

## Car Dashboard Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Projet dashboard</b>	<b>1</b>
<b>2</b>	<b>Bug List</b>	<b>5</b>
<b>3</b>	<b>Hierarchical Index</b>	<b>7</b>
3.1	Class Hierarchy . . . . .	7
<b>4</b>	<b>Class Index</b>	<b>11</b>
4.1	Class List . . . . .	11
<b>5</b>	<b>File Index</b>	<b>15</b>
5.1	File List . . . . .	15
<b>6</b>	<b>Class Documentation</b>	<b>19</b>
6.1	afficheKmHenri Class Reference . . . . .	19
6.1.1	Constructor & Destructor Documentation . . . . .	20
6.1.1.1	afficheKmHenri() . . . . .	20
6.1.2	Member Function Documentation . . . . .	20
6.1.2.1	paint() . . . . .	20
6.2	aiguille_Lea Class Reference . . . . .	21
6.3	ArrowOilInna Class Reference . . . . .	22
6.3.1	Member Function Documentation . . . . .	23
6.3.1.1	paint() . . . . .	24
6.4	ArrowOilTinna Class Reference . . . . .	24
6.4.1	Member Function Documentation . . . . .	25
6.4.1.1	paint() . . . . .	26

6.5	<a href="#">ArrowSpeedometerInna Class Reference</a>	26
6.5.1	<a href="#">Member Function Documentation</a>	27
6.5.1.1	<a href="#">paint()</a>	28
6.6	<a href="#">ArrowTachometerInna Class Reference</a>	28
6.6.1	<a href="#">Member Function Documentation</a>	29
6.6.1.1	<a href="#">paint()</a>	30
6.7	<a href="#">cadran Class Reference</a>	30
6.8	<a href="#">CadranEss Class Reference</a>	32
6.9	<a href="#">cadrantcontour Class Reference</a>	33
6.9.1	<a href="#">Member Function Documentation</a>	33
6.9.1.1	<a href="#">paint()</a>	34
6.10	<a href="#">cadrantessence Class Reference</a>	34
6.10.1	<a href="#">Member Function Documentation</a>	35
6.10.1.1	<a href="#">paint()</a>	35
6.11	<a href="#">CadrantFlorian Class Reference</a>	35
6.11.1	<a href="#">Constructor &amp; Destructor Documentation</a>	36
6.11.1.1	<a href="#">CadrantFlorian()</a>	36
6.11.2	<a href="#">Member Function Documentation</a>	37
6.11.2.1	<a href="#">boundingRect()</a>	37
6.11.2.2	<a href="#">generateAngle()</a>	37
6.11.2.3	<a href="#">paint()</a>	37
6.12	<a href="#">cadrantHenri Class Reference</a>	38
6.13	<a href="#">cadranhuile Class Reference</a>	39
6.13.1	<a href="#">Member Function Documentation</a>	40
6.13.1.1	<a href="#">paint()</a>	40
6.14	<a href="#">CadranTourParMin Class Reference</a>	40
6.15	<a href="#">cadranttour Class Reference</a>	42
6.15.1	<a href="#">Member Function Documentation</a>	42
6.15.1.1	<a href="#">paint()</a>	43
6.16	<a href="#">cadrantVirtuel Class Reference</a>	43

6.17	<a href="#">cadrantvitesse Class Reference</a>	44
6.18	<a href="#">CadranVitesse Class Reference</a>	46
6.19	<a href="#">CarDialerPrototype Class Reference</a>	47
6.20	<a href="#">CarSpeedDial Class Reference</a>	48
6.21	<a href="#">clignot Class Reference</a>	49
6.22	<a href="#">clignotantInna Class Reference</a>	50
6.22.1	<a href="#">Constructor &amp; Destructor Documentation</a>	51
6.22.1.1	<a href="#">clignotantInna()</a>	52
6.22.2	<a href="#">Member Function Documentation</a>	52
6.22.2.1	<a href="#">paint()</a>	52
6.23	<a href="#">Compteur_kilometre_lea Class Reference</a>	52
6.23.1	<a href="#">Constructor &amp; Destructor Documentation</a>	53
6.23.1.1	<a href="#">Compteur_kilometre_lea()</a>	53
6.23.2	<a href="#">Member Function Documentation</a>	54
6.23.2.1	<a href="#">paint()</a>	54
6.24	<a href="#">compteurKmlInna Class Reference</a>	55
6.24.1	<a href="#">Member Function Documentation</a>	56
6.24.1.1	<a href="#">paint()</a>	56
6.25	<a href="#">EngineHeatGauge Class Reference</a>	56
6.26	<a href="#">EngineTInna Class Reference</a>	57
6.26.1	<a href="#">Member Function Documentation</a>	58
6.26.1.1	<a href="#">paint()</a>	59
6.27	<a href="#">FlorianClignotant Class Reference</a>	59
6.27.1	<a href="#">Constructor &amp; Destructor Documentation</a>	60
6.27.1.1	<a href="#">FlorianClignotant()</a>	60
6.27.2	<a href="#">Member Function Documentation</a>	61
6.27.2.1	<a href="#">boundingRect()</a>	61
6.28	<a href="#">FlorianCompteurKm Class Reference</a>	61
6.28.1	<a href="#">Constructor &amp; Destructor Documentation</a>	62
6.28.1.1	<a href="#">FlorianCompteurKm()</a>	62

6.28.2	Member Function Documentation	63
6.28.2.1	boundingRect()	63
6.28.2.2	paint()	63
6.29	FlorianVoyant Class Reference	64
6.29.1	Member Function Documentation	65
6.29.1.1	boundingRect()	65
6.29.1.2	paint()	65
6.30	fuel_guage Class Reference	66
6.31	FuelEngineInna Class Reference	67
6.31.1	Member Function Documentation	68
6.31.1.1	paint()	68
6.32	FuelGauge Class Reference	69
6.33	FuelGauge_lea Class Reference	70
6.33.1	Constructor & Destructor Documentation	71
6.33.1.1	FuelGauge_lea()	71
6.33.2	Member Function Documentation	71
6.33.2.1	paint()	71
6.34	FuelLevelInna Class Reference	72
6.34.1	Member Function Documentation	73
6.34.1.1	paint()	74
6.35	Harc Class Reference	74
6.36	harout_scene Class Reference	76
6.37	henri_scene Class Reference	78
6.38	horloge_lea Class Reference	79
6.39	hugo_boite_vitesse Class Reference	81
6.39.1	Constructor & Destructor Documentation	82
6.39.1.1	hugo_boite_vitesse()	82
6.39.2	Member Function Documentation	82
6.39.2.1	paint()	82
6.40	hugo_Compteur Class Reference	83

6.40.1	Constructor & Destructor Documentation . . . . .	84
6.40.1.1	hugo_Compteur() . . . . .	84
6.40.2	Member Function Documentation . . . . .	85
6.40.2.1	paint() . . . . .	85
6.41	hugo_ecran Class Reference . . . . .	86
6.41.1	Constructor & Destructor Documentation . . . . .	87
6.41.1.1	hugo_ecran() . . . . .	87
6.41.2	Member Function Documentation . . . . .	87
6.41.2.1	paint() . . . . .	87
6.42	hugo_MyGraphicsitem Class Reference . . . . .	88
6.42.1	Constructor & Destructor Documentation . . . . .	89
6.42.1.1	hugo_MyGraphicsitem() . . . . .	89
6.42.2	Member Function Documentation . . . . .	90
6.42.2.1	paint() . . . . .	90
6.43	hugo_scene Class Reference . . . . .	90
6.44	hugo_voyant_warning Class Reference . . . . .	92
6.44.1	Constructor & Destructor Documentation . . . . .	93
6.44.1.1	hugo_voyant_warning() . . . . .	93
6.44.2	Member Function Documentation . . . . .	93
6.44.2.1	MAJ() . . . . .	93
6.44.2.2	paint() . . . . .	93
6.45	hugo_voyants_clignotant Class Reference . . . . .	94
6.45.1	Constructor & Destructor Documentation . . . . .	95
6.45.1.1	hugo_voyants_clignotant() . . . . .	95
6.45.2	Member Function Documentation . . . . .	95
6.45.2.1	paint() . . . . .	95
6.46	hugo_voyants_simples Class Reference . . . . .	96
6.46.1	Constructor & Destructor Documentation . . . . .	97
6.46.1.1	hugo_voyants_simples() . . . . .	97
6.46.2	Member Function Documentation . . . . .	97

6.46.2.1	paint()	97
6.47	IconInna Class Reference	98
6.47.1	Member Function Documentation	99
6.47.1.1	init()	99
6.48	iconOnOff Class Reference	100
6.49	iconOnOffInna Class Reference	101
6.49.1	Constructor & Destructor Documentation	102
6.49.1.1	iconOnOffInna()	102
6.49.2	Member Function Documentation	103
6.49.2.1	paint()	103
6.50	InfoWindowInna Class Reference	103
6.50.1	Member Function Documentation	104
6.50.1.1	paint()	104
6.51	inna_scene Class Reference	105
6.52	jauge_temperature_lea Class Reference	106
6.52.1	Constructor & Destructor Documentation	107
6.52.1.1	jauge_temperature_lea()	107
6.52.2	Member Function Documentation	108
6.52.2.1	paint()	108
6.53	jaugeClignotantHenri Class Reference	109
6.54	jaugeEssenceHenri Class Reference	110
6.55	jaugeTemperatureHenri Class Reference	111
6.56	jaugeToursMinuteHenri Class Reference	112
6.57	jaugeVirtuel Class Reference	113
6.58	Jonas_compteur Class Reference	114
6.58.1	Constructor & Destructor Documentation	115
6.58.1.1	Jonas_compteur() [1/2]	116
6.58.1.2	Jonas_compteur() [2/2]	116
6.58.2	Member Function Documentation	116
6.58.2.1	boundingRect()	116



6.58.2.2	<a href="#">paint()</a>	117
6.58.2.3	<a href="#">speedToAngle()</a>	117
6.59	<a href="#">jonas_compteurKm Class Reference</a>	118
6.59.1	<a href="#">Constructor &amp; Destructor Documentation</a>	119
6.59.1.1	<a href="#">jonas_compteurKm()</a>	119
6.59.2	<a href="#">Member Function Documentation</a>	119
6.59.2.1	<a href="#">boundingRect()</a>	119
6.59.2.2	<a href="#">paint()</a>	119
6.60	<a href="#">Jonas_scene Class Reference</a>	120
6.61	<a href="#">jonas_voyant_clignotant Class Reference</a>	122
6.61.1	<a href="#">Constructor &amp; Destructor Documentation</a>	123
6.61.1.1	<a href="#">jonas_voyant_clignotant()</a>	123
6.61.2	<a href="#">Member Function Documentation</a>	123
6.61.2.1	<a href="#">boundingRect()</a>	123
6.61.2.2	<a href="#">paint()</a>	123
6.61.2.3	<a href="#">updateTurn()</a>	124
6.62	<a href="#">jonas_voyant_simple Class Reference</a>	124
6.62.1	<a href="#">Member Function Documentation</a>	125
6.62.1.1	<a href="#">boundingRect()</a>	125
6.62.1.2	<a href="#">paint()</a>	125
6.63	<a href="#">jonas_voyant_warning Class Reference</a>	126
6.63.1	<a href="#">Member Function Documentation</a>	127
6.63.1.1	<a href="#">boundingRect()</a>	127
6.63.1.2	<a href="#">paint()</a>	127
6.64	<a href="#">karim_scene Class Reference</a>	128
6.65	<a href="#">lea_clignottants Class Reference</a>	130
6.65.1	<a href="#">Constructor &amp; Destructor Documentation</a>	131
6.65.1.1	<a href="#">lea_clignottants()</a>	131
6.65.2	<a href="#">Member Function Documentation</a>	131
6.65.2.1	<a href="#">MAJ()</a>	131

6.66	<a href="#">Lea_scene Class Reference</a>	132
6.67	<a href="#">lea_SceneDeFond Class Reference</a>	133
6.68	<a href="#">LEDS Class Reference</a>	134
6.69	<a href="#">Leo_display Class Reference</a>	135
6.70	<a href="#">Leo_gauge Class Reference</a>	136
6.71	<a href="#">Leo_indicator Class Reference</a>	138
6.72	<a href="#">Leo_label Class Reference</a>	139
6.73	<a href="#">Leo_object Class Reference</a>	140
6.74	<a href="#">Leo_scene Class Reference</a>	141
6.75	<a href="#">loto_dashboard Class Reference</a>	143
6.76	<a href="#">loto_scene Class Reference</a>	144
6.77	<a href="#">MainWindow Class Reference</a>	146
6.77.1	<a href="#">Constructor &amp; Destructor Documentation</a>	147
6.77.1.1	<a href="#">MainWindow()</a>	147
6.77.2	<a href="#">Member Function Documentation</a>	147
6.77.2.1	<a href="#">acceleration()</a>	147
6.78	<a href="#">maroua_scene Class Reference</a>	148
6.79	<a href="#">mygraphicItem Class Reference</a>	150
6.80	<a href="#">objet_virtuel Class Reference</a>	151
6.80.1	<a href="#">Constructor &amp; Destructor Documentation</a>	152
6.80.1.1	<a href="#">objet_virtuel()</a>	152
6.80.2	<a href="#">Member Function Documentation</a>	152
6.80.2.1	<a href="#">getValue()</a>	152
6.80.2.2	<a href="#">getValueMax()</a>	152
6.80.2.3	<a href="#">setValue()</a>	152
6.81	<a href="#">OilGaugeInna Class Reference</a>	153
6.81.1	<a href="#">Member Function Documentation</a>	154
6.81.1.1	<a href="#">paint()</a>	154
6.82	<a href="#">onoff Class Reference</a>	155
6.83	<a href="#">onoffpaintHenri Class Reference</a>	156

6.84	portesHenri Class Reference	157
6.85	PRND_Inna Class Reference	159
6.85.1	Member Function Documentation	160
6.85.1.1	paint()	160
6.86	scene_globale Class Reference	161
6.86.1	Constructor & Destructor Documentation	163
6.86.1.1	scene_globale()	163
6.87	sceneDeFond Class Reference	163
6.88	sceneDeFondHenri Class Reference	164
6.89	SceneFlorian Class Reference	165
6.90	sceneGlobale Class Reference	167
6.91	speedometer_Lea Class Reference	168
6.91.1	Constructor & Destructor Documentation	169
6.91.1.1	speedometer_Lea()	170
6.91.2	Member Function Documentation	170
6.91.2.1	paint()	170
6.92	SpeedometerInna Class Reference	171
6.92.1	Member Function Documentation	172
6.92.1.1	paint()	172
6.93	Tachometer Class Reference	173
6.94	TachometerGauge_Lea Class Reference	174
6.94.1	Constructor & Destructor Documentation	175
6.94.1.1	TachometerGauge_Lea()	175
6.94.2	Member Function Documentation	175
6.94.2.1	paint()	176
6.95	TachometerInna Class Reference	176
6.95.1	Member Function Documentation	177
6.95.1.1	paint()	177
6.96	tachometre Class Reference	178
6.97	tempGauge Class Reference	179
6.98	voyant_Lea Class Reference	180
6.98.1	Constructor & Destructor Documentation	181
6.98.1.1	voyant_Lea()	181
6.98.2	Member Function Documentation	182
6.98.2.1	paint()	182
6.99	Voyants Class Reference	182
6.100	voyants_warning Class Reference	184
6.101	VoyantsMaroua Class Reference	185
6.102	voyantsYoussef Class Reference	186
6.103	warning_lea Class Reference	187
6.104	warninghenri Class Reference	188
6.105	Youssef_Scene Class Reference	189

<b>7</b>	<b>File Documentation</b>	<b>191</b>
7.1	<a href="#">serveur/Henri/affichekmhenri.cpp File Reference</a>	191
7.1.1	<a href="#">Detailed Description</a>	191
7.2	<a href="#">serveur/Hugo/hugo_boite_vitesse.cpp File Reference</a>	192
7.2.1	<a href="#">Detailed Description</a>	192
7.3	<a href="#">serveur/Hugo/hugo_compteur.cpp File Reference</a>	192
7.3.1	<a href="#">Detailed Description</a>	193
7.4	<a href="#">serveur/Hugo/hugo_ecran.cpp File Reference</a>	193
7.4.1	<a href="#">Detailed Description</a>	193
7.5	<a href="#">serveur/Hugo/hugo_scene.cpp File Reference</a>	194
7.5.1	<a href="#">Detailed Description</a>	194
7.6	<a href="#">serveur/Hugo/hugo_voyant_warning.cpp File Reference</a>	194
7.6.1	<a href="#">Detailed Description</a>	195
7.7	<a href="#">serveur/Hugo/hugo_voyants_simples.cpp File Reference</a>	195
7.7.1	<a href="#">Detailed Description</a>	196
7.8	<a href="#">serveur/Inna/arrowoillinna.cpp File Reference</a>	196
7.8.1	<a href="#">Detailed Description</a>	196
7.9	<a href="#">serveur/Inna/arrowoiltinna.cpp File Reference</a>	197
7.9.1	<a href="#">Detailed Description</a>	197
7.10	<a href="#">serveur/Inna/arrowspeedometerinna.cpp File Reference</a>	197
7.10.1	<a href="#">Detailed Description</a>	198
7.11	<a href="#">serveur/Inna/arrowspeedometerinna.cpp File Reference</a>	198
7.11.1	<a href="#">Detailed Description</a>	199
7.12	<a href="#">serveur/Inna/clignotantinna.cpp File Reference</a>	199
7.12.1	<a href="#">Detailed Description</a>	200
7.13	<a href="#">serveur/Inna/compteurkminna.cpp File Reference</a>	200
7.13.1	<a href="#">Detailed Description</a>	201
7.14	<a href="#">serveur/Inna/engineinna.cpp File Reference</a>	201
7.14.1	<a href="#">Detailed Description</a>	201
7.15	<a href="#">serveur/Inna/fuelengineinna.cpp File Reference</a>	202

7.15.1 Detailed Description . . . . .	202
7.16 serveur/Inna/fuellevelinna.cpp File Reference . . . . .	202
7.16.1 Detailed Description . . . . .	203
7.17 serveur/Inna/iconinna.cpp File Reference . . . . .	203
7.17.1 Detailed Description . . . . .	204
7.18 serveur/Inna/icononoffinna.cpp File Reference . . . . .	204
7.18.1 Detailed Description . . . . .	205
7.19 serveur/Inna/infowindowinna.cpp File Reference . . . . .	205
7.19.1 Detailed Description . . . . .	206
7.20 serveur/Inna/inna_scene.cpp File Reference . . . . .	206
7.20.1 Detailed Description . . . . .	206
7.21 serveur/Inna/oilgaugeinna.cpp File Reference . . . . .	206
7.21.1 Detailed Description . . . . .	207
7.22 serveur/Inna/prnd_inna.cpp File Reference . . . . .	207
7.22.1 Detailed Description . . . . .	208
7.23 serveur/Inna/speedometerinna.cpp File Reference . . . . .	208
7.23.1 Detailed Description . . . . .	209
7.24 serveur/Inna/tachometerinna.cpp File Reference . . . . .	209
7.24.1 Detailed Description . . . . .	209
7.25 serveur/Lea/compteur_kilometre_lea.cpp File Reference . . . . .	210
7.25.1 Detailed Description . . . . .	210
7.26 serveur/Lea/fuelgauge_lea.cpp File Reference . . . . .	210
7.26.1 Detailed Description . . . . .	211
7.27 serveur/Lea/jauge_temperature_lea.cpp File Reference . . . . .	211
7.27.1 Detailed Description . . . . .	211
7.28 serveur/Lea/lea_clignotants.cpp File Reference . . . . .	212
7.28.1 Detailed Description . . . . .	212
7.29 serveur/Lea/lea_scene.cpp File Reference . . . . .	212
7.29.1 Detailed Description . . . . .	213
7.30 serveur/Lea/speedometer_lea.cpp File Reference . . . . .	213
7.30.1 Detailed Description . . . . .	213
7.31 serveur/Lea/tachometergauge_lea.cpp File Reference . . . . .	213
7.31.1 Detailed Description . . . . .	214
7.32 serveur/Lea/voyant_lea.cpp File Reference . . . . .	214
7.32.1 Detailed Description . . . . .	214
7.33 serveur/Lea/warning_lea.cpp File Reference . . . . .	215
7.33.1 Detailed Description . . . . .	215



# Chapter 1

## Projet dashboard

Ce document a pour but de clarifier l'ensemble des opérations nécessaires afin d'intégrer au programme un dashboard ou une fonctionnalité, de lister l'ensemble des fonctionnalités/objets disponibles, et de répertorier les éventuels bugs apparus lors de l'utilisation du programme.

### Intégrer son projet au serveur

Cette section explique en détails la manière d'intégrer son dashboard au serveur. Le but premier du serveur est de permettre un changement dynamique de dashboard à la suite d'un simple message du client. Il est également important d'harmoniser le nommage des différents objets ainsi que de leur fonction, et ce afin que le programme soit fonctionnel pour l'ensemble des dashboards. Ainsi, il est nécessaire de suivre les étapes suivantes avant d'intégrer son projet au serveur.

- Une classe `scene_globale`, dérivant de `QGraphicsScene` a été créée. Elle permet de pouvoir passer dynamiquement d'un dashboard à un autre. Votre classe scène, que vous pouvez nommer `prénom_scene` et qui contient l'ensemble de vos éléments graphiques, doit donc elle même hériter de `scene_globale`.
- De même, une classe `objet_virtuel` contenant 3 méthodes (`setValue`, `setValueMax`, `getValue`) a été créée. Toutes vos classes compteur, voyant, jauge, etc... doivent hériter de cette classe, afin de rendre tous les objets fonctionnels.
- Une fois l'ensemble de ces opérations effectuées, votre projet devrait être en mesure d'intégrer le serveur. On objet `scene_globale *dashboard` a déjà été intégré à "mainwindow.h". Il vous suffit donc de rajouter un `#include "prénom_scene"` dans le "mainwindow.h". Ensuite, dans la fonction `MainWindow::reception()` de "mainwindow.cpp", vous pouvez définir votre dashboard dans la partie correspondant au message "CANN DASHBOARD":
  - `delete dashboard;` → permet de supprimer le dashboard défini au préalable.
  - `dashboard = new prénom_scene;` → définit l'objet dashboard avec votre classe dérivant de `scene_globale`.
  - `ui->graphicsView->setScene(dashboard);` → attribut votre scène au `QgraphicsView`.

Une fois votre scène intégrée au serveur, il est alors possible d'y intégrer de nouveaux objets et fonctionnalités. Vous pouvez tester la réussite de l'intégration de votre dashboard en envoyant le message CANN DASHBOARD prénom à partir du client.

## Intégration d'un nouvel objet au client et au serveur

L'ensemble des objets accessibles pour tous les dashboards sont définis en tant que `objet_virtuel` dans "scene↔\_globale.h". Ceci implique qu'un dashboard ne doit pas obligatoirement posséder in situ l'ensemble des objets et fonctionnalités définis dans "scene\_globale.h". Il est en revanche bien évidemment déconseillé de lancer à partir du client des messages CANN concernant des fonctionnalités non incluses dans votre dashboard. Suivez ces différentes étapes pour ajouter un objet au programme:

- Dans un premier temps, il faut créer votre message CANN de la forme "CANN votre\_message " dans le client. Dans le fichier clientio.c, ajouter "votre\_message" à la liste des commandes déjà disponibles de la fonction `validate_message`. Puis dans le fichier main.c du client, ajoutez votre commande "CANN votre\_message" ainsi qu'une brève description de votre message dans la commande "HELP" de la fonction `main`.
- Puis, dans le fichier mainwindow.cpp du serveur, ajoutez votre message sous forme de condition dans la liste déjà présente de la fonction `reception`. N'hésitez pas à rajouter des conditions pour vérifier si les valeurs associées à votre message sont acceptables.

## Liste des messages CANN

Objet	Description	Message CANN	Valeurs
Vitesse	Compteur vitesse	CANN SPEED X	X=vitesse
CompteTours	Compteur tour/min moteur	CANN RPM X	X=rpm
VoyantBatterie	Voyant indiquant changement de batterie	CANN BATTERY_LIGHT	0 éteint, 1 allumé
Essence	Jauge essence	CANN GAZ X	X=d'essence restant
position	Feux de position	CANN LIGHT X	0=éteint, 1=allumé
croisement	Feux de croisement	CANN LIGHT X	0=éteint, 2=allumé
route	Feux de route	CANN LIGHT X	0=éteint, 3=allumé
Clignotant	Allume les clignotants gauche, droit ou les deux ensemble (feux d'avertissement)	CANN TURN X	1=clignotant droit, -1=clignotant gauche, 2 clignotant les deux 0=éteint
Warning	Allume le warning et met la valeur des clignotants à 2	CANN WARNING X	0=éteint, 1=allumé
AdaptiveSuspension↔ Dampers	Voyant indiquant que ASD est utilisé	CANN ASD X	0 éteint, 1 allumé
AutomaticTransmission↔ Mode	Indique le mode de transmission actuellement utilisé	CANN MODE X	1=P, 2=R, 3=N, 4=D
FrontAntifog	feux de brouillard avants	CANN FRONT_FOG X	0=éteint, 1=allumé
RearAntifog	feux de brouillard arrières	CANN REAR_FOG X	0=éteint, 1=allumé
SeatBelt	Ceinture de sécurité	CANN SEAT_BELT X	0=éteint, 1=allumé
RearWindowHeating	Chauffage de la glace arrière	CANN RW_HEAT X	0=éteint, 1=allumé
CheckEngine	Voyant d'anomalie du moteur	CANN CHECK_ENGINE X	0=éteint, 1=allumé
OpenDoorDriver	Porte avant conducteur ouverte	CANN OPEN_DOOR_DRIVER X	0=éteint, 1=allumé
OpenDoorFront↔ Passenger	Porte avant passager ouverte	CANN OPEN_DOOR_↔ FRONT_PASSENGER X	0=éteint, 1=allumé



Objet	Description	Message CANN	Valeurs
OpenDoorBackRight↔ Passenger	Porte arrière droite ouverte	CANN OPEN_DOOR_↔ BACK_R_PASSENGER X	0=éteint, 1=allumé
OpenDoorBackLeft↔ Passenger	Porte arrière gauche ouverte	CANN OPEN_DOOR_↔ BACK_L_PASSENGER X	0=éteint, 1=allumé
AdaptiveCruiseControl	Voyant qui s'allume quand "AdaptiveCruiseControl" est activé	CANN CRUISE_CONTROL X	0=éteint, 1=allumé
AirbagOn	Voyant qui s'allume quand "AirBag" est activé	CANN AIRBAG_ON X	0=éteint, 1=allumé
BonnetOpen	Voyant qui s'allume quand le capot est ouvert	CANN BONNET_OPEN X	0=éteint, 1=allumé
BootOpen	Voyant qui s'allume quand le coffre est ouvert	CANN BOOT_OPEN X	0=éteint, 1=allumé
CruiseControlOn	Voyant qui s'allume quand le limiteur de vitesse est activé	CANN CRUISE_CON↔ TROL_ON X	0=éteint, 1=allumé
OilTemp	Température de l'huile dans le moteur	CANN OIL_T X	x entre 0 et value_max
oilLevel	Niveau de l'huile dans le moteur	CANN OIL_L X	x entre 0 et value_max
jaugTemperature	la température du moteur	CANN ENGINE_T X	x jusqu'au value_max
SpeedLimit	Affichage de la limite de vitesse	CANN SPEED_LIMIT X	X=Limite de vitesse
ABS	ABS	CANN ABS X	0=éteint, 1=allumé
handbrake	Frein à main enclenché	CANN HANDBRAKE X	0=éteint, 1=allumé

## Bugs répertoriés



## Chapter 2

# Bug List

File [hugo\\_compteur.cpp](#)

L'utilisation du paramètre critique semble être soumis à certains bugs.



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

QGraphicsItem	
aiguille_Lea . . . . .	21
cadrantcontour . . . . .	33
cadrantessence . . . . .	34
cadranthuile . . . . .	39
cadrantVirtuel . . . . .	43
CarDialerPrototype . . . . .	47
EngineHeatGauge . . . . .	56
FuelEngineInna . . . . .	67
FuelGauge . . . . .	69
InfoWindowInna . . . . .	103
jaugeVirtuel . . . . .	113
lea_SceneDeFond . . . . .	133
mygraphicItem . . . . .	150
objet_virtuel . . . . .	151
afficheKmHenri . . . . .	19
ArrowOilInna . . . . .	22
ArrowOilTInna . . . . .	24
ArrowSpeedometerInna . . . . .	26
ArrowTachometerInna . . . . .	28
cadran . . . . .	30
CadranEss . . . . .	32
CadrantFlorian . . . . .	35
cadrantHenri . . . . .	38
CadranTourParMin . . . . .	40
cadranttour . . . . .	42
cadrantvitesse . . . . .	44
CadranVitesse . . . . .	46
CarSpeedDial . . . . .	48
clignot . . . . .	49
Compteur_kilometre_lea . . . . .	52
compteurKmInna . . . . .	55
EngineTInna . . . . .	57
FlorianClignotant . . . . .	59
FlorianCompteurKm . . . . .	61

FlorianVoyant	64
fuel_guage	66
FuelGauge_lea	70
FuelLevelInna	72
Harc	74
horloge_lea	79
hugo_boite_vitesse	81
hugo_Compteur	83
hugo_ecran	86
hugo_MyGraphicsitem	88
hugo_voyant_warning	92
hugo_voyants_clignotant	94
hugo_voyants_simples	96
IconInna	98
clignotantInna	50
iconOnOff	100
iconOnOffInna	101
PRND_Inna	159
jauge_temperature_lea	106
jaugeClignotantHenri	109
jaugeEssenceHenri	110
jaugeTemperatureHenri	111
jaugeToursMinuteHenri	112
Jonas_compteur	114
jonas_compteurKm	118
jonas_voyant_clignotant	122
jonas_voyant_simple	124
jonas_voyant_warning	126
lea_clignottants	130
LEDS	134
Leo_display	135
Leo_indicator	138
Leo_object	140
Leo_gauge	136
loto_dashboard	143
onoff	155
onoffpaintHenri	156
portesHenri	157
speedometer_Lea	168
TachometerGauge_Lea	174
tachometre	178
tempGauge	179
voyant_Lea	180
Voyants	182
voyants_warning	184
VoyantsMaroua	185
voyantsYoucef	186
warning_lea	187
warninghenri	188
OilGaugeInna	153
sceneDeFond	163
sceneDeFondHenri	164
SpeedometerInna	171
Tachometer	173
TachometerInna	176
QGraphicsScene	
scene_globale	161
harout_scene	76

henri_scene . . . . .	78
hugo_scene . . . . .	90
inna_scene . . . . .	105
Jonas_scene . . . . .	120
karim_scene . . . . .	128
Lea_scene . . . . .	132
Leo_scene . . . . .	141
loto_scene . . . . .	144
maroua_scene . . . . .	148
SceneFlorian . . . . .	165
Youcef_Scene . . . . .	189
sceneGlobale . . . . .	167
QGraphicsSimpleTextItem	
Leo_label . . . . .	139
QMainWindow	
MainWindow . . . . .	146
MainWindow . . . . .	146
MainWindow . . . . .	146
MainWindow . . . . .	146





## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">afficheKmHenri</a>	19
<a href="#">aiguille_Lea</a>	21
<a href="#">ArrowOilInna</a>	22
<a href="#">ArrowOilTInna</a>	24
<a href="#">ArrowSpeedometerInna</a>	26
<a href="#">ArrowTachometerInna</a>	28
<a href="#">cadran</a>	30
<a href="#">CadranEss</a>	32
<a href="#">cadrantcontour</a>	33
<a href="#">cadrantessence</a>	34
<a href="#">CadrantFlorian</a>	35
<a href="#">cadrantHenri</a>	38
<a href="#">cadranhuile</a>	39
<a href="#">CadranTourParMin</a>	40
<a href="#">cadranttour</a>	42
<a href="#">cadrantVirtuel</a>	43
<a href="#">cadrantvitesse</a>	44
<a href="#">CadranVitesse</a>	46
<a href="#">CarDialerPrototype</a>	47
<a href="#">CarSpeedDial</a>	48
<a href="#">clignot</a>	49
<a href="#">clignotantInna</a>	50
<a href="#">Compteur_kilometre_lea</a>	52
<a href="#">compteurKmlInna</a>	55
<a href="#">EngineHeatGauge</a>	56
<a href="#">EngineTInna</a>	57
<a href="#">FlorianClignotant</a>	59
<a href="#">FlorianCompteurKm</a>	61
<a href="#">FlorianVoyant</a>	64
<a href="#">fuel_guage</a>	66
<a href="#">FuelEngineInna</a>	67
<a href="#">FuelGauge</a>	69
<a href="#">FuelGauge_lea</a>	70
<a href="#">FuelLevelInna</a>	72
<a href="#">Harc</a>	74

harout_scene	76
henri_scene	78
horloge_lea	79
hugo_boite_vitesse	81
hugo_Compteur	83
hugo_ecran	86
hugo_MyGraphicsItem	88
hugo_scene	90
hugo_voyant_warning	92
hugo_voyants_clignotant	94
hugo_voyants_simples	96
IconInna	98
iconOnOff	100
iconOnOffInna	101
InfoWindowInna	103
inna_scene	105
jauge_temperature_lea	106
jaugeClignotantHenri	109
jaugeEssenceHenri	110
jaugeTemperatureHenri	111
jaugeToursMinuteHenri	112
jaugeVirtuel	113
Jonas_compteur	114
jonas_compteurKm	118
Jonas_scene	120
jonas_voyant_clignotant	122
jonas_voyant_simple	124
jonas_voyant_warning	126
karim_scene	128
lea_clignottants	130
Lea_scene	132
lea_SceneDeFond	133
LEDS	134
Leo_display	135
Leo_gauge	136
Leo_indicator	138
Leo_label	139
Leo_object	140
Leo_scene	141
loto_dashboard	143
loto_scene	144
MainWindow	146
maroua_scene	148
mygraphicItem	150
objet_virtuel	151
OilGaugeInna	153
onoff	155
onoffpaintHenri	156
portesHenri	157
PRND_Inna	159
scene_globale	161
sceneDeFond	163
sceneDeFondHenri	164
SceneFlorian	165
sceneGlobale	167
speedometer_Lea	168
SpeedometerInna	171
Tachometer	173

TachometerGauge_Lea . . . . .	174
TachometerInna . . . . .	176
tachometre . . . . .	178
tempGauge . . . . .	179
voyant_Lea . . . . .	180
Voyants . . . . .	182
voyants_warning . . . . .	184
VoyantsMaroua . . . . .	185
voyantsYucef . . . . .	186
warning_lea . . . . .	187
warninghenri . . . . .	188
Yucef_Scene . . . . .	189



## Chapter 5

# File Index

### 5.1 File List

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

serveur/mainwindow.h	??
serveur/objet_virtuel.h	??
serveur/scene_globale.h	??
serveur/Florian/cadrantflorian.h	??
serveur/Florian/cadrantvirtuel.h	??
serveur/Florian/florianclignotant.h	??
serveur/Florian/floriancompteurkm.h	??
serveur/Florian/florianvoyant.h	??
serveur/Florian/jaugevirtuel.h	??
serveur/Florian/mainwindow.h	??
serveur/Florian/scenedefond.h	??
serveur/Florian/sceneflorian.h	??
serveur/Florian/sceneglobale.h	??
serveur/Harout/cadran.h	??
serveur/Harout/harc.h	??
serveur/Harout/harout_scene.h	??
serveur/Henri/affichekmhenri.cpp	
Classe dérivant de <a href="#">objet_virtuel</a> permettant l'affichage du nombre de km parcourut	191
serveur/Henri/affichekmhenri.h	??
serveur/Henri/cadranthenri.h	??
serveur/Henri/henri_scene.h	??
serveur/Henri/jaugeclignotanthenri.h	??
serveur/Henri/jaugeessencehenri.h	??
serveur/Henri/jaugetemperaturehenri.h	??
serveur/Henri/jaugetoursminutehenri.h	??
serveur/Henri/mainwindow.h	??
serveur/Henri/onoff.h	??
serveur/Henri/onoffpainthenri.h	??
serveur/Henri/porteshenri.h	??
serveur/Henri/scenedefondhenri.h	??
serveur/Henri/warninghenri.h	??
serveur/Hugo/hugo_boite_vitesse.cpp	
Classe dérivant de <a href="#">objet_virtuel</a> permettant l'affichage du mode de transmission automatique du véhicule	192
serveur/Hugo/hugo_boite_vitesse.h	??

serveur/Hugo/hugo_compteur.cpp	
Classe compteur, permettant la création de compteurs paramétrables avec aiguille et effet de verre	192
serveur/Hugo/hugo_compteur.h	??
serveur/Hugo/hugo_ecran.cpp	
Classe dérivant de <a href="#">objet_virtuel</a> permettant un affichage en temps réel du temps ainsi que de la distance parcourue depuis le redémarrage du programme	193
serveur/Hugo/hugo_ecran.h	??
serveur/Hugo/hugo_mygraphicsitem.h	??
serveur/Hugo/hugo_scene.cpp	
Classe dérivée de <a href="#">scene_globale</a> où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans <a href="#">scene_globale.h</a>	194
serveur/Hugo/hugo_scene.h	??
serveur/Hugo/hugo_voyant_warning.cpp	
Classe dérivée de <a href="#">objet_virtuel</a> permettant l'affichage d'un voyant warning clignotant	194
serveur/Hugo/hugo_voyant_warning.h	??
serveur/Hugo/hugo_voyants_clignotant.h	??
serveur/Hugo/hugo_voyants_simples.cpp	
Classe permettant l'affichage de voyants simples i.e. dont l'état est allumé ou éteint	195
serveur/Hugo/hugo_voyants_simples.h	??
serveur/Inna/arrowoillinna.cpp	
Classe <a href="#">ArrowOilInna</a> dérivée de la classe <a href="#">objet_virtuel</a>	196
serveur/Inna/arrowoillinna.h	??
serveur/Inna/arrowoiltinna.cpp	
Classe <a href="#">ArrowOilTinna</a> dérivée de la classe <a href="#">objet_virtuel</a>	197
serveur/Inