61  
  STATE\_PLAYING, 61  
  STATE\_PLAYING\_INTRO, 61  
  STATE\_QUITTING, 61  
  STATE\_WAITING\_PLAYERS, 61  
  state\_mutex, 74  
ayoaron : :core : :CGame, 74  
  ~CGame, 76  
  CGame, 75  
  DeltaTemps, 78

g, 78  
getDt, 76  
gfx, 78  
init, 77  
n, 79  
p, 79  
phy, 79  
run, 77  
s, 79  
still\_running, 79  
still\_running\_mutex, 79  
stoooooop, 78  
then, 79  
ayoaron : :core : :CGameEngine, 79  
  ~CGameEngine, 83  
  ai\_path, 96  
  attachGameEngine, 83  
  attachGraphicsEngine, 84  
  attachNetworkEngine, 84  
  attachPhysicEngine, 85  
  attachSoundEngine, 85  
  bot\_names, 96  
  cfile, 96  
  CGameEngine, 83  
  collectAvailableAis, 86  
  collectAvailableMaps, 86  
  cons, 96  
  engine\_name, 96  
  events\_queue, 96  
  frame, 86  
  framect, 97  
  ge, 97  
  getConfigData, 87  
  getConsole, 87  
  getGameEngine, 87  
  getGraphicsEngine, 88  
  getMapPath, 88  
  getNetworkEngine, 89  
  getParent, 89  
  getSoundEngine, 90  
  gixe, 97  
  gr, 97  
  isPaused, 90  
  l\_ais, 97  
  l\_maps, 97  
  makeAiName, 90  
  map\_path, 97  
  ne, 98  
  parent, 98  
  passive\_mode, 98  
    pause, 98  
    phy, 98  
    processEvent, 91  
    processQueue, 91  
    pushEvent, 92  
    queue\_mutex, 98  
    se, 98  
    sendMessageToGame, 92  
    sendMessageToGraphics, 93  
    sendMessageToNetwork, 94  
    sendMessageToPhysic, 94  
    sendMessageToSound, 95  
    state, 99  
      STATE\_LOADING\_ROUND, 83  
      STATE\_NOTHING, 83  
      STATE\_PLAYING, 83  
      STATE\_PLAYING\_INTRO, 83  
      STATE\_QUITTING, 83  
      STATE\_WAITING\_PLAYERS, 83  
      state\_mutex, 99  
      togglePause, 95  
      wip, 99  
  ayoaron : :core : :CGameRound, 99  
    ~CGameRound, 102  
    CGameRound, 101  
    donn, 106  
    frame, 102  
    galax, 106  
    isFull, 103  
    l\_players, 106  
    load, 103  
    loadMap, 104  
    loadPlayers, 105  
    makeFromCollect, 105  
    ON\_CREATION, 101  
    ON\_FINISH, 101  
    ON\_NOTHING, 101  
    ON\_PLAYING, 101  
    parent, 106  
    ro, 107  
    state, 107  
  ayoaron : :core : :CLogger, 107  
    ~CLogger, 109  
    ALL, 109  
    CLogger, 109  
    f, 110  
    log, 109  
    logEvent, 109  
    ONLY\_ERRORS, 109  
    setLogFilename, 109

type, 110  
WARNING\_OR\_ERRORS, 109  
ayoaron : core : CParameterAnalyser, 110  
    CParameterAnalyser, 112  
    data, 112  
ayoaron : core : CPlayer, 112  
    ~CPlayer, 114  
    CPlayer, 114  
    getName, 114  
    name, 114  
ayoaron : core : CPlayerOptions, 115  
    f\_data, 116  
    i\_data, 116  
    IA\_PLAYER, 116  
    level, 116  
    NETWORK\_PLAYER, 116  
    player\_name, 116  
    player\_type, 116  
    s\_data, 116  
    SELF, 116  
    team\_nb, 116  
ayoaron : core : CRoundOptions, 117  
    AI\_EASY, 117  
    AI\_HARD, 118  
    AI\_MEDIUM, 117  
    clear, 118  
    CRoundOptions, 118  
    l\_difficulties, 118  
    l\_po, 118  
    map\_config, 118  
    map\_name, 118  
    play\_as, 118  
ayoaron : core : CShared, 119  
    CShared, 121  
    data, 123  
    get, 121  
    make, 122  
    unmake, 122  
ayoaron : core : CSharedInstance, 123  
    ~CSharedInstance, 125  
    CSharedInstance, 125  
    log, 125  
    mutex\_gui, 125  
ayoaron : core : CXMLExtractor, 125  
    ~CXMLExtractor, 126  
    CXMLExtractor, 126  
    getAi, 126  
    getMap, 126  
    hdl, 127  
ayoaron : data, 14  
    BOOL, 15  
    COLOR, 15  
    DATATYPE, 14  
    FLOAT, 15  
    MATRIX, 15  
    NOTYPE, 15  
    SINT, 15  
    str2dtyp, 15  
    STRING, 15  
    strtowchar, 15  
    UINT, 15  
    VECTOR, 15  
    wchartostr, 16  
ayoaron : data : CConfigFile, 127  
    ~CConfigFile, 128  
    CConfigFile, 128  
    data, 130  
    getData, 128  
    isIn, 128  
    isOfType, 129  
    parse, 129  
ayoaron : data : CData, 130  
    ~CData, 132  
    CData, 132  
    data, 146  
    drop, 133  
    getbool, 133  
    getcol, 134  
    getDataList, 134  
    getDataType, 135  
    getf64, 136  
    getmat4, 137  
    gets32, 137  
    getstr, 138  
    getStrFromData, 138  
    getu32, 139  
    getvec3, 140  
    setData, 140–144  
    setDataFromStr, 145  
ayoaron : data : CDataNode, 146  
    ~CDataNode, 149  
    addEnfant, 150  
    CDataNode, 149  
    createEnfant, 150  
    donn, 172  
    drop, 150  
    dropEnfant, 151  
    getbool, 151  
    getcol, 152  
    getData, 153

getDownList, 154  
getDownListCompared, 155  
getDataType, 156  
getEnfant, 157  
getf64, 157  
getIteratorEnfantBegin, 158  
getIteratorEnfantEnd, 159  
getmat4, 159  
getNbEnfant, 160  
getNiveau, 161  
getObjetListCompared, 161  
getParent, 162  
getRoot, 163  
gets32, 163  
getstr, 164  
getu32, 165  
getvec3, 166  
l\_enfant, 172  
niveau, 172  
parent, 172  
root, 173  
setData, 167–171  
setParent, 171  
ayoaron : :data : :CDataStructure, 173  
l\_bool, 174  
l\_color, 174  
l\_f64, 174  
l\_mat4, 174  
l\_s32, 174  
l\_str, 175  
l\_u32, 175  
l\_vec3, 175  
ayoaron : :data : :CFileManager, 175  
~CFileManager, 177  
CFileManager, 177  
config, 183  
filesys, 183  
getConfig, 177  
getDataTypeWstr, 177  
readConfigFile, 178  
readGalaxFile, 179  
readObjectFile, 179  
readSecteurFile, 180  
writeGalaxFile, 181  
writeObjectFile, 181  
writeSavegame, 182  
writeSecteurFile, 182  
ayoaron : :data : :SConfigFileData, 184  
f\_data, 184  
i\_data, 185  
s\_data, 185  
ayoaron : :event, 16  
ayoaron : :event : :CEngineEvent, 185  
ADD\_NODE, 187  
CHOSEN\_MAP, 187  
DEL\_PLAYER, 187  
GAME\_HAS\_STARTED, 186  
GAME\_PROP, 186  
i\_data, 187  
LOAD, 187  
MOVE\_NODE, 187  
MSG\_NICK, 186  
NEW\_PLAYER, 186  
operator==, 187  
REFUSED, 186  
REM\_NODE, 187  
REMOVE\_MACHINE, 187  
RUN\_MULTI, 186  
s\_data, 187  
SEND\_GAME\_PROP, 186  
serialize, 187  
type, 188  
v\_data, 188  
WELCOME, 186  
ayoaron : :event : :CEventHandler, 188  
~CEventHandler, 190  
CEventHandler, 189  
getParent, 190  
HasJoystick, 191  
isKeyPressed, 190  
joy, 191  
KeyIsPressed, 191  
mouse, 191  
OnEvent, 191  
onEvent, 190  
parent, 192  
ayoaron : :event : :CGameEventHandler,  
192  
~CGameEventHandler, 193  
CGameEventHandler, 194  
getParent, 194  
HasJoystick, 195  
isKeyPressed, 194  
joy, 196  
KeyIsPressed, 196  
mouse, 196  
OnEvent, 195  
onEvent, 194  
parent, 196  
ayoaron : :event : :CMenuEventHandler, 196

~CMenuEventHandler, 198  
CMenuEventHandler, 198  
getParent, 198  
HasJoystick, 199  
isKeyPressed, 198  
joy, 199  
KeyIsPressed, 200  
mouse, 200  
OnEvent, 199  
onEvent, 199  
parent, 200  
ayoaron : :event : :CSerializableVector3df,  
    200  
    CSerializableVector3df, 201  
    serialize, 201  
ayoaron : :graphic, 17  
ayoaron : :graphic : :CGauge, 201  
    ~CGauge, 203  
    background, 206  
    CGauge, 203  
    draw, 204  
    panel, 206  
    pos, 206  
    scale, 206  
    setBackground, 204  
    setPosition, 205  
    setScale, 205  
ayoaron : :graphic : :CGaugeMoteur, 207  
    ~CGaugeMoteur, 209  
    affiche\_perm, 213  
    background, 213  
    CGaugeMoteur, 209  
    draw, 209  
    moteurs, 213  
    Nmoteur, 213  
    panel, 213  
    pos, 213  
    scale, 214  
    setBackground, 210  
    setCadres, 210  
    setNmoteur, 211  
    setPosition, 211  
    setRegime, 212  
    setScale, 212  
    valsmot, 214  
ayoaron : :graphic : :CGaugeVitesse, 214  
    ~CGaugeVitesse, 217  
    aiguille, 223  
    background, 223  
    CGaugeVitesse, 216  
            draw, 217  
            limang, 223  
            limits, 223  
            panel, 224  
            pos, 224  
            recGRXY, 224  
            recGRZ, 224  
            recGVX, 224  
            recGVY, 224  
            recGVZ, 224  
            scale, 224  
            setBackground, 217  
            setCurseur, 218  
            setLimit, 218  
            setLimitAngulaire, 219  
            setPosition, 219  
            setRectGRXY, 220  
            setRectGRZ, 220  
            setRectGVX, 221  
            setRectGVY, 221  
            setRectGVZ, 222  
            setScale, 222  
            vite, 225  
            vrot, 225  
            ayoaron : :graphic : :CGraphicsEngine, 225  
                ~CGraphicsEngine, 229  
                attachGameEngine, 230  
                attachGraphicsEngine, 230  
                attachNetworkEngine, 230  
                attachPhysicEngine, 231  
                attachSoundEngine, 231  
                cfile, 244  
                CGraphicsEngine, 229  
                driver, 244  
                engine\_name, 245  
                events\_queue, 245  
                frame, 232  
                ge, 245  
                getConfigData, 233  
                getDbgLevel, 234  
                getDevice, 234  
                getDriver, 235  
                getGameEngine, 235  
                getGraphicsEngine, 236  
                getGui, 236  
                getIrrRenderer, 237  
                getNetworkEngine, 237  
                getParent, 237  
                getSceneGraph, 238  
                getSoundEngine, 238

gixe, 245  
gui, 245  
idevice, 245  
irrRenderer, 245  
l\_eh, 246  
l\_meshs, 246  
lastFPS, 246  
ne, 246  
ON\_GAME, 229  
ON\_LOADING, 229  
ON\_MENUS, 229  
options, 246  
parent, 246  
passive\_mode, 246  
phy, 247  
processEvent, 239  
processQueue, 239  
pushEvent, 240  
queue\_mutex, 247  
requestClose, 241  
se, 247  
sendMessageToGame, 241  
sendMessageToGraphics, 242  
sendMessageToNetwork, 242  
sendMessageToPhysic, 243  
sendMessageToSound, 243  
setState, 244  
sg, 247  
state, 247  
STATE\_LOADING\_ROUND, 228  
STATE\_NOTHING, 228  
STATE\_PLAYING, 228  
STATE\_PLAYING\_INTRO, 228  
STATE\_QUITTING, 228  
STATE\_WAITING\_PLAYERS, 228  
state\_mutex, 247  
sys, 248  
ayoaron : :graphic : :CGraphicsOptions, 248  
  ~CGraphicsOptions, 249  
  bits, 249  
  CGraphicsOptions, 249  
  debuglvl, 249  
  drv, 249  
  fullscreen, 249  
  resolution, 249  
  shadows, 249  
  Vsync, 249  
ayoaron : :graphic : :CGuiSystem, 250  
  ~CGuiSystem, 252  
  CGuiSystem, 252  
close, 253  
console\_visible, 267  
displayAide, 253  
displayGameGui, 254  
displayIngameMenu, 254  
displayLoadgame, 255  
displayLoading, 255  
displayMenuPrincipal, 256  
displayOptions, 257  
displaySelectPlayers, 257  
GAME, 252  
gaugepanel, 267  
getMapList, 257  
getPanel, 258  
handleClose, 258  
handleDisplayAideIG, 258  
handleDisplayAideMP, 259  
handleDisplayLoadgame, 259  
handleDisplayLoading, 260  
handleDisplayOptionsIG, 260  
handleDisplayOptionsMP, 261  
handleProcessCommand, 261  
handleRetourToIngame, 262  
handleRetourToMain, 262  
INGAME\_MENU, 252  
ingame\_menu, 267  
isConsoleVisible, 263  
isPanelVisible, 263  
LOAD\_MENU, 252  
loadGame, 263  
MAIN\_MENU, 252  
onClic, 264  
OPTIONS\_MENU, 252  
panel, 267  
parent, 267  
render, 265  
SAVE\_MENU, 252  
setParent, 265  
showErrorMessage, 265  
toggleConsole, 266  
toggleIngameMenu, 266  
togglePanel, 266  
txtVaiss, 266  
vectortoString, 266  
ayoaron : :graphic : :CSceneGraph, 267  
  ~CSceneGraph, 269  
  addNode, 269  
  addNodeAndPropagate, 270  
  CSceneGraph, 269  
  moveNode, 270

moveNodeAndPropagate, 271  
o, 273  
parent, 273  
removeNode, 271  
removeNodeAndPropagate, 272  
removeObserver, 272  
render, 272  
setObserver, 272  
smgr, 273  
smgr\_mutex, 273  
tourneCamera, 273  
translateCamera, 273  
ayoaron : :graphic : CVaisseauPanel, 274  
  ~CVaisseauPanel, 275  
  addGauge, 275, 276  
  background, 281  
  compteurs, 281  
  CVaisseauPanel, 275  
  device, 281  
  draw2DImage, 277  
  drawAll, 278  
  drawBackground, 278  
  getGauge, 279  
  getGlobalScale, 279  
  getVideoDriver, 279  
  scale, 281  
  setBackground, 280  
  setOriginalSize, 280  
ayoaron : :graphic : :datamot, 282  
  cadre1, 282  
  cadre2, 282  
  reversible, 282  
  valmot, 282  
ayoaron : :objets, 17  
  creaobjet, 18  
  mrround, 18  
  vmin, 19  
ayoaron : :objets : :CAsteriode, 282  
  ~CAsteriode, 286  
  addAbsoluteEffort, 286  
  addEnfant, 287  
  addLocalEffort, 287  
  addRelativeEffort, 288  
  cam, 314  
  caml, 315  
  camp, 315  
  camup, 315  
  CAsteriode, 286  
  donn, 315  
  dropEnfant, 289  
  forceResultante, 315  
  frame, 289  
  getAbsoluteForceResultante, 290  
  getAbsoluteMomentResultante, 291  
  getAbsolutePosition, 291  
  getAbsoluteRitesse, 292  
  getAbsoluteRotation, 292  
  getAbsoluteVitesse, 293  
  getDonn, 294  
  getEnfant, 294  
  getGlobalInertie, 294  
  getIteratorEnfantBegin, 295  
  getIteratorEnfantEnd, 295  
  getLocalForceResultante, 295  
  getLocalInertie, 295  
  getLocalMomentResultante, 296  
  getLocalRitesse, 296  
  getLocalVitesse, 297  
  getMasse, 298  
  getMatrix3, 299  
  getNbEnfant, 300  
  getNode, 300  
  getRelativeForceResultante, 300  
  getRelativeInertie, 301  
  getRelativeMomentResultante, 302  
  getRelativePosition, 302  
  getRelativeRitesse, 303  
  getRelativeRotation, 303  
  getRelativeVitesse, 304  
  getVector, 305  
  hascamera, 316  
  l\_enfant, 316  
  loadObjetData, 306  
  momentResultant, 316  
  node, 316  
  parent, 317  
  postPhysique, 306  
  prePhysique, 307  
  resetEffort, 307  
  setAbsoluteRitesse, 307  
  setAbsoluteVitesse, 307  
  setLocalRitesse, 308  
  setLocalVitesse, 309  
  setMatrix3, 309  
  setParent, 310  
  setRelativePosition, 311  
  setRelativeRitesse, 311  
  setRelativeRotation, 312  
  setRelativeVitesse, 313  
  setVector, 313

srho, 317  
world, 317  
ayoaron : :objets : :CBoiteNoire, 317  
  ~CBoiteNoire, 319  
  CBoiteNoire, 319  
  f, 320  
  filename, 320  
  frame, 319  
  logstr, 319  
  logvect, 320  
  temps, 320  
ayoaron : :objets : :CModulePilotage, 321  
  ~CModulePilotage, 322  
  accmax, 323  
  aramax, 324  
  calcConsigne, 322  
  CModulePilotage, 322  
  consigne, 324  
  consigne, 324  
  deltavit, 324  
  deltavitrot, 324  
  dt1, 324  
  DTV, 324  
  DTVR, 324  
  parent, 324  
  setConsigne, 323  
  SRV, 325  
  SV, 325  
  temps, 325  
ayoaron : :objets : :CMoteur, 325  
  ~CMoteur, 329  
  addAbsoluteEffort, 329  
  addEnfant, 330  
  addLocalEffort, 330  
  addRelativeEffort, 331  
  cam, 362  
  caml, 362  
  camp, 362  
  camup, 362  
  CMoteur, 329  
  donn, 362  
  dropEnfant, 332  
  flux, 363  
  flux2, 363  
  force, 363  
  forceResultante, 363  
  frame, 332  
  getAbsoluteForceResultante, 333  
  getAbsoluteMomentResultante, 334  
  getAbsolutePosition, 334  
          getAbsoluteRitesse, 335  
          getAbsoluteRotation, 335  
          getAbsoluteVitesse, 336  
          getDonn, 337  
          getEnfant, 337  
          getForce, 337  
          getGlobalInertie, 338  
          getIteratorEnfantBegin, 339  
          getIteratorEnfantEnd, 339  
          getLocalForceResultante, 339  
          getLocalInertie, 339  
          getLocalMomentResultante, 340  
          getLocalRitesse, 340  
          getLocalVitesse, 341  
          getMasse, 342  
          getMatrix3, 342  
          getNbEnfant, 343  
          getNode, 343  
          getPoussee, 344  
          getPousseeMax, 344  
          getPP, 345  
          getPP2, 345  
          getRegime, 345  
          getRelativeForceResultante, 346  
          getRelativeInertie, 347  
          getRelativeMomentResultante, 347  
          getRelativePosition, 348  
          getRelativeRitesse, 348  
          getRelativeRotation, 349  
          getRelativeVitesse, 349  
          getVector, 350  
          hascamera, 363  
          isReversible, 351  
          l\_enfant, 363  
          loadObjetData, 352  
          momentResultant, 364  
          node, 364  
          parent, 364  
          postPhysique, 352  
          PP, 364  
          PP2, 365  
          prePhysique, 353  
          resetEffort, 354  
          setAbsoluteRitesse, 354  
          setAbsoluteVitesse, 354  
          setLocalRitesse, 355  
          setLocalVitesse, 355  
          setMatrix3, 356  
          setParent, 357  
          setRegime, 357

setRelativePosition, 358  
setRelativeRitesse, 359  
setRelativeRotation, 359  
setRelativeVitesse, 360  
setVector, 361  
snd1, 365  
snd2, 365  
temps, 365  
world, 365  
ayoaron : :objets : :CPousMouv, 365  
~CPousMouv, 366  
abs, 366  
CPousMouv, 366  
h, 366  
max, 367  
operator\*, 367  
operator+, 367  
operator-, 367  
operator/, 367  
P, 368  
sum, 367  
ayoaron : :objets : :CVaisseau, 368  
~CVaisseau, 372  
addAbsoluteEffort, 373  
addEnfant, 373  
addLocalEffort, 374  
addpoussee, 375  
addRelativeEffort, 375  
blackbox, 406  
cam, 406  
caml, 406  
camp, 406  
camup, 406  
consigne, 406  
consignerotvitesse, 406  
consignevitesse, 407  
CVaisseau, 372  
defineRegineFM, 375  
donn, 407  
dropEnfant, 376  
DTV, 407  
DTVR, 407  
forceResultante, 407  
frame, 376  
getAbsoluteForceResultante, 377  
getAbsoluteMomentResultante, 378  
getAbsolutePosition, 378  
getAbsoluteRitesse, 379  
getAbsoluteRotation, 379  
getAbsoluteVitesse, 380  
getDonn, 381  
getEnfant, 381  
getGlobalInertie, 381  
getIteratorEnfantBegin, 382  
getIteratorEnfantEnd, 382  
getLocalForceResultante, 382  
getLocalInertie, 382  
getLocalMomentResultante, 383  
getLocalRitesse, 383  
getLocalVitesse, 384  
getMasse, 385  
getMatrix3, 386  
getNbEnfant, 387  
getNode, 387  
getRelativeForceResultante, 387  
getRelativeInertie, 388  
getRelativeMomentResultante, 389  
getRelativePosition, 389  
getRelativeRitesse, 390  
getRelativeRotation, 390  
getRelativeVitesse, 391  
getVector, 392  
hascamera, 407  
l\_enfant, 408  
loadObjetData, 393  
momentResultant, 408  
moteurs, 408  
Nmoteur, 408  
node, 408  
parent, 409  
pilote, 409  
postPhysique, 393  
prePhysique, 393  
resetconsigne, 394  
resetEffort, 394  
resetmouvement, 394  
resetpousse, 395  
setAbsoluteRitesse, 395  
setAbsoluteVitesse, 396  
setLocalRitesse, 396  
setLocalVitesse, 397  
setMatrix3, 398  
setParent, 398  
setRelativePosition, 399  
setRelativeRitesse, 400  
setRelativeRotation, 400  
setRelativeVitesse, 401  
setVector, 402  
toggleconsigne, 403  
vitavar, 403

vitgadr, 403  
vithaba, 404  
vitlace, 404  
vitmax, 409  
vitroul, 405  
vittang, 405  
world, 409  
ayoaron : :objets : CVaisseauX1, 409  
  ~CVaisseauX1, 414  
  addAbsoluteEffort, 414  
  addEnfant, 415  
  addLocalEffort, 415  
  addpoussee, 416  
  addRelativeEffort, 416  
  blackbox, 449  
  cam, 449  
  caml, 449  
  camp, 449  
  camup, 449  
  cntmot, 449  
  cntvit, 450  
  consigne, 450  
  consignerotvitesse, 450  
  consignevitesse, 450  
  CVaisseauX1, 414  
  defineRegineFM, 417  
  donn, 450  
  dropEnfant, 418  
  DTV, 451  
  DTVR, 451  
  forceResultante, 451  
  frame, 418  
  getAbsoluteForceResultante, 419  
  getAbsoluteMomentResultante, 419  
  getAbsolutePosition, 420  
  getAbsoluteRitesse, 420  
  getAbsoluteRotation, 421  
  getAbsoluteVitesse, 422  
  getDonn, 422  
  getEnfant, 423  
  getGlobalInertie, 423  
  getIteratorEnfantBegin, 424  
  getIteratorEnfantEnd, 424  
  getLocalForceResultante, 424  
  getLocalInertie, 424  
  getLocalMomentResultante, 425  
  getLocalRitesse, 425  
  getLocalVitesse, 426  
  getMasse, 427  
  getMatrix3, 427  
  getNbEnfant, 428  
  getNode, 428  
  getRelativeForceResultante, 429  
  getRelativeInertie, 430  
  getRelativeMomentResultante, 430  
  getRelativePosition, 431  
  getRelativeRitesse, 431  
  getRelativeRotation, 432  
  getRelativeVitesse, 432  
  getVector, 433  
  hascamera, 451  
  l\_enfant, 451  
  loadObjetData, 434  
  momentResultant, 451  
  moteurs, 452  
  Nmoteur, 452  
  node, 452  
  parent, 452  
  pilote, 452  
  postPhysique, 435  
  prePhysique, 436  
  resetconsigne, 437  
  resetEffort, 437  
  resetmouvement, 437  
  resetpousse, 438  
  setAbsoluteRitesse, 438  
  setAbsoluteVitesse, 439  
  setLocalRitesse, 439  
  setLocalVitesse, 440  
  setMatrix3, 441  
  setParent, 441  
  setRelativePosition, 442  
  setRelativeRitesse, 443  
  setRelativeRotation, 443  
  setRelativeVitesse, 444  
  setVector, 445  
  toggleconsigne, 446  
  vitavar, 446  
  vitgadr, 447  
  vithaba, 447  
  vitlace, 448  
  vitmax, 452  
  vitroul, 448  
  vittang, 448  
  world, 453  
ayoaron : :physic, 19  
  GENERAL, 19  
  LOCAL, 19  
  PARENT, 19  
  REPERE, 19

ayoaron : :physic : :CEffort, 453  
    ~CEffort, 454  
    CEffort, 454  
    force, 456  
    getDirection, 454  
    getForce, 454  
    getMoment, 455  
    getNorme, 455  
    getPtApplic, 455  
    position, 457  
    setForce, 456  
ayoaron : :physic : :CGalaxie, 457  
    ~CGalaxie, 458  
    CGalaxie, 458  
    donn, 462  
    frame, 458  
    getDonnee, 459  
    getGraphicsEngine, 459  
    getNom, 459  
    getNSecteur, 460  
    getSecteur, 460  
    getSecteurCourant, 460  
    getSoundEngine, 460  
    gfx, 462  
    l\_secteur, 462  
    loadData, 460  
    parent, 462  
    secteurcourant, 462  
    setNom, 461  
    setSecteurCourant, 461  
    sfx, 462  
ayoaron : :physic : :CObjet, 463  
    ~CObjet, 465  
    addAnimator, 465  
    bougeObjet, 465  
    cam, 475  
    caml, 475  
    camp, 475  
    CObjet, 465  
    donn, 475  
    EffortResultant, 476  
    frame, 466  
    getAbsoluteRotVelocity, 467  
    getAbsoluteVelocity, 468  
    getCam, 468  
    getdInertie, 468  
    getdMasse, 468  
    getInertie, 468  
    getLocalRotVelocity, 469  
    getLocalVelocity, 469  
            getMasse, 469  
            getMesh, 470  
            getNode, 470  
            getObjetParent, 470  
            getRelativeRotVelocity, 471  
            getRelativeVelocity, 471  
            getSceneManager, 471  
            getSecteurParent, 471  
            gravityAffective, 476  
            hascamera, 476  
            isResultanteNulle, 476  
            l\_enfant, 476  
            loadObjetData, 472  
            mesh, 476  
            MomentResultant, 476  
            NID, 476  
            node, 476  
            normalMap, 477  
            objparent, 477  
            preFrame, 472  
            resetEffort, 472  
            resetmouvement, 473  
            separent, 477  
            setInertie, 473  
            setPosition, 473  
            setScale, 474  
            setVitesse, 475  
ayoaron : :physic : :CPhysicEngine, 477  
    ~CPhysicEngine, 480  
    attachGameEngine, 480  
    attachGraphicsEngine, 481  
    attachNetworkEngine, 481  
    attachPhysicEngine, 482  
    attachSoundEngine, 482  
    cfile, 490  
    CPhysicEngine, 480  
    engine\_name, 490  
    events\_queue, 491  
    frame, 483  
    ge, 491  
    getConfigData, 483  
    getGameEngine, 483  
    getGraphicsEngine, 484  
    getNetworkEngine, 484  
    getParent, 485  
    getSoundEngine, 485  
    gixe, 491  
    ne, 491  
    parent, 491  
    passive\_mode, 491

phy, 492  
processEvent, 486  
processQueue, 486  
pushEvent, 487  
queue\_mutex, 492  
se, 492  
sendMessageToGame, 487  
sendMessageToGraphics, 488  
sendMessageToNetwork, 489  
sendMessageToPhysic, 489  
sendMessageToSound, 490  
state, 492  
STATE\_LOADING\_ROUND, 480  
STATE\_NOTHING, 480  
STATE\_PLAYING, 480  
STATE\_PLAYING\_INTRO, 480  
STATE\_QUITTING, 480  
STATE\_WAITING\_PLAYERS, 480  
state\_mutex, 492  
ayoaron : :physic : :CSecteur, 493  
  ~CSecteur, 495  
  CSecteur, 494  
  donn, 501  
  frame, 495  
  getDataNode, 496  
  getGfxEngine, 496  
  getGlobalEffort, 497  
  getNom, 497  
  getParent, 498  
  getSceneManager, 498  
  getSfxEngine, 498  
  IDsecteur, 501  
  I\_objets, 501  
  loadData, 499  
  parent, 501  
  setNom, 500  
  smgr, 501  
  vaisseaujoueur, 501  
ayoaron : :physic : :CSolideRigide, 502  
  ~CSolideRigide, 506  
  addAbsoluteEffort, 506  
  addEnfant, 507  
  addLocalEffort, 507  
  addRelativeEffort, 508  
  cam, 533  
  caml, 533  
  camp, 533  
  camup, 533  
  CSolideRigide, 505  
  donn, 533  
  dropEnfant, 509  
  forceResultante, 534  
  frame, 509  
  getAbsoluteForceResultante, 510  
  getAbsoluteMomentResultante, 510  
  getAbsolutePosition, 511  
  getAbsoluteRitesse, 511  
  getAbsoluteRotation, 512  
  getAbsoluteVitesse, 512  
  getDonn, 513  
  getEnfant, 513  
  getGlobalInertie, 514  
  getIteratorEnfantBegin, 514  
  getIteratorEnfantEnd, 514  
  getLocalForceResultante, 514  
  getLocalInertie, 515  
  getLocalMomentResultante, 515  
  getLocalRitesse, 515  
  getLocalVitesse, 516  
  getMasse, 517  
  getMatrix3, 517  
  getNbEnfant, 518  
  getNode, 519  
  getRelativeForceResultante, 519  
  getRelativeInertie, 520  
  getRelativeMomentResultante, 520  
  getRelativePosition, 520  
  getRelativeRitesse, 521  
  getRelativeRotation, 522  
  getRelativeVitesse, 522  
  getVector, 523  
  hascamera, 534  
  I\_enfant, 534  
  loadObjetData, 524  
  momentResultant, 534  
  node, 534  
  parent, 535  
  postPhysique, 525  
  prePhysique, 525  
  resetEffort, 525  
  setAbsoluteRitesse, 526  
  setAbsoluteVitesse, 526  
  setLocalRitesse, 526  
  setLocalVitesse, 527  
  setMatrix3, 528  
  setParent, 528  
  setRelativePosition, 529  
  setRelativeRitesse, 530  
  setRelativeRotation, 530  
  setRelativeVitesse, 531

setVector, 532  
world, 535  
ayoaron : :physic : :CWorld, 535  
~CWorld, 537  
CWorld, 536  
denn, 539  
frame, 537  
getGlobalEffort, 538  
getSceneManager, 538  
l\_objets, 540  
loadData, 539  
smgr, 540  
ayoaron : :reseau, 19  
ayoaron : :reseau : :CInternetNetworkInterface, 540  
~CInternetNetworkInterface, 542  
CInternetNetworkInterface, 542  
getType, 542  
INTERNET, 542  
LAN, 542  
parent, 545  
sendAdd, 542  
sendMove, 543  
sendRem, 543  
sendToAllGameTCP, 544  
sendToAllGameUDP, 545  
sendToServerTCP, 545  
type, 546  
ayoaron : :reseau : :CLANNetworkInterface, 546  
~CLANNetworkInterface, 548  
CLANNetworkInterface, 548  
getType, 548  
INTERNET, 548  
LAN, 548  
parent, 552  
sendAdd, 548  
sendMove, 549  
sendRem, 549  
sendToAllGameTCP, 550  
sendToAllGameUDP, 551  
sendToServerTCP, 551  
type, 552  
ayoaron : :reseau : :CNetworkEngine, 552  
~CNetworkEngine, 557  
attachGameEngine, 557  
attachGraphicsEngine, 557  
attachNetworkEngine, 558  
attachPhysicEngine, 558  
attachSoundEngine, 559  
broadcastGameProposition, 559  
buildInternetInterface, 560  
buildLANInterface, 560  
cfile, 580  
cleanInterface, 560  
clientConnectsToServer, 561  
CNetworkEngine, 556  
engine\_name, 580  
events\_queue, 580  
frame, 561  
ge, 580  
getConfigData, 562  
getGameEngine, 562  
getGraphicsEngine, 563  
getMachineFromAddress, 563  
getMachineFromId, 564  
getMachineFromPlayerName, 564  
getNetworkEngine, 565  
getNextMachineId, 565  
getParent, 565  
getSoundEngine, 566  
getTCPSocket, 566  
gffe, 580  
handleAcceptTCP, 566  
INTERNET\_CLIENT, 556  
INTERNET\_SERVER, 556  
io, 580  
is\_running, 567  
isAddressRegistered, 567  
l\_machines, 581  
l\_receive\_queue, 581  
LAN\_CLIENT, 556  
LAN\_SERVER, 556  
listenGamePropositions, 568  
listenMulti3secondsUDP, 568  
multicast\_address, 581  
multicastReceive, 569  
ne, 581  
network\_buffer, 581  
next\_machine\_id, 581  
ni, 581  
NONE, 556  
parent, 582  
passive\_mode, 582  
phy, 582  
play\_as, 582  
playAsWhat, 569  
port\_multicast, 582  
port\_tcp\_server\_connection, 582  
port\_udp\_reception, 582

processEvent, 570  
processQueue, 570  
processReceivedEvents, 571  
processReceivedEventsQueue, 572  
propositions\_mutex, 582  
pushEvent, 572  
pushReceivedEvent, 573  
queue\_mutex, 582  
queueEventIntoGame, 573  
received\_events, 583  
received\_propositions, 583  
running, 583  
s\_multicast, 583  
s\_tcp, 583  
s\_udp\_in, 583  
se, 583  
sendEventTCP, 573  
sendEventToAll, 574  
sendEventUDP, 574  
sendMessageToGame, 574  
sendMessageToGraphics, 574  
sendMessageToNetwork, 575  
sendMessageToPhysic, 576  
sendMessageToSound, 576  
sendMulticast, 577  
serverThreadTCPReceive, 577  
setPlayAs, 578  
state, 583  
STATE\_LOADING\_ROUND, 556  
STATE\_NOTHING, 556  
STATE\_PLAYING, 556  
STATE\_PLAYING\_INTRO, 556  
STATE\_QUITTING, 556  
STATE\_WAITING\_PLAYERS, 556  
state\_mutex, 583  
tcp\_acceptor, 584  
threadSendMulticast, 578  
udp\_remote\_endpoint, 584  
udpAsyncRead, 579  
ayoaron : :reseau : :CNetworkInterface, 584  
CNetworkInterface, 586  
getType, 586  
INTERNET, 586  
LAN, 586  
parent, 589  
sendAdd, 586  
sendMove, 587  
sendRem, 587  
sendToAllGameTCP, 588  
sendToAllGameUDP, 588  
sendToServerTCP, 589  
type, 589  
ayoaron : :reseau : :CNetworkMachine, 590  
~CNetworkMachine, 592  
address, 595  
attachPlayer, 592  
CNetworkMachine, 591, 592  
connect, 592  
getAddress, 593  
getId, 593  
hasNick, 593  
id, 596  
io, 596  
l\_players, 596  
network\_buffer, 596  
parent, 596  
removePlayer, 593, 594  
s\_tcp, 596  
s\_udp, 596  
tcpAsyncReceive, 594  
tcpSend, 595  
tcpThreadSend, 595  
udp\_ep, 596  
udpSend, 595  
ayoaron : :reseau : :CNetworkPlayer, 597  
CNetworkPlayer, 598  
getName, 598  
parent, 599  
s\_tcp, 599  
setParent, 599  
background  
ayoaron : :graphic : :CGauge, 206  
ayoaron : :graphic : :CGaugeMoteur,  
213  
ayoaron : :graphic : :CGaugeVitesse,  
223  
ayoaron : :graphic : :CVaisseauPanel,  
281  
bits  
ayoaron : :graphic : :CGraphicsOptions,  
249  
blackbox  
ayoaron : :objets : :CVaisseau, 406  
ayoaron : :objets : :CVaisseauX1, 449  
BOOL  
ayoaron : :data, 15  
bot\_names  
ayoaron : :core : :CGameEngine, 96  
bougeObjet

ayoaron :: physic :: CObjet, 465  
broadcastGameProposition  
    ayoaron :: reseau :: CNetworkEngine, 559  
BUFFER\_SIZE  
    CNetworkEngine.h, 675  
buildInternetInterface  
    ayoaron :: reseau :: CNetworkEngine, 560  
buildLANInterface  
    ayoaron :: reseau :: CNetworkEngine, 560  
cadre1  
    ayoaron :: graphic :: datamot, 282  
cadre2  
    ayoaron :: graphic :: datamot, 282  
CAiPlayer  
    ayoaron :: core :: CAiPlayer, 51  
CAiPlayer.cpp, 600  
CAiPlayer.h, 602  
calcConsigne  
    ayoaron :: objets :: CModulePilotage, 322  
cam  
    ayoaron :: objets :: CAsteroide, 314  
    ayoaron :: objets :: CMoteur, 362  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 449  
    ayoaron :: physic :: CObjet, 475  
    ayoaron :: physic :: CSolideRigide, 533  
caml  
    ayoaron :: objets :: CAsteroide, 315  
    ayoaron :: objets :: CMoteur, 362  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 449  
    ayoaron :: physic :: CObjet, 475  
    ayoaron :: physic :: CSolideRigide, 533  
camp  
    ayoaron :: objets :: CAsteroide, 315  
    ayoaron :: objets :: CMoteur, 362  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 449  
    ayoaron :: physic :: CObjet, 475  
    ayoaron :: physic :: CSolideRigide, 533  
camup  
    ayoaron :: objets :: CAsteroide, 315  
    ayoaron :: objets :: CMoteur, 362  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 449  
    ayoaron :: physic :: CObjet, 475  
    ayoaron :: physic :: CSolideRigide, 533  
CAsteroide  
    ayoaron :: objets :: CAsteroide, 286  
CAsteroide.cpp, 603  
CAsteroide.h, 604  
CAvailableAi.h, 605  
CAvailableMap.h, 607  
CBoiteNoire  
    ayoaron :: objets :: CBoiteNoire, 319  
CBoiteNoire.cpp, 608  
CBoiteNoire.h, 609  
CConfigFile  
    ayoaron :: data :: CConfigFile, 128  
CConfigFile.cpp, 611  
CConfigFile.h, 612  
CConsole  
    ayoaron :: core :: CConsole, 55  
CConsole.cpp, 613  
CConsole.h, 614  
CData  
    ayoaron :: data :: CData, 132  
CData.cpp, 615  
CData.h, 616  
CDataNode  
    ayoaron :: data :: CDataNode, 149  
CDataNode.cpp, 618  
CDataNode.h, 619  
CEffort  
    ayoaron :: physic :: CEffort, 454  
CEffort.cpp, 621  
CEffort.h, 622  
CEngine  
    ayoaron :: core :: CEngine, 62  
CEngine.cpp, 623  
CEngine.h, 624  
CEngineEvent.h, 625  
CEventHandler  
    ayoaron :: event :: CEventHandler, 189  
CEventHandler.h, 627  
cfile  
    ayoaron :: audio :: CSoundEngine, 45  
    ayoaron :: core :: CEngine, 72  
    ayoaron :: core :: CGameEngine, 96  
    ayoaron :: graphic :: CGraphicsEngine, 244  
    ayoaron :: physic :: CPhysicEngine, 490  
    ayoaron :: reseau :: CNetworkEngine, 580  
    CFileManager

ayoaron : :data : :CFileManager, 177  
CFileManager.cpp, 628  
CFileManager.h, 629  
CGalaxie  
    ayoaron : :physic : :CGalaxie, 458  
CGalaxie.cpp, 631  
CGalaxie.h, 632  
CGame  
    ayoaron : :core : :CGame, 75  
CGame.cpp, 633  
CGame.h, 634  
CGameEngine  
    ayoaron : :core : :CGameEngine, 83  
CGameEngine.cpp, 636  
CGameEngine.h, 637  
CGameEventHandler  
    ayoaron : :event : :CGameEventHandler, 194  
CGameEventHandler.cpp, 638  
CGameEventHandler.h, 639  
CGameRound  
    ayoaron : :core : :CGameRound, 101  
CGameRound.cpp, 640  
CGameRound.h, 642  
CGauge  
    ayoaron : :graphic : :CGauge, 203  
CGauge.cpp, 643  
CGauge.h, 644  
CGaugeMoteur  
    ayoaron : :graphic : :CGaugeMoteur, 209  
CGaugeMoteur.cpp, 646  
CGaugeMoteur.h, 647  
CGaugeVitesse  
    ayoaron : :graphic : :CGaugeVitesse, 216  
CGaugeVitesse.cpp, 648  
CGaugeVitesse.h, 650  
CGraphicsEngine  
    ayoaron : :graphic : :CGraphicsEngine, 229  
CGraphicsEngine.cpp, 651  
CGraphicsEngine.h, 652  
CGraphicsOptions  
    ayoaron : :graphic : :CGraphicsOptions, 249  
CGraphicsOptions.h, 654  
CGuiSystem  
    ayoaron : :graphic : :CGuiSystem, 252  
CGuiSystem.cpp, 655  
    CGuiSystem.h, 656  
    CHOSEN\_MAP  
        ayoaron : :event : :CEngineEvent, 187  
    CIInternetNetworkInterface  
        ayoaron : :reseau : :CIInternetNetworkInterface, 542  
    CIInternetNetworkInterface.cpp, 658  
    CIInternetNetworkInterface.h, 659  
    CLANNetworkInterface  
        ayoaron : :reseau : :CLANNetworkInterface, 548  
    CLANNetworkInterface.cpp, 660  
    CLANNetworkInterface.h, 662  
    cleanInterface  
        ayoaron : :reseau : :CNetworkEngine, 560  
    clear  
        ayoaron : :core : :CRoundOptions, 118  
    clientConnectsToServer  
        ayoaron : :reseau : :CNetworkEngine, 561  
    CLogger  
        ayoaron : :core : :CLogger, 109  
    CLogger.cpp, 663  
    CLogger.h, 664  
    close  
        ayoaron : :graphic : :CGuiSystem, 253  
    CMenuEventHandler  
        ayoaron : :event : :CMenuEventHandler, 198  
    CMenuEventHandler.cpp, 666  
    CMenuEventHandler.h, 666  
    CModulePilotage  
        ayoaron : :objets : :CModulePilotage, 322  
    CModulePilotage.cpp, 668  
    CModulePilotage.h, 668  
    CMoteur  
        ayoaron : :objets : :CMoteur, 329  
    CMoteur.cpp, 670  
    CMoteur.h, 670  
    CNetworkEngine  
        ayoaron : :reseau : :CNetworkEngine, 556  
        CNetworkEngine.cpp, 672  
        CNetworkEngine.h, 673  
            BUFFER\_SIZE, 675  
    CNetworkInterface  
        ayoaron : :reseau : :CNetworkInterface, 586

CNetworkInterface.cpp, 675  
CNetworkInterface.h, 676  
CNetworkMachine  
    ayoaron :: reseau :: CNetworkMachine, 591, 592  
CNetworkMachine.cpp, 677  
CNetworkMachine.h, 678  
CNetworkPlayer  
    ayoaron :: reseau :: CNetworkPlayer, 598  
CNetworkPlayer.cpp, 680  
CNetworkPlayer.h, 682  
cntmot  
    ayoaron :: objets :: CVaisseauX1, 449  
cntvit  
    ayoaron :: objets :: CVaisseauX1, 450  
COBJet  
    ayoaron :: physic :: COBJet, 465  
COBJet.cpp, 683  
COBJet.h, 684  
collectAvailableAis  
    ayoaron :: core :: CGameEngine, 86  
collectAvailableMaps  
    ayoaron :: core :: CGameEngine, 86  
COLOR  
    ayoaron :: data, 15  
compteurs  
    ayoaron :: graphic :: CVaisseauPanel, 281  
config  
    ayoaron :: data :: CFileManager, 183  
connect  
    ayoaron :: reseau :: CNetworkMachine, 592  
cons  
    ayoaron :: core :: CGameEngine, 96  
consigne  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 450  
consignerotvitesse  
    ayoaron :: objets :: CVaisseau, 406  
    ayoaron :: objets :: CVaisseauX1, 450  
consignevit  
    ayoaron :: objets :: CModulePilotage, 324  
consignevitesse  
    ayoaron :: objets :: CVaisseau, 407  
    ayoaron :: objets :: CVaisseauX1, 450  
consignevitrot  
    ayoaron :: core :: CSharedInstance, 125  
    ayoaron :: objets :: CModulePilotage, 324  
    console\_visible  
    ayoaron :: graphic :: CGuiSystem, 267  
CParameterAnalyser  
    ayoaron :: core :: CParameterAnalyser, 112  
CParameterAnalyser.cpp, 685  
CParameterAnalyser.h, 686  
CPhysicEngine  
    ayoaron :: physic :: CPhysicEngine, 480  
CPhysicEngine.cpp, 688  
CPhysicEngine.h, 689  
CPlayer  
    ayoaron :: core :: CPlayer, 114  
CPlayer.cpp, 690  
CPlayer.h, 691  
CPlayerOptions.h, 692  
CPousMouv  
    ayoaron :: objets :: CPousMouv, 366  
creaobjet  
    ayoaron :: objets, 18  
creaobjet.cpp, 693  
creaobjet.h, 694  
createEnfant  
    ayoaron :: data :: CDataNode, 150  
CRoundOptions  
    ayoaron :: core :: CRoundOptions, 118  
CRoundOptions.h, 696  
CSceneGraph  
    ayoaron :: graphic :: CSceneGraph, 269  
CSceneGraph.cpp, 697  
CSceneGraph.h, 698  
CSecteur  
    ayoaron :: physic :: CSecteur, 494  
CSecteur.cpp, 699  
CSecteur.h, 700  
CSerializableVector3df  
    ayoaron :: event :: CSerializableVector3df, 201  
CShared  
    ayoaron :: core :: CShared, 121  
CShared.cpp, 702  
CShared.h, 703  
CSharedInstance  
    ayoaron :: core :: CSharedInstance, 125  
CSolideRigide

ayoaron ::physic ::CSolideRigide, 505  
CSolideRigide.cpp, 704  
CSolideRigide.h, 705  
CSound  
    ayoaron ::audio ::CSound, 22  
CSound.cpp, 706  
CSound.h, 707  
CSoundEngine  
    ayoaron ::audio ::CSoundEngine, 31  
CSoundEngine.cpp, 708  
CSoundEngine.h, 709  
CSoundOptions  
    ayoaron ::audio ::CSoundOptions, 48  
CSoundOptions.h, 710  
CVaisseau  
    ayoaron ::objets ::CVaisseau, 372  
CVaisseau.cpp, 711  
CVaisseau.h, 712  
CVaisseauPanel  
    ayoaron ::graphic ::CVaisseauPanel, 275  
CVaisseauPanel.cpp, 713  
CVaisseauPanel.h, 714  
CVaisseauX1  
    ayoaron ::objets ::CVaisseauX1, 414  
CVaisseauX1.cpp, 715  
CVaisseauX1.h, 716  
CWorld  
    ayoaron ::physic ::CWorld, 536  
CWorld.cpp, 718  
CWorld.h, 718  
CXMLExtractor  
    ayoaron ::core ::CXMLExtractor, 126  
CXMLExtractor.cpp, 720  
CXMLExtractor.h, 720  
data  
    ayoaron ::core ::CParameterAnalyser, 112  
    ayoaron ::core ::CShared, 123  
    ayoaron ::data ::CConfigFile, 130  
    ayoaron ::data ::CData, 146  
DATATYPE  
    ayoaron ::data, 14  
debuglvl  
    ayoaron ::graphic ::CGraphicsOptions, 249  
defineRegimeFM  
    ayoaron ::objets ::CVaisseau, 375  
    ayoaron ::objets ::CVaisseauX1, 417  
DEL\_PLAYER  
    ayoaron ::event ::CEngineEvent, 187  
DeltaTemps  
    ayoaron ::core ::CGame, 78  
deltavit  
    ayoaron ::objets ::CModulePilotage, 324  
deltavitrot  
    ayoaron ::objets ::CModulePilotage, 324  
description  
    ayoaron ::core ::CAvailableMap, 53  
destrSound  
    ayoaron ::audio ::CSound, 22  
device  
    ayoaron ::audio ::CSoundEngine, 45  
    ayoaron ::graphic ::CVaisseauPanel, 281  
displayAide  
    ayoaron ::graphic ::CGuiSystem, 253  
displayGameGui  
    ayoaron ::graphic ::CGuiSystem, 254  
displayIngameMenu  
    ayoaron ::graphic ::CGuiSystem, 254  
displayLoadgame  
    ayoaron ::graphic ::CGuiSystem, 255  
displayLoading  
    ayoaron ::graphic ::CGuiSystem, 255  
displayMenuPrincipal  
    ayoaron ::graphic ::CGuiSystem, 256  
displayOptions  
    ayoaron ::graphic ::CGuiSystem, 257  
displaySelectPlayers  
    ayoaron ::graphic ::CGuiSystem, 257  
distfac  
    ayoaron ::audio ::CSoundOptions, 48  
donn  
    ayoaron ::core ::CGameRound, 106  
    ayoaron ::data ::CDataNode, 172  
    ayoaron ::objets ::CAsteroide, 315  
    ayoaron ::objets ::CMoteur, 362  
    ayoaron ::objets ::CVaisseau, 407  
    ayoaron ::objets ::CVaisseauX1, 450  
    ayoaron ::physic ::CGalaxie, 462  
    ayoaron ::physic ::CObjet, 475  
    ayoaron ::physic ::CSecteur, 501  
    ayoaron ::physic ::CSolideRigide, 533  
    ayoaron ::physic ::CWorld, 539

doplfac  
    ayoaron ::audio ::CSoundOptions, 49

draw  
    ayoaron ::graphic ::CGauge, 204  
    ayoaron ::graphic ::CGaugeMoteur, 209  
    ayoaron ::graphic ::CGaugeVitesse, 217

draw2DImage  
    ayoaron ::graphic ::CVaisseauPanel, 277

drawAll  
    ayoaron ::graphic ::CVaisseauPanel, 278

drawBackground  
    ayoaron ::graphic ::CVaisseauPanel, 278

driver  
    ayoaron ::graphic ::CGraphicsEngine, 244

drop  
    ayoaron ::data ::CData, 133  
    ayoaron ::data ::CDataNode, 150

dropAllSound  
    ayoaron ::audio ::CSoundEngine, 34

dropEnfant  
    ayoaron ::data ::CDataNode, 151  
    ayoaron ::objets ::CAsteriode, 289  
    ayoaron ::objets ::CMoteur, 332  
    ayoaron ::objets ::CVaisseau, 376  
    ayoaron ::objets ::CVaisseauX1, 418  
    ayoaron ::physic ::CSolideRigide, 509

dropSound  
    ayoaron ::audio ::CSoundEngine, 35

drv  
    ayoaron ::graphic ::CGraphicsOptions, 249

dt1  
    ayoaron ::objets ::CModulePilotage, 324

DTV  
    ayoaron ::objets ::CModulePilotage, 324  
    ayoaron ::objets ::CVaisseau, 407  
    ayoaron ::objets ::CVaisseauX1, 451

DTV  
    ayoaron ::objets ::CModulePilotage, 324  
    ayoaron ::objets ::CVaisseau, 407

ayoaron ::objets ::CVaisseauX1, 451

ayoaron ::objets ::CVaisseauX1, 451

EffortResultant  
    ayoaron ::physic ::CObjet, 476

engine\_name  
    ayoaron ::audio ::CSoundEngine, 45  
    ayoaron ::core ::CEngine, 72  
    ayoaron ::core ::CGameEngine, 96  
    ayoaron ::graphic ::CGraphicsEngine, 245  
    ayoaron ::physic ::CPhysicEngine, 490  
    ayoaron ::reseau ::CNetworkEngine, 580

events\_queue  
    ayoaron ::audio ::CSoundEngine, 45  
    ayoaron ::core ::CEngine, 72  
    ayoaron ::core ::CGameEngine, 96  
    ayoaron ::graphic ::CGraphicsEngine, 245  
    ayoaron ::physic ::CPhysicEngine, 491  
    ayoaron ::reseau ::CNetworkEngine, 580

f  
    ayoaron ::core ::CLogger, 110  
    ayoaron ::objets ::CBoiteNoire, 320

f\_data  
    ayoaron ::core ::CPlayerOptions, 116  
    ayoaron ::data ::SConfigFileData, 184

filename  
    ayoaron ::objets ::CBoiteNoire, 320

filesys  
    ayoaron ::data ::CFileManager, 183

FLOAT  
    ayoaron ::data, 15

flux  
    ayoaron ::objets ::CMoteur, 363

flux2  
    ayoaron ::objets ::CMoteur, 363

force  
    ayoaron ::objets ::CMoteur, 363  
    ayoaron ::physic ::CEffort, 456

forceResultante  
    ayoaron ::objets ::CAsteriode, 315  
    ayoaron ::objets ::CMoteur, 363  
    ayoaron ::objets ::CVaisseau, 407  
    ayoaron ::objets ::CVaisseauX1, 451  
    ayoaron ::physic ::CSolideRigide, 534

frame  
    ayoaron ::audio ::CSoundEngine, 35  
    ayoaron ::core ::CEngine, 64  
    ayoaron ::core ::CGameEngine, 86  
    ayoaron ::core ::CGameRound, 102  
    ayoaron ::graphic ::CGraphicsEngine, 232  
    ayoaron ::objets ::CAsteroide, 289  
    ayoaron ::objets ::CBoiteNoire, 319  
    ayoaron ::objets ::CMoteur, 332  
    ayoaron ::objets ::CVaisseau, 376  
    ayoaron ::objets ::CVaisseauX1, 418  
    ayoaron ::physic ::CGalaxie, 458  
    ayoaron ::physic ::COBJet, 466  
    ayoaron ::physic ::CPhysicEngine, 483  
    ayoaron ::physic ::CSecteur, 495  
    ayoaron ::physic ::CSolideRigide, 509  
    ayoaron ::physic ::CWorld, 537  
    ayoaron ::reseau ::CNetworkEngine, 561  
frametc  
    ayoaron ::core ::CGameEngine, 97  
fullscrn  
    ayoaron ::graphic ::CGraphicsOptions, 249  
g  
    ayoaron ::core ::CGame, 78  
galax  
    ayoaron ::core ::CGameRound, 106  
GAME  
    ayoaron ::graphic ::CGuiSystem, 252  
GAME\_HAS\_STARTED  
    ayoaron ::event ::CEngineEvent, 186  
GAME\_PROP  
    ayoaron ::event ::CEngineEvent, 186  
game\_round, 599  
gaugepanel  
    ayoaron ::graphic ::CGuiSystem, 267  
ge  
    ayoaron ::audio ::CSoundEngine, 45  
    ayoaron ::core ::CEngine, 72  
    ayoaron ::core ::CGameEngine, 97  
    ayoaron ::graphic ::CGraphicsEngine, 245  
    ayoaron ::physic ::CPhysicEngine, 491  
    ayoaron ::reseau ::CNetworkEngine, 580  
GENERAL  
    ayoaron ::physic, 19  
get  
    ayoaron ::core ::CShared, 121  
getAbsoluteForceResultante  
    ayoaron ::objets ::CAsteroide, 290  
    ayoaron ::objets ::CMoteur, 333  
    ayoaron ::objets ::CVaisseau, 377  
    ayoaron ::objets ::CVaisseauX1, 419  
    ayoaron ::physic ::CSolideRigide, 510  
getAbsoluteMomentResultante  
    ayoaron ::objets ::CAsteroide, 291  
    ayoaron ::objets ::CMoteur, 334  
    ayoaron ::objets ::CVaisseau, 378  
    ayoaron ::objets ::CVaisseauX1, 419  
    ayoaron ::physic ::CSolideRigide, 510  
getAbsolutePosition  
    ayoaron ::objets ::CAsteroide, 291  
    ayoaron ::objets ::CMoteur, 334  
    ayoaron ::objets ::CVaisseau, 378  
    ayoaron ::objets ::CVaisseauX1, 420  
    ayoaron ::physic ::CSolideRigide, 511  
getAbsoluteRitesse  
    ayoaron ::objets ::CAsteroide, 292  
    ayoaron ::objets ::CMoteur, 335  
    ayoaron ::objets ::CVaisseau, 379  
    ayoaron ::objets ::CVaisseauX1, 420  
    ayoaron ::physic ::CSolideRigide, 511  
getAbsoluteRotation  
    ayoaron ::objets ::CAsteroide, 292  
    ayoaron ::objets ::CMoteur, 335  
    ayoaron ::objets ::CVaisseau, 379  
    ayoaron ::objets ::CVaisseauX1, 421  
    ayoaron ::physic ::CSolideRigide, 512  
getAbsoluteRotVelocity  
    ayoaron ::physic ::COBJet, 467  
getAbsoluteVelocity  
    ayoaron ::physic ::COBJet, 468  
getAbsoluteVitesse  
    ayoaron ::objets ::CAsteroide, 293  
    ayoaron ::objets ::CMoteur, 336  
    ayoaron ::objets ::CVaisseau, 380  
    ayoaron ::objets ::CVaisseauX1, 422  
    ayoaron ::physic ::CSolideRigide, 512  
getAddress  
    ayoaron ::reseau ::CNetworkMachine, 593  
getAi  
    ayoaron ::core ::CXMLExtractor, 126  
getbool

ayoaron ::data ::CData, 133  
ayoaron ::data ::CDataNode, 151  
getCam  
    ayoaron ::physic ::CObjet, 468  
getcol  
    ayoaron ::data ::CData, 134  
    ayoaron ::data ::CDataNode, 152  
getConfig  
    ayoaron ::data ::CFileManager, 177  
getConfigData  
    ayoaron ::audio ::CSoundEngine, 36  
    ayoaron ::core ::CEngine, 64  
    ayoaron ::core ::CGameEngine, 87  
    ayoaron ::graphic ::CGraphicsEngine, 233  
    ayoaron ::physic ::CPhysicsEngine, 483  
    ayoaron ::reseau ::CNetworkEngine, 562  
getConsole  
    ayoaron ::core ::CGameEngine, 87  
getData  
    ayoaron ::data ::CConfigFile, 128  
    ayoaron ::data ::CDataNode, 153  
getDataList  
    ayoaron ::data ::CData, 134  
    ayoaron ::data ::CDataNode, 154  
getDataListCompared  
    ayoaron ::data ::CDataNode, 155  
getDataNode  
    ayoaron ::physic ::CSecteur, 496  
getDataType  
    ayoaron ::data ::CData, 135  
    ayoaron ::data ::CDataNode, 156  
getDataTypeWstr  
    ayoaron ::data ::CFileManager, 177  
getDbgLevel  
    ayoaron ::graphic ::CGraphicsEngine, 234  
getDevice  
    ayoaron ::audio ::CSoundEngine, 37  
    ayoaron ::graphic ::CGraphicsEngine, 234  
getdInertie  
    ayoaron ::physic ::CObjet, 468  
getDirection  
    ayoaron ::physic ::CEffort, 454  
getdMasse  
    ayoaron ::physic ::CObjet, 468  
getDonn

    ayoaron ::objets ::CAsteroide, 294  
    ayoaron ::objets ::CMoteur, 337  
    ayoaron ::objets ::CVaisseau, 381  
    ayoaron ::objets ::CVaisseauX1, 422  
    ayoaron ::physic ::CSolideRigide, 513  
getDonnee  
    ayoaron ::physic ::CGalaxie, 459  
getDriver  
    ayoaron ::graphic ::CGraphicsEngine, 235  
getDt  
    ayoaron ::core ::CGame, 76  
getEnfant  
    ayoaron ::data ::CDataNode, 157  
    ayoaron ::objets ::CAsteroide, 294  
    ayoaron ::objets ::CMoteur, 337  
    ayoaron ::objets ::CVaisseau, 381  
    ayoaron ::objets ::CVaisseauX1, 423  
    ayoaron ::physic ::CSolideRigide, 513  
getf64  
    ayoaron ::data ::CData, 136  
    ayoaron ::data ::CDataNode, 157  
getForce  
    ayoaron ::objets ::CMoteur, 337  
    ayoaron ::physic ::CEffort, 454  
getGameEngine  
    ayoaron ::audio ::CSoundEngine, 37  
    ayoaron ::core ::CEngine, 65  
    ayoaron ::core ::CGameEngine, 87  
    ayoaron ::graphic ::CGraphicsEngine, 235  
    ayoaron ::physic ::CPhysicsEngine, 483  
    ayoaron ::reseau ::CNetworkEngine, 562  
getGauge  
    ayoaron ::graphic ::CVaisseauPanel, 279  
getGfxEngine  
    ayoaron ::physic ::CSecteur, 496  
getGlobalEffort  
    ayoaron ::physic ::CSecteur, 497  
    ayoaron ::physic ::CWorld, 538  
getGlobalInertie  
    ayoaron ::objets ::CAsteroide, 294  
    ayoaron ::objets ::CMoteur, 338  
    ayoaron ::objets ::CVaisseau, 381  
    ayoaron ::objets ::CVaisseauX1, 423  
    ayoaron ::physic ::CSolideRigide, 514  
getGlobalScale

ayoaron ::graphic ::CVaisseauPanel, getLocalMomentResultante  
279  
ayoaron ::objets ::CAsteroid, 296  
getGraphicsEngine  
ayoaron ::audio ::CSoundEngine, 38  
ayoaron ::core ::CEngine, 65  
ayoaron ::core ::CGameEngine, 88  
ayoaron ::graphic ::CGraphicsEngine, getLocalRitesse  
236  
ayoaron ::physic ::CGalaxie, 459  
ayoaron ::physic ::CPhysicEngine,  
484  
ayoaron ::reseau ::CNetworkEngine,  
563  
getGui  
ayoaron ::graphic ::CGraphicsEngine, getLocalVelocity  
236  
ayoaron ::objets ::CObjet, 469  
getId  
ayoaron ::reseau ::CNetworkMachine,  
593  
getInertie  
ayoaron ::physic ::CObjet, 468  
getIrrRenderer  
ayoaron ::graphic ::CGraphicsEngine, getMachineFromAddress  
237  
ayoaron ::reseau ::CNetworkEngine,  
563  
getIteratorEnfantBegin  
ayoaron ::data ::CDataNode, 158  
ayoaron ::objets ::CAsteroid, 295  
ayoaron ::objets ::CMoteur, 339  
ayoaron ::objets ::CVaisseau, 382  
ayoaron ::objets ::CVaisseauX1, 426  
ayoaron ::physic ::CSolideRigide, 514  
getIteratorEnfantEnd  
ayoaron ::data ::CDataNode, 159  
ayoaron ::objets ::CAsteroid, 295  
ayoaron ::objets ::CMoteur, 339  
ayoaron ::objets ::CVaisseau, 382  
ayoaron ::objets ::CVaisseauX1, 424  
ayoaron ::physic ::CSolideRigide, 514  
getLocalForceResultante  
ayoaron ::objets ::CAsteroid, 295  
ayoaron ::objets ::CMoteur, 339  
ayoaron ::objets ::CVaisseau, 382  
ayoaron ::objets ::CVaisseauX1, 424  
ayoaron ::physic ::CSolideRigide, 514  
getLocallInertie  
ayoaron ::objets ::CAsteroid, 295  
ayoaron ::objets ::CMoteur, 339  
ayoaron ::objets ::CVaisseau, 382  
ayoaron ::objets ::CVaisseauX1, 424  
ayoaron ::physic ::CSolideRigide, 515  
getLocalRotVelocity  
ayoaron ::physic ::CObjet, 469  
getLocalVelocity  
ayoaron ::objets ::CObjet, 469  
getLocalVitesse  
ayoaron ::objets ::CAsteroid, 297  
ayoaron ::objets ::CMoteur, 341  
ayoaron ::objets ::CVaisseau, 384  
ayoaron ::objets ::CVaisseauX1, 426  
ayoaron ::physic ::CSolideRigide, 516  
getMachineFromAddress  
ayoaron ::reseau ::CNetworkEngine,  
563  
getMachineFromId  
ayoaron ::reseau ::CNetworkEngine,  
564  
getMachineFromPlayerName  
ayoaron ::reseau ::CNetworkEngine,  
564  
getMap  
ayoaron ::core ::CXMLExtractor, 126  
getMapList  
ayoaron ::graphic ::CGuiSystem, 257  
getMapPath  
ayoaron ::core ::CGameEngine, 88  
getMasse  
ayoaron ::objets ::CAsteroid, 298  
ayoaron ::objets ::CMoteur, 342  
ayoaron ::objets ::CVaisseau, 385  
ayoaron ::objets ::CVaisseauX1, 427  
ayoaron ::physic ::CObjet, 469  
ayoaron ::physic ::CSolideRigide, 517  
getmat4  
ayoaron ::data ::CData, 137  
ayoaron ::data ::CDataNode, 159  
getMatrix3  
ayoaron ::objets ::CAsteroid, 299  
ayoaron ::objets ::CMoteur, 342

ayoaron ::objets ::CVaisseau, 386  
ayoaron ::objets ::CVaisseauX1, 427  
ayoaron ::physic ::CSolideRigide, 517  
getMesh  
    ayoaron ::physic ::CObjet, 470  
getMoment  
    ayoaron ::physic ::CEffort, 455  
getName  
    ayoaron ::core ::CAiPlayer, 51  
    ayoaron ::core ::CPlayer, 114  
    ayoaron ::reseau ::CNetworkPlayer,  
        598  
getNbEnfant  
    ayoaron ::data ::CDataNode, 160  
    ayoaron ::objets ::CAsteroid, 300  
    ayoaron ::objets ::CMoteur, 343  
    ayoaron ::objets ::CVaisseau, 387  
    ayoaron ::objets ::CVaisseauX1, 428  
    ayoaron ::physic ::CSolideRigide, 518  
getNetworkEngine  
    ayoaron ::audio ::CSoundEngine, 38  
    ayoaron ::core ::CEngine, 66  
    ayoaron ::core ::CGameEngine, 89  
    ayoaron ::graphic ::CGraphicsEngine,  
        237  
    ayoaron ::physic ::CPhysicEngine,  
        484  
    ayoaron ::reseau ::CNetworkEngine,  
        565  
getNextMachinelD  
    ayoaron ::reseau ::CNetworkEngine,  
        565  
getNiveau  
    ayoaron ::data ::CDataNode, 161  
getNode  
    ayoaron ::objets ::CAsteroid, 300  
    ayoaron ::objets ::CMoteur, 343  
    ayoaron ::objets ::CVaisseau, 387  
    ayoaron ::objets ::CVaisseauX1, 428  
    ayoaron ::physic ::CObjet, 470  
    ayoaron ::physic ::CSolideRigide, 519  
getNom  
    ayoaron ::physic ::CGalaxie, 459  
    ayoaron ::physic ::CSecteur, 497  
getNorme  
    ayoaron ::physic ::CEffort, 455  
getNSecteur  
    ayoaron ::physic ::CGalaxie, 460  
getObjetListCompared  
    ayoaron ::data ::CDataNode, 161  
getObjetParent  
    ayoaron ::physic ::CObjet, 470  
getPanel  
    ayoaron ::graphic ::CGuiSystem, 258  
getParent  
    ayoaron ::audio ::CSoundEngine, 39  
    ayoaron ::core ::CEngine, 66  
    ayoaron ::core ::CGameEngine, 89  
    ayoaron ::data ::CDataNode, 162  
    ayoaron ::event ::CEventHandler, 190  
    ayoaron ::event ::CGameEventHand-  
        ler, 194  
    ayoaron ::event ::CMenuEventHand-  
        ler, 198  
    ayoaron ::graphic ::CGraphicsEngine,  
        237  
    ayoaron ::physic ::CPhysicEngine,  
        485  
    ayoaron ::physic ::CSecteur, 498  
    ayoaron ::reseau ::CNetworkEngine,  
        565  
getPitch  
    ayoaron ::audio ::CSound, 22  
getPos  
    ayoaron ::audio ::CSound, 22  
getPousse  
    ayoaron ::objets ::CMoteur, 344  
getPousseMax  
    ayoaron ::objets ::CMoteur, 344  
getPP  
    ayoaron ::objets ::CMoteur, 345  
getPP2  
    ayoaron ::objets ::CMoteur, 345  
getPtApplic  
    ayoaron ::physic ::CEffort, 455  
getRegime  
    ayoaron ::objets ::CMoteur, 345  
getRelativeForceResultante  
    ayoaron ::objets ::CAsteroid, 300  
    ayoaron ::objets ::CMoteur, 346  
    ayoaron ::objets ::CVaisseau, 387  
    ayoaron ::objets ::CVaisseauX1, 429  
    ayoaron ::physic ::CSolideRigide, 519  
getRelativeInertie  
    ayoaron ::objets ::CAsteroid, 301  
    ayoaron ::objets ::CMoteur, 347  
    ayoaron ::objets ::CVaisseau, 388  
    ayoaron ::objets ::CVaisseauX1, 430  
    ayoaron ::physic ::CSolideRigide, 520  
getRelativeMomentResultante

ayoaron ::objets ::CAsteroid, 302  
ayoaron ::objets ::CMoteur, 347  
ayoaron ::objets ::CVaisseau, 389  
ayoaron ::objets ::CVaisseauX1, 430  
ayoaron ::physic ::CSolideRigide, 520  
getRelativePosition  
    ayoaron ::objets ::CAsteroid, 302  
    ayoaron ::objets ::CMoteur, 348  
    ayoaron ::objets ::CVaisseau, 389  
    ayoaron ::objets ::CVaisseauX1, 431  
    ayoaron ::physic ::CSolideRigide, 520  
getRelativeRitesse  
    ayoaron ::objets ::CAsteroid, 303  
    ayoaron ::objets ::CMoteur, 348  
    ayoaron ::objets ::CVaisseau, 390  
    ayoaron ::objets ::CVaisseauX1, 431  
    ayoaron ::physic ::CSolideRigide, 521  
getRelativeRotation  
    ayoaron ::objets ::CAsteroid, 303  
    ayoaron ::objets ::CMoteur, 349  
    ayoaron ::objets ::CVaisseau, 390  
    ayoaron ::objets ::CVaisseauX1, 432  
    ayoaron ::physic ::CSolideRigide, 522  
getRelativeRotVelocity  
    ayoaron ::physic ::CObjet, 471  
getRelativeVelocity  
    ayoaron ::physic ::CObjet, 471  
getRelativeVitesse  
    ayoaron ::objets ::CAsteroid, 304  
    ayoaron ::objets ::CMoteur, 349  
    ayoaron ::objets ::CVaisseau, 391  
    ayoaron ::objets ::CVaisseauX1, 432  
    ayoaron ::physic ::CSolideRigide, 522  
getRoot  
    ayoaron ::data ::CDataNode, 163  
gets32  
    ayoaron ::data ::CData, 137  
    ayoaron ::data ::CDataNode, 163  
getSceneGraph  
    ayoaron ::graphic ::CGraphicsEngine, 238  
getSceneManager  
    ayoaron ::physic ::CObjet, 471  
    ayoaron ::physic ::CSecteur, 498  
    ayoaron ::physic ::CWorld, 538  
getSecteur  
    ayoaron ::physic ::CGalaxie, 460  
getSecteurCourant  
    ayoaron ::physic ::CGalaxie, 460  
getSecteurParent

    ayoaron ::physic ::CObjet, 471  
    ayoaron ::physic ::CSecteur, 498  
    ayoaron ::physic ::CWorld, 538  
    ayoaron ::graphic ::CGraphicsEngine, 238  
    ayoaron ::audio ::CSound, 22  
    ayoaron ::reseau ::CIternetNetworkInterface, 542  
    ayoaron ::reseau ::CLANNetworkInterface, 548  
    ayoaron ::reseau ::CNetworkInterface, 586  
    ayoaron ::data ::CData, 139  
    ayoaron ::data ::CDataNode, 165  
getVec3  
    ayoaron ::data ::CData, 140  
    ayoaron ::data ::CDataNode, 166  
getVector  
    ayoaron ::objets ::CAsteroid, 305  
    ayoaron ::objets ::CMoteur, 350  
    ayoaron ::objets ::CVaisseau, 392  
    ayoaron ::objets ::CVaisseauX1, 433  
    ayoaron ::physic ::CSolideRigide, 523  
getVideoDriver  
    ayoaron ::graphic ::CVaisseauPanel, 279  
getVit  
    ayoaron ::audio ::CSound, 23  
getVolume

ayoaron ::audio ::CSound, 23  
gfx  
    ayoaron ::core ::CGame, 78  
    ayoaron ::physic ::CGalaxie, 462  
gfxe  
    ayoaron ::audio ::CSoundEngine, 46  
    ayoaron ::core ::CEngine, 72  
    ayoaron ::core ::CGameEngine, 97  
    ayoaron ::graphic ::CGraphicsEngine, 245  
    ayoaron ::physic ::CPhysicEngine, 491  
    ayoaron ::reseau ::CNetworkEngine, 580  
gr  
    ayoaron ::core ::CGameEngine, 97  
gravityAffective  
    ayoaron ::physic ::CObjet, 476  
gui  
    ayoaron ::graphic ::CGraphicsEngine, 245  
h  
    ayoaron ::objets ::CPousMouv, 366  
handleAcceptTCP  
    ayoaron ::core ::CConsole, 55  
    ayoaron ::reseau ::CNetworkEngine, 566  
handleClose  
    ayoaron ::graphic ::CGuiSystem, 258  
handleDisplayAideIG  
    ayoaron ::graphic ::CGuiSystem, 258  
handleDisplayAideMP  
    ayoaron ::graphic ::CGuiSystem, 259  
handleDisplayLoadgame  
    ayoaron ::graphic ::CGuiSystem, 259  
handleDisplayLoading  
    ayoaron ::graphic ::CGuiSystem, 260  
handleDisplayOptionsIG  
    ayoaron ::graphic ::CGuiSystem, 260  
handleDisplayOptionsMP  
    ayoaron ::graphic ::CGuiSystem, 261  
handleProcessCommand  
    ayoaron ::graphic ::CGuiSystem, 261  
handleRetourToIngame  
    ayoaron ::graphic ::CGuiSystem, 262  
handleRetourToMain  
    ayoaron ::graphic ::CGuiSystem, 262  
hascamera  
    ayoaron ::objets ::CAsteroide, 316  
ayoaron ::objets ::CMoteur, 363  
ayoaron ::objets ::CVaisseau, 407  
ayoaron ::objets ::CVaisseauX1, 451  
ayoaron ::physic ::CObjet, 476  
ayoaron ::physic ::CSolideRigide, 534  
HasJoystick  
ayoaron ::event ::CEventHandler, 191  
ayoaron ::event ::CGameEventHandler, 195  
ayoaron ::event ::CMenuEventHandler, 199  
hasNick  
ayoaron ::reseau ::CNetworkMachine, 593  
hdl  
ayoaron ::core ::CXMLExtractor, 127  
i\_data  
ayoaron ::core ::CPlayerOptions, 116  
ayoaron ::data ::SConfigFileData, 185  
ayoaron ::event ::CEngineEvent, 187  
IA\_PLAYER  
ayoaron ::core ::CPlayerOptions, 116  
id  
ayoaron ::reseau ::CNetworkMachine, 596  
idevice  
ayoaron ::graphic ::CGraphicsEngine, 245  
IDsecteur  
ayoaron ::physic ::CSecteur, 501  
INGAME\_MENU  
ayoaron ::graphic ::CGuiSystem, 252  
ingame\_menu  
ayoaron ::graphic ::CGuiSystem, 267  
init  
ayoaron ::core ::CGame, 77  
INTERNET  
ayoaron ::reseau ::CIinternetNetworkInterface, 542  
ayoaron ::reseau ::CLANNetworkInterface, 548  
ayoaron ::reseau ::CNetworkInterface, 586  
INTERNET\_CLIENT  
ayoaron ::reseau ::CNetworkEngine, 556  
INTERNET\_SERVER  
ayoaron ::reseau ::CNetworkEngine, 556

io  
    ayoaron ::core ::CConsole, 57  
    ayoaron ::reseau ::CNetworkEngine, 580  
    ayoaron ::reseau ::CNetworkMachine, 596

irrRenderer  
    ayoaron ::graphic ::CGraphicsEngine, 245

is\_running  
    ayoaron ::reseau ::CNetworkEngine, 567

isAddressRegistered  
    ayoaron ::reseau ::CNetworkEngine, 567

isConsoleVisible  
    ayoaron ::graphic ::CGuiSystem, 263

isDoomed  
    ayoaron ::audio ::CSound, 23

isFull  
    ayoaron ::core ::CGameRound, 103

isIn  
    ayoaron ::data ::CConfigFile, 128

isKeyPressed  
    ayoaron ::event ::CEventHandler, 190  
    ayoaron ::event ::CGameEventHandler, 194  
    ayoaron ::event ::CMenuEventHandler, 198

isOfType  
    ayoaron ::data ::CConfigFile, 129

isPanelVisible  
    ayoaron ::graphic ::CGuiSystem, 263

isPaused  
    ayoaron ::core ::CGameEngine, 90

isPlaying  
    ayoaron ::audio ::CSound, 23

isRepeat  
    ayoaron ::audio ::CSound, 23

isResultanteNulle  
    ayoaron ::physic ::CObjet, 476

isReversible  
    ayoaron ::objets ::CMoteur, 351

joy  
    ayoaron ::event ::CEventHandler, 191  
    ayoaron ::event ::CGameEventHandler, 196  
    ayoaron ::event ::CMenuEventHandler, 199

KeyIsPressed  
    ayoaron ::event ::CEventHandler, 191  
    ayoaron ::event ::CGameEventHandler, 196  
    ayoaron ::event ::CMenuEventHandler, 200

l\_ais  
    ayoaron ::core ::CGameEngine, 97

l\_bool  
    ayoaron ::data ::CDataStructure, 174

l\_color  
    ayoaron ::data ::CDataStructure, 174

l\_difficulties  
    ayoaron ::core ::CRoundOptions, 118

l\_eh  
    ayoaron ::graphic ::CGraphicsEngine, 246

l\_enfant  
    ayoaron ::data ::CDataNode, 172  
    ayoaron ::objets ::CAsteroide, 316  
    ayoaron ::objets ::CMoteur, 363  
    ayoaron ::objets ::CVaisseau, 408  
    ayoaron ::objets ::CVaisseauX1, 451  
    ayoaron ::physic ::CObjet, 476  
    ayoaron ::physic ::CSolideRigide, 534

l\_f64  
    ayoaron ::data ::CDataStructure, 174

l\_machines  
    ayoaron ::reseau ::CNetworkEngine, 581

l\_maps  
    ayoaron ::core ::CGameEngine, 97

l\_mat4  
    ayoaron ::data ::CDataStructure, 174

l\_meshs  
    ayoaron ::graphic ::CGraphicsEngine, 246

l\_objets  
    ayoaron ::physic ::CSecteur, 501  
    ayoaron ::physic ::CWorld, 540

l\_players  
    ayoaron ::core ::CGameRound, 106  
    ayoaron ::reseau ::CNetworkMachine, 596

l\_po  
    ayoaron ::core ::CRoundOptions, 118

l\_receive\_queue  
    ayoaron ::reseau ::CNetworkEngine, 581

`I_s32`  
    `ayoaron ::data ::CDataStructure, 174`

`I_secteur`  
    `ayoaron ::physic ::CGalaxie, 462`

`I_snd`  
    `ayoaron ::audio ::CSoundEngine, 46`

`I_str`  
    `ayoaron ::data ::CDataStructure, 175`

`I_u32`  
    `ayoaron ::data ::CDataStructure, 175`

`I_vec3`  
    `ayoaron ::data ::CDataStructure, 175`

`LAN`  
    `ayoaron ::reseau ::CIInternetNetworkInterface, 542`

`ayoaron ::reseau ::CLANNetworkInterface, 548`

`ayoaron ::reseau ::CNetworkInterface, 586`

`LAN_CLIENT`  
    `ayoaron ::reseau ::CNetworkEngine, 556`

`LAN_SERVER`  
    `ayoaron ::reseau ::CNetworkEngine, 556`

`lastFPS`  
    `ayoaron ::graphic ::CGraphicsEngine, 246`

`level`  
    `ayoaron ::core ::CPlayerOptions, 116`

`limang`  
    `ayoaron ::graphic ::CGaugeVitesse, 223`

`limits`  
    `ayoaron ::graphic ::CGaugeVitesse, 223`

`listenGamePropositions`  
    `ayoaron ::reseau ::CNetworkEngine, 568`

`listenMulti3secondsUDP`  
    `ayoaron ::reseau ::CNetworkEngine, 568`

`LOAD`  
    `ayoaron ::event ::CEngineEvent, 187`

`load`  
    `ayoaron ::core ::CGameRound, 103`

`LOAD_MENU`  
    `ayoaron ::graphic ::CGuiSystem, 252`

`loadData`  
    `ayoaron ::physic ::CGalaxie, 460`

`ayoaron ::physic ::CSecteur, 499`

`ayoaron ::physic ::CWorld, 539`

`loadGame`  
    `ayoaron ::graphic ::CGuiSystem, 263`

`loadMap`  
    `ayoaron ::core ::CGameRound, 104`

`loadObjetData`  
    `ayoaron ::objets ::CAsteroide, 306`

`ayoaron ::objets ::CMoteur, 352`

`ayoaron ::objets ::CVaisseau, 393`

`ayoaron ::objets ::CVaisseauX1, 434`

`ayoaron ::physic ::COobjet, 472`

`ayoaron ::physic ::CSolideRigide, 524`

`loadPlayers`  
    `ayoaron ::core ::CGameRound, 105`

`loadSound`  
    `ayoaron ::audio ::CSound, 23`

`LOCAL`  
    `ayoaron ::physic, 19`

`log`  
    `ayoaron ::core ::CLogger, 109`

`ayoaron ::core ::CSharedInstance, 125`

`logEvent`  
    `ayoaron ::core ::CLogger, 109`

`logstr`  
    `ayoaron ::objets ::CBoiteNoire, 319`

`logvect`  
    `ayoaron ::objets ::CBoiteNoire, 320`

`main`  
    `main.cpp, 723`

`main.cpp, 722`  
    `main, 723`

`MAIN_MENU`  
    `ayoaron ::graphic ::CGuiSystem, 252`

`make`  
    `ayoaron ::core ::CShared, 122`

`makeAiName`  
    `ayoaron ::core ::CGameEngine, 90`

`makeFromCollect`  
    `ayoaron ::core ::CGameRound, 105`

`map_config`  
    `ayoaron ::core ::CRoundOptions, 118`

`map_name`  
    `ayoaron ::core ::CRoundOptions, 118`

`map_path`  
    `ayoaron ::core ::CGameEngine, 97`

`MATRIX`  
    `ayoaron ::data, 15`

max  
    ayoaron ::objets ::CPousMouv, 367

maxdist  
    ayoaron ::audio ::CSoundOptions, 49

mesh  
    ayoaron ::physic ::CObjet, 476

mesh\_file  
    ayoaron ::core ::CAvailableMap, 53

mindist  
    ayoaron ::audio ::CSoundOptions, 49

MomentResultant  
    ayoaron ::physic ::CObjet, 476

momentResultant  
    ayoaron ::objets ::CAsteriode, 316  
    ayoaron ::objets ::CMoteur, 364  
    ayoaron ::objets ::CVaisseau, 408  
    ayoaron ::objets ::CVaisseauX1, 451  
    ayoaron ::physic ::CSolideRigide, 534

moteurs  
    ayoaron ::graphic ::CGaugeMoteur, 213  
    ayoaron ::objets ::CVaisseau, 408  
    ayoaron ::objets ::CVaisseauX1, 452

mouse  
    ayoaron ::event ::CEventHandler, 191  
    ayoaron ::event ::CGameEventHandler, 196  
    ayoaron ::event ::CMenuEventHandler, 200

MOVE\_NODE  
    ayoaron ::event ::CEngineEvent, 187

moveNode  
    ayoaron ::graphic ::CSceneGraph, 270

moveNodeAndPropagate  
    ayoaron ::graphic ::CSceneGraph, 271

mround  
    ayoaron ::objets, 18

MSG\_NICK  
    ayoaron ::event ::CEngineEvent, 186

multicast\_address  
    ayoaron ::reseau ::CNetworkEngine, 581

multicastReceive  
    ayoaron ::reseau ::CNetworkEngine, 569

mutex\_gui

ayoaron ::core ::CSharedInstance, 125

n

    ayoaron ::core ::CGame, 79

name  
    ayoaron ::core ::CAvailableMap, 54  
    ayoaron ::core ::CPlayer, 114

ne  
    ayoaron ::audio ::CSoundEngine, 46  
    ayoaron ::core ::CEngine, 72  
    ayoaron ::core ::CGameEngine, 98  
    ayoaron ::graphic ::CGraphicsEngine, 246  
    ayoaron ::physic ::CPhysicEngine, 491  
    ayoaron ::reseau ::CNetworkEngine, 581

NETWORK\_PLAYER  
    ayoaron ::core ::CPlayerOptions, 116

network\_buffer  
    ayoaron ::reseau ::CNetworkEngine, 581  
    ayoaron ::reseau ::CNetworkMachine, 596

NEW\_PLAYER  
    ayoaron ::event ::CEngineEvent, 186

next\_machine\_id  
    ayoaron ::reseau ::CNetworkEngine, 581

ni  
    ayoaron ::reseau ::CNetworkEngine, 581

NID  
    ayoaron ::physic ::CObjet, 476

niveau  
    ayoaron ::data ::CDataNode, 172

Nmoteur  
    ayoaron ::graphic ::CGaugeMoteur, 213  
    ayoaron ::objets ::CVaisseau, 408  
    ayoaron ::objets ::CVaisseauX1, 452

node  
    ayoaron ::objets ::CAsteriode, 316  
    ayoaron ::objets ::CMoteur, 364  
    ayoaron ::objets ::CVaisseau, 408  
    ayoaron ::objets ::CVaisseauX1, 452  
    ayoaron ::physic ::CObjet, 476  
    ayoaron ::physic ::CSolideRigide, 534

NONE

ayoaron ::reseau ::CNetworkEngine, 556  
normalMap  
    ayoaron ::physic ::CObjet, 477  
NOTYPE  
    ayoaron ::data, 15  
  
o  
    ayoaron ::graphic ::CSceneGraph, 273  
objparent  
    ayoaron ::physic ::CObjet, 477  
ON\_CREATION  
    ayoaron ::core ::CGameRound, 101  
ON\_FINISH  
    ayoaron ::core ::CGameRound, 101  
ON\_GAME  
    ayoaron ::graphic ::CGraphicsEngine, 229  
ON\_LOADING  
    ayoaron ::graphic ::CGraphicsEngine, 229  
ON\_MENUS  
    ayoaron ::graphic ::CGraphicsEngine, 229  
ON\_NOTHING  
    ayoaron ::core ::CGameRound, 101  
ON\_PLAYING  
    ayoaron ::core ::CGameRound, 101  
onClic  
    ayoaron ::graphic ::CGuiSystem, 264  
OnEvent  
    ayoaron ::event ::CEventHandler, 191  
    ayoaron ::event ::CGameEventHand-  
        ler, 195  
    ayoaron ::event ::CMenuEventHand-  
        ler, 199  
onEvent  
    ayoaron ::event ::CEventHandler, 190  
    ayoaron ::event ::CGameEventHand-  
        ler, 194  
    ayoaron ::event ::CMenuEventHand-  
        ler, 199  
ONLY\_ERRORS  
    ayoaron ::core ::CLogger, 109  
operator\*  
    ayoaron ::objets ::CPousMouv, 367  
operator+  
    ayoaron ::objets ::CPousMouv, 367  
operator-  
    ayoaron ::objets ::CPousMouv, 367  
operator/  
    ayoaron ::objets ::CPousMouv, 367  
operator==  
    ayoaron ::event ::CEngineEvent, 187  
options  
    ayoaron ::audio ::CSoundEngine, 46  
    ayoaron ::graphic ::CGraphicsEngine, 246  
OPTIONS\_MENU  
    ayoaron ::graphic ::CGuiSystem, 252  
  
P  
    ayoaron ::objets ::CPousMouv, 368  
p  
    ayoaron ::core ::CGame, 79  
panel  
    ayoaron ::graphic ::CGauge, 206  
    ayoaron ::graphic ::CGaugeMoteur, 213  
    ayoaron ::graphic ::CGaugeVitesse, 224  
    ayoaron ::graphic ::CGuiSystem, 267  
PARENT  
    ayoaron ::physic, 19  
parent  
    ayoaron ::audio ::CSound, 27  
    ayoaron ::audio ::CSoundEngine, 46  
    ayoaron ::core ::CConsole, 58  
    ayoaron ::core ::CEngine, 73  
    ayoaron ::core ::CGameEngine, 98  
    ayoaron ::core ::CGameRound, 106  
    ayoaron ::data ::CDataNode, 172  
    ayoaron ::event ::CEventHandler, 192  
    ayoaron ::event ::CGameEventHand-  
        ler, 196  
    ayoaron ::event ::CMenuEventHand-  
        ler, 200  
    ayoaron ::graphic ::CGraphicsEngine, 246  
    ayoaron ::graphic ::CGuiSystem, 267  
    ayoaron ::graphic ::CSceneGraph, 273  
    ayoaron ::objets ::CAsteroide, 317  
    ayoaron ::objets ::CModulePilotage, 324  
    ayoaron ::objets ::CMoteur, 364  
    ayoaron ::objets ::CVaisseau, 409  
    ayoaron ::objets ::CVaisseauX1, 452  
    ayoaron ::physic ::CGalaxie, 462

ayoaron :: physic :: CPhysicEngine, 491  
ayoaron :: physic :: CSecteur, 501  
ayoaron :: physic :: CSolideRigide, 535  
ayoaron :: reseau :: CIinternetNetworkInterface, 545  
ayoaron :: reseau :: CLANNetworkInterface, 552  
ayoaron :: reseau :: CNetworkEngine, 582  
ayoaron :: reseau :: CNetworkInterface, 589  
ayoaron :: reseau :: CNetworkMachine, 596  
ayoaron :: reseau :: CNetworkPlayer, 599  
parse  
ayoaron :: data :: CConfigFile, 129  
passive\_mode  
ayoaron :: audio :: CSoundEngine, 46  
ayoaron :: core :: CEngine, 73  
ayoaron :: core :: CGameEngine, 98  
ayoaron :: graphic :: CGraphicsEngine, 246  
ayoaron :: physic :: CPhysicEngine, 491  
ayoaron :: reseau :: CNetworkEngine, 582  
pause  
ayoaron :: core :: CGameEngine, 98  
phy  
ayoaron :: audio :: CSoundEngine, 46  
ayoaron :: core :: CEngine, 73  
ayoaron :: core :: CGame, 79  
ayoaron :: core :: CGameEngine, 98  
ayoaron :: graphic :: CGraphicsEngine, 247  
ayoaron :: physic :: CPhysicEngine, 492  
ayoaron :: reseau :: CNetworkEngine, 582  
physique, 20  
pilote  
ayoaron :: objets :: CVaisseau, 409  
ayoaron :: objets :: CVaisseauX1, 452  
play\_as  
ayoaron :: core :: CRoundOptions, 118  
ayoaron :: reseau :: CNetworkEngine, 582  
playAsWhat  
ayoaron :: reseau :: CNetworkEngine, 569  
player\_name  
ayoaron :: core :: CPlayerOptions, 116  
player\_type  
ayoaron :: core :: CPlayerOptions, 116  
playSound  
ayoaron :: audio :: CSound, 24  
port\_multicast  
ayoaron :: reseau :: CNetworkEngine, 582  
port\_tcp\_server\_connection  
ayoaron :: reseau :: CNetworkEngine, 582  
port\_udp\_reception  
ayoaron :: reseau :: CNetworkEngine, 582  
pos  
ayoaron :: audio :: CSound, 27  
ayoaron :: graphic :: CGauge, 206  
ayoaron :: graphic :: CGaugeMoteur, 213  
ayoaron :: graphic :: CGaugeVitesse, 224  
position  
ayoaron :: physic :: CEffort, 457  
postPhysique  
ayoaron :: objets :: CAsteroide, 306  
ayoaron :: objets :: CMoteur, 352  
ayoaron :: objets :: CVaisseau, 393  
ayoaron :: objets :: CVaisseauX1, 435  
ayoaron :: physic :: CSolideRigide, 525  
PP  
ayoaron :: objets :: CMoteur, 364  
PP2  
ayoaron :: objets :: CMoteur, 365  
preFrame  
ayoaron :: physic :: CObjet, 472  
prePhysique  
ayoaron :: objets :: CAsteroide, 307  
ayoaron :: objets :: CMoteur, 353  
ayoaron :: objets :: CVaisseau, 393  
ayoaron :: objets :: CVaisseauX1, 436  
ayoaron :: physic :: CSolideRigide, 525  
processCommand  
ayoaron :: core :: CConsole, 56  
processEvent  
ayoaron :: audio :: CSoundEngine, 40  
ayoaron :: core :: CEngine, 67  
ayoaron :: core :: CGameEngine, 91

ayoaron ::graphic ::CGraphicsEngine, 239  
ayoaron :: physic ::CPhysicEngine, 486  
ayoaron :: reseau ::CNetworkEngine, 570  
processQueue  
    ayoaron :: audio ::CSoundEngine, 40  
    ayoaron :: core ::CEngine, 68  
    ayoaron :: core ::CGameEngine, 91  
    ayoaron :: graphic ::CGraphicsEngine, 239  
    ayoaron :: physic ::CPhysicEngine, 486  
    ayoaron :: reseau ::CNetworkEngine, 570  
processReceivedEvents  
    ayoaron :: reseau ::CNetworkEngine, 571  
processReceivedEventsQueue  
    ayoaron :: reseau ::CNetworkEngine, 572  
propositions\_mutex  
    ayoaron :: reseau ::CNetworkEngine, 582  
pushEvent  
    ayoaron :: audio ::CSoundEngine, 41  
    ayoaron :: core ::CEngine, 68  
    ayoaron :: core ::CGameEngine, 92  
    ayoaron :: graphic ::CGraphicsEngine, 240  
    ayoaron :: physic ::CPhysicEngine, 487  
    ayoaron :: reseau ::CNetworkEngine, 572  
pushReceivedEvent  
    ayoaron :: reseau ::CNetworkEngine, 573  
queue\_mutex  
    ayoaron :: audio ::CSoundEngine, 47  
    ayoaron :: core ::CEngine, 73  
    ayoaron :: core ::CGameEngine, 98  
    ayoaron :: graphic ::CGraphicsEngine, 247  
    ayoaron :: physic ::CPhysicEngine, 492  
    ayoaron :: reseau ::CNetworkEngine, 582  
queueEventIntoGame

ayoaron :: reseau ::CNetworkEngine, 573  
readConfigFile  
    ayoaron :: data ::CFileManager, 178  
readGalaxFile  
    ayoaron :: data ::CFileManager, 179  
readObjectFile  
    ayoaron :: data ::CFileManager, 179  
readSecteurFile  
    ayoaron :: data ::CFileManager, 180  
received\_events  
    ayoaron :: reseau ::CNetworkEngine, 583  
received\_propositions  
    ayoaron :: reseau ::CNetworkEngine, 583  
recGRXY  
    ayoaron :: graphic ::CGaugeVitesse, 224  
recGRZ  
    ayoaron :: graphic ::CGaugeVitesse, 224  
recGVX  
    ayoaron :: graphic ::CGaugeVitesse, 224  
recGVY  
    ayoaron :: graphic ::CGaugeVitesse, 224  
recGVZ  
    ayoaron :: graphic ::CGaugeVitesse, 224  
REFUSED  
    ayoaron :: event ::CEngineEvent, 186  
REM\_NODE  
    ayoaron :: event ::CEngineEvent, 187  
REMOVE\_MACHINE  
    ayoaron :: event ::CEngineEvent, 187  
removeNode  
    ayoaron :: graphic ::CSceneGraph, 271  
removeNodeAndPropagate  
    ayoaron :: graphic ::CSceneGraph, 272  
removeObserver  
    ayoaron :: graphic ::CSceneGraph, 272  
removePlayer  
    ayoaron :: reseau ::CNetworkMachine, 593, 594

render  
    ayoaron ::graphic ::CGuiSystem, 265  
    ayoaron ::graphic ::CSceneGraph,  
        272

REPERE  
    ayoaron ::physic, 19

requestClose  
    ayoaron ::graphic ::CGraphicsEngine,  
        241

resetconsigne  
    ayoaron ::objets ::CVaisseau, 394  
    ayoaron ::objets ::CVaisseauX1, 437

resetEffort  
    ayoaron ::objets ::CAsteriode, 307  
    ayoaron ::objets ::CMoteur, 354  
    ayoaron ::objets ::CVaisseau, 394  
    ayoaron ::objets ::CVaisseauX1, 437  
    ayoaron ::physic ::CObjet, 472  
    ayoaron ::physic ::CSolideRigide, 525

resetmouvement  
    ayoaron ::objets ::CVaisseau, 394  
    ayoaron ::objets ::CVaisseauX1, 437  
    ayoaron ::physic ::CObjet, 473

resetpousse  
    ayoaron ::objets ::CVaisseau, 395  
    ayoaron ::objets ::CVaisseauX1, 438

resolution  
    ayoaron ::graphic ::CGraphicsOptions,  
        249

reversible  
    ayoaron ::graphic ::datamot, 282

ro  
    ayoaron ::core ::CGameRound, 107

rolloff  
    ayoaron ::audio ::CSoundOptions,  
        49

root  
    ayoaron ::data ::CDataNode, 173

round\_options, 599

run  
    ayoaron ::core ::CGame, 77

RUN\_MULTI  
    ayoaron ::event ::CEngineEvent, 186

running  
    ayoaron ::reseau ::CNetworkEngine,  
        583

s  
    ayoaron ::core ::CConsole, 58  
    ayoaron ::core ::CGame, 79

    s\_data  
        ayoaron ::core ::CPlayerOptions, 116  
        ayoaron ::data ::SConfigFileData, 185  
        ayoaron ::event ::CEngineEvent, 187

    s\_multicast  
        ayoaron ::reseau ::CNetworkEngine,  
            583

    s\_tcp  
        ayoaron ::reseau ::CNetworkEngine,  
            583

    s\_udp  
        ayoaron ::reseau ::CNetworkMachine,  
            596

    s\_udp\_in  
        ayoaron ::reseau ::CNetworkPlayer,  
            599

    s\_udp  
        ayoaron ::reseau ::CNetworkMachine,  
            596

    s\_udp\_in  
        ayoaron ::reseau ::CNetworkEngine,  
            583

SAVE\_MENU  
    ayoaron ::graphic ::CGuiSystem, 252

scale  
    ayoaron ::graphic ::CGauge, 206  
    ayoaron ::graphic ::CGaugeMoteur,  
        214

    ayoaron ::graphic ::CGaugeVitesse,  
        224

    ayoaron ::graphic ::CVaisseauPanel,  
        281

script\_filename  
    ayoaron ::core ::CAvailableAi, 52

se  
    ayoaron ::audio ::CSoundEngine, 47  
    ayoaron ::core ::CEngine, 73  
    ayoaron ::core ::CGameEngine, 98  
    ayoaron ::graphic ::CGraphicsEngine,  
        247

    ayoaron ::physic ::CPhysicEngine,  
        492

    ayoaron ::reseau ::CNetworkEngine,  
        583

    secparent  
        ayoaron ::physic ::CObjet, 477

    secteurcourant  
        ayoaron ::physic ::CGalaxie, 462

SELF  
    ayoaron ::core ::CPlayerOptions, 116

SEND\_GAME\_PROP  
    ayoaron ::event ::CEngineEvent, 186

sendAdd  
    ayoaron :: reseau :: CIinternetNetworkInterface, 542  
    ayoaron :: reseau :: CLANNetworkInterface, 548  
    ayoaron :: reseau :: CNetworkInterface, 586  
sendEventTCP  
    ayoaron :: reseau :: CNetworkEngine, 573  
sendEventToAll  
    ayoaron :: reseau :: CNetworkEngine, 574  
sendEventUDP  
    ayoaron :: reseau :: CNetworkEngine, 574  
sendMessageToGame  
    ayoaron :: audio :: CSoundEngine, 41  
    ayoaron :: core :: CEngine, 69  
    ayoaron :: core :: CGameEngine, 92  
    ayoaron :: graphic :: CGraphicsEngine, 241  
    ayoaron :: physic :: CPhysicEngine, 487  
    ayoaron :: reseau :: CNetworkEngine, 574  
sendMessageToGraphics  
    ayoaron :: audio :: CSoundEngine, 42  
    ayoaron :: core :: CEngine, 70  
    ayoaron :: core :: CGameEngine, 93  
    ayoaron :: graphic :: CGraphicsEngine, 242  
    ayoaron :: physic :: CPhysicEngine, 488  
    ayoaron :: reseau :: CNetworkEngine, 574  
sendMessageToNetwork  
    ayoaron :: audio :: CSoundEngine, 43  
    ayoaron :: core :: CEngine, 70  
    ayoaron :: core :: CGameEngine, 94  
    ayoaron :: graphic :: CGraphicsEngine, 242  
    ayoaron :: physic :: CPhysicEngine, 489  
    ayoaron :: reseau :: CNetworkEngine, 575  
sendMessageToPhysic  
    ayoaron :: audio :: CSoundEngine, 43  
    ayoaron :: core :: CEngine, 71  
    ayoaron :: core :: CGameEngine, 94  
    ayoaron :: graphic :: CGraphicsEngine, 243  
    ayoaron :: physic :: CPhysicEngine, 489  
    ayoaron :: reseau :: CNetworkEngine, 576  
sendMessageToSound  
    ayoaron :: audio :: CSoundEngine, 44  
    ayoaron :: core :: CEngine, 71  
    ayoaron :: core :: CGameEngine, 95  
    ayoaron :: graphic :: CGraphicsEngine, 243  
    ayoaron :: physic :: CPhysicEngine, 490  
    ayoaron :: reseau :: CNetworkEngine, 576  
sendMove  
    ayoaron :: reseau :: CIinternetNetworkInterface, 543  
    ayoaron :: reseau :: CLANNetworkInterface, 549  
    ayoaron :: reseau :: CNetworkInterface, 587  
sendMulticast  
    ayoaron :: reseau :: CNetworkEngine, 577  
sendRem  
    ayoaron :: reseau :: CIinternetNetworkInterface, 543  
    ayoaron :: reseau :: CLANNetworkInterface, 549  
    ayoaron :: reseau :: CNetworkInterface, 587  
sendToAllGameTCP  
    ayoaron :: reseau :: CIinternetNetworkInterface, 544  
    ayoaron :: reseau :: CLANNetworkInterface, 550  
    ayoaron :: reseau :: CNetworkInterface, 588  
sendToAllGameUDP  
    ayoaron :: reseau :: CIinternetNetworkInterface, 545  
    ayoaron :: reseau :: CLANNetworkInterface, 551  
    ayoaron :: reseau :: CNetworkInterface, 588  
sendToServerTCP  
    ayoaron :: reseau :: CIinternetNetworkInterface, 545

ayoaron ::reseau ::CLANNetworkInterface, 551  
ayoaron ::reseau ::CNetworkInterface, 589  
serialize  
    ayoaron ::event ::CEngineEvent, 187  
    ayoaron ::event ::CSerializableVector3df, 201  
serverThreadTCPReceive  
    ayoaron ::core ::CConsole, 57  
    ayoaron ::reseau ::CNetworkEngine, 577  
setAbsoluteRitesse  
    ayoaron ::objets ::CAsteriode, 307  
    ayoaron ::objets ::CMoteur, 354  
    ayoaron ::objets ::CVaisseau, 395  
    ayoaron ::objets ::CVaisseauX1, 438  
    ayoaron ::physic ::CSolideRigide, 526  
setAbsoluteVitesse  
    ayoaron ::objets ::CAsteriode, 307  
    ayoaron ::objets ::CMoteur, 354  
    ayoaron ::objets ::CVaisseau, 396  
    ayoaron ::objets ::CVaisseauX1, 439  
    ayoaron ::physic ::CSolideRigide, 526  
setBackground  
    ayoaron ::graphic ::CGauge, 204  
    ayoaron ::graphic ::CGaugeMoteur, 210  
    ayoaron ::graphic ::CGaugeVitesse, 217  
    ayoaron ::graphic ::CVaisseauPanel, 280  
setCadres  
    ayoaron ::graphic ::CGaugeMoteur, 210  
setConsigne  
    ayoaron ::objets ::CModulePilotage, 323  
setCurseur  
    ayoaron ::graphic ::CGaugeVitesse, 218  
setData  
    ayoaron ::data ::CData, 140–144  
    ayoaron ::data ::CDataNode, 167–171  
setDataFromStr  
    ayoaron ::data ::CData, 145  
setDopplerProperties  
    ayoaron ::audio ::CSoundEngine, 44  
setForce  
    ayoaron ::physic ::CEffort, 456  
setInertie  
    ayoaron ::physic ::CObjet, 473  
setLimit  
    ayoaron ::graphic ::CGaugeVitesse, 218  
setLimitAngulaire  
    ayoaron ::graphic ::CGaugeVitesse, 219  
setListenerPos  
    ayoaron ::audio ::CSoundEngine, 44  
setLocalRitesse  
    ayoaron ::objets ::CAsteriode, 308  
    ayoaron ::objets ::CMoteur, 355  
    ayoaron ::objets ::CVaisseau, 396  
    ayoaron ::objets ::CVaisseauX1, 439  
    ayoaron ::physic ::CSolideRigide, 526  
setLocalVitesse  
    ayoaron ::objets ::CAsteriode, 309  
    ayoaron ::objets ::CMoteur, 355  
    ayoaron ::objets ::CVaisseau, 397  
    ayoaron ::objets ::CVaisseauX1, 440  
    ayoaron ::physic ::CSolideRigide, 527  
setLogFilename  
    ayoaron ::core ::CLogger, 109  
setMatrix3  
    ayoaron ::objets ::CAsteriode, 309  
    ayoaron ::objets ::CMoteur, 356  
    ayoaron ::objets ::CVaisseau, 398  
    ayoaron ::objets ::CVaisseauX1, 441  
    ayoaron ::physic ::CSolideRigide, 528  
setNmoteur  
    ayoaron ::graphic ::CGaugeMoteur, 211  
setNom  
    ayoaron ::physic ::CGalaxie, 461  
    ayoaron ::physic ::CSecteur, 500  
setObserver  
    ayoaron ::graphic ::CSceneGraph, 272  
setOriginalSize  
    ayoaron ::graphic ::CVaisseauPanel, 280  
setParent  
    ayoaron ::data ::CDataNode, 171  
    ayoaron ::graphic ::CGuiSystem, 265  
    ayoaron ::objets ::CAsteriode, 310  
    ayoaron ::objets ::CMoteur, 357  
    ayoaron ::objets ::CVaisseau, 398  
    ayoaron ::objets ::CVaisseauX1, 441

ayoaron :: physic :: CSolideRigide, 528  
ayoaron :: reseau :: CNetworkPlayer, 599  
setPitch  
    ayoaron :: audio :: CSound, 24  
setPlayAs  
    ayoaron :: reseau :: CNetworkEngine, 578  
setPos  
    ayoaron :: audio :: CSound, 25  
setPosition  
    ayoaron :: graphic :: CGauge, 205  
    ayoaron :: graphic :: CGaugeMoteur, 211  
    ayoaron :: graphic :: CGaugeVitesse, 219  
    ayoaron :: physic :: CObjet, 473  
setRectGRXY  
    ayoaron :: graphic :: CGaugeVitesse, 220  
setRectGRZ  
    ayoaron :: graphic :: CGaugeVitesse, 220  
setRectGVX  
    ayoaron :: graphic :: CGaugeVitesse, 221  
setRectGVY  
    ayoaron :: graphic :: CGaugeVitesse, 221  
setRectGVZ  
    ayoaron :: graphic :: CGaugeVitesse, 222  
setRegime  
    ayoaron :: graphic :: CGaugeMoteur, 212  
    ayoaron :: objets :: CMoteur, 357  
setRelativePosition  
    ayoaron :: objets :: CAsteroide, 311  
    ayoaron :: objets :: CMoteur, 358  
    ayoaron :: objets :: CVaisseau, 399  
    ayoaron :: objets :: CVaisseauX1, 442  
    ayoaron :: physic :: CSolideRigide, 529  
setRelativeRitesse  
    ayoaron :: objets :: CAsteroide, 311  
    ayoaron :: objets :: CMoteur, 359  
    ayoaron :: objets :: CVaisseau, 400  
    ayoaron :: objets :: CVaisseauX1, 443  
    ayoaron :: physic :: CSolideRigide, 530  
setRelativeRotation  
    ayoaron :: objets :: CAsteroide, 312  
    ayoaron :: objets :: CMoteur, 359  
    ayoaron :: objets :: CVaisseau, 400  
    ayoaron :: objets :: CVaisseauX1, 443  
    ayoaron :: physic :: CSolideRigide, 531  
setRelativeVitesse  
    ayoaron :: objets :: CAsteroide, 313  
    ayoaron :: objets :: CMoteur, 360  
    ayoaron :: objets :: CVaisseau, 401  
    ayoaron :: objets :: CVaisseauX1, 444  
    ayoaron :: physic :: CSolideRigide, 531  
setScale  
    ayoaron :: graphic :: CGauge, 205  
    ayoaron :: graphic :: CGaugeMoteur, 212  
    ayoaron :: graphic :: CGaugeVitesse, 222  
    ayoaron :: physic :: CObjet, 474  
setSecteurCourant  
    ayoaron :: physic :: CGalaxie, 461  
setState  
    ayoaron :: graphic :: CGraphicsEngine, 244  
setType  
    ayoaron :: audio :: CSound, 25  
setVector  
    ayoaron :: objets :: CAsteroide, 313  
    ayoaron :: objets :: CMoteur, 361  
    ayoaron :: objets :: CVaisseau, 402  
    ayoaron :: objets :: CVaisseauX1, 445  
    ayoaron :: physic :: CSolideRigide, 532  
setVit  
    ayoaron :: audio :: CSound, 26  
setVitesse  
    ayoaron :: physic :: CObjet, 475  
setVolume  
    ayoaron :: audio :: CSound, 26  
sfx  
    ayoaron :: physic :: CGalaxie, 462  
sg  
    ayoaron :: graphic :: CGraphicsEngine, 247  
shadows  
    ayoaron :: graphic :: CGraphicsOptions, 249  
showErrorMessage  
    ayoaron :: graphic :: CGuiSystem, 265  
SINT  
    ayoaron :: data, 15  
smgr  
    ayoaron :: audio :: CSoundEngine, 47

ayoaron ::graphic ::CSceneGraph, 273  
ayoaron ::physic ::CSecteur, 501  
ayoaron ::physic ::CWorld, 540  
smgr\_mutex  
    ayoaron ::graphic ::CSceneGraph, 273  
snd  
    ayoaron ::audio ::CSound, 27  
snd1  
    ayoaron ::objets ::CMoteur, 365  
snd2  
    ayoaron ::objets ::CMoteur, 365  
SND\_2D\_INSTANT  
    ayoaron ::audio, 13  
SND\_2D\_MUSIC  
    ayoaron ::audio, 13  
SND\_3D\_CONTINUOUS  
    ayoaron ::audio, 13  
SND\_3D\_INSTANT  
    ayoaron ::audio, 13  
sndsource  
    ayoaron ::audio ::CSound, 27  
sndtype  
    ayoaron ::audio ::CSound, 27  
SOUND\_TYPE  
    ayoaron ::audio, 13  
srho  
    ayoaron ::objets ::CAsteroide, 317  
SRV  
    ayoaron ::objets ::CModulePilotage, 325  
state  
    ayoaron ::audio ::CSoundEngine, 47  
    ayoaron ::core ::CEngine, 73  
    ayoaron ::core ::CGameEngine, 99  
    ayoaron ::core ::CGameRound, 107  
    ayoaron ::graphic ::CGraphicsEngine, 247  
    ayoaron ::physic ::CPhysicEngine, 492  
    ayoaron ::reseau ::CNetworkEngine, 583  
STATE\_LOADING\_ROUND  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228  
    ayoaron ::physic ::CPhysicEngine, 480  
    ayoaron ::reseau ::CNetworkEngine, 556  
STATE\_NOTHING  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228  
    ayoaron ::physic ::CPhysicEngine, 480  
    ayoaron ::reseau ::CNetworkEngine, 556  
STATE\_PLAYING  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228  
    ayoaron ::physic ::CPhysicEngine, 480  
    ayoaron ::reseau ::CNetworkEngine, 556  
STATE\_PLAYING\_INTRO  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228  
    ayoaron ::physic ::CPhysicEngine, 480  
    ayoaron ::reseau ::CNetworkEngine, 556  
STATE\_QUITTING  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228  
    ayoaron ::physic ::CPhysicEngine, 480  
    ayoaron ::reseau ::CNetworkEngine, 556  
STATE\_WAITING\_PLAYERS  
    ayoaron ::audio ::CSoundEngine, 31  
    ayoaron ::core ::CEngine, 61  
    ayoaron ::core ::CGameEngine, 83  
    ayoaron ::graphic ::CGraphicsEngine, 228

ayoaron :: physic :: CPhysicEngine, 480  
ayoaron :: reseau :: CNetworkEngine, 556  
state\_mutex  
    ayoaron :: audio :: CSoundEngine, 47  
    ayoaron :: core :: CEngine, 74  
    ayoaron :: core :: CGameEngine, 99  
    ayoaron :: graphic :: CGraphicsEngine, 247  
    ayoaron :: physic :: CPhysicEngine, 492  
    ayoaron :: reseau :: CNetworkEngine, 583  
still\_running  
    ayoaron :: core :: CGame, 79  
still\_running\_mutex  
    ayoaron :: core :: CGame, 79  
stooooop  
    ayoaron :: core :: CGame, 78  
stopSound  
    ayoaron :: audio :: CSound, 26  
str2dtyp  
    ayoaron :: data, 15  
STRING  
    ayoaron :: data, 15  
strtowchar  
    ayoaron :: data, 15  
sum  
    ayoaron :: objets :: CPousMouv, 367  
SV  
    ayoaron :: objets :: CModulePilotage, 325  
sys  
    ayoaron :: graphic :: CGraphicsEngine, 248  
tcp\_acceptor  
    ayoaron :: core :: CConsole, 58  
    ayoaron :: reseau :: CNetworkEngine, 584  
tcpAsyncReceive  
    ayoaron :: reseau :: CNetworkMachine, 594  
tcpSend  
    ayoaron :: reseau :: CNetworkMachine, 595  
tcpThreadSend  
    ayoaron :: reseau :: CNetworkMachine, 595  
team\_nb  
    ayoaron :: core :: CPlayerOptions, 116  
temps  
    ayoaron :: objets :: CBoiteNoire, 320  
    ayoaron :: objets :: CModulePilotage, 325  
    ayoaron :: objets :: CMoteur, 365  
then  
    ayoaron :: core :: CGame, 79  
threadSendMulticast  
    ayoaron :: reseau :: CNetworkEngine, 578  
toggleconsigne  
    ayoaron :: objets :: CVaisseau, 403  
    ayoaron :: objets :: CVaisseauX1, 446  
toggleConsole  
    ayoaron :: graphic :: CGuiSystem, 266  
toggleIngameMenu  
    ayoaron :: graphic :: CGuiSystem, 266  
togglePanel  
    ayoaron :: graphic :: CGuiSystem, 266  
togglePause  
    ayoaron :: core :: CGameEngine, 95  
tourneCamera  
    ayoaron :: graphic :: CSceneGraph, 273  
translateCamera  
    ayoaron :: graphic :: CSceneGraph, 273  
txtVaiss  
    ayoaron :: graphic :: CGuiSystem, 266  
type  
    ayoaron :: core :: CLogger, 110  
    ayoaron :: event :: CEngineEvent, 188  
    ayoaron :: reseau :: CIInternetNetworkInterface, 546  
    ayoaron :: reseau :: CLANNetworkInterface, 552  
    ayoaron :: reseau :: CNetworkInterface, 589  
udp\_ep  
    ayoaron :: reseau :: CNetworkMachine, 596  
udp\_remote\_endpoint  
    ayoaron :: reseau :: CNetworkEngine, 584  
udpAsyncRead  
    ayoaron :: reseau :: CNetworkEngine, 579

udpSend  
    ayoaron ::reseau ::CNetworkMachine,  
        595

UINT  
    ayoaron ::data, 15

unmake  
    ayoaron ::core ::CShared, 122

v\_data  
    ayoaron ::event ::CEngineEvent, 188

vaisseaujoueur  
    ayoaron ::physic ::CSecteur, 501

valmot  
    ayoaron ::graphic ::datamot, 282

valsmot  
    ayoaron ::graphic ::CGaugeMoteur,  
        214

VECTOR  
    ayoaron ::data, 15

vectortoString  
    ayoaron ::graphic ::CGuiSystem, 266

vit  
    ayoaron ::audio ::CSound, 27

vitavar  
    ayoaron ::objets ::CVaisseau, 403  
    ayoaron ::objets ::CVaisseauX1, 446

vite  
    ayoaron ::graphic ::CGaugeVitesse,  
        225

vitgadr  
    ayoaron ::objets ::CVaisseau, 403  
    ayoaron ::objets ::CVaisseauX1, 447

vithaba  
    ayoaron ::objets ::CVaisseau, 404  
    ayoaron ::objets ::CVaisseauX1, 447

vitlace  
    ayoaron ::objets ::CVaisseau, 404  
    ayoaron ::objets ::CVaisseauX1, 448

vitmax  
    ayoaron ::objets ::CVaisseau, 409  
    ayoaron ::objets ::CVaisseauX1, 452

vitroul  
    ayoaron ::objets ::CVaisseau, 405  
    ayoaron ::objets ::CVaisseauX1, 448

vittang  
    ayoaron ::objets ::CVaisseau, 405  
    ayoaron ::objets ::CVaisseauX1, 448

vmin  
    ayoaron ::objets, 19

volume

    ayoaron ::audio ::CSoundOptions,  
        49

vrot  
    ayoaron ::graphic ::CGaugeVitesse,  
        225

Vsync  
    ayoaron ::graphic ::CGraphicsOptions,  
        249

    WARNING\_OR\_ERRORS

    ayoaron ::core ::CLogger, 109

wchartostr  
    ayoaron ::data, 16

WELCOME  
    ayoaron ::event ::CEngineEvent, 186

wip  
    ayoaron ::core ::CGameEngine, 99

world  
    ayoaron ::objets ::CAsteroide, 317  
    ayoaron ::objets ::CMoteur, 365  
    ayoaron ::objets ::CVaisseau, 409  
    ayoaron ::objets ::CVaisseauX1, 453  
    ayoaron ::physic ::CSolideRigide, 535

writeGalaxFile  
    ayoaron ::data ::CFileManager, 181

writeObjectFile  
    ayoaron ::data ::CFileManager, 181

writeSavegame  
    ayoaron ::data ::CFileManager, 182

writeSecteurFile  
    ayoaron ::data ::CFileManager, 182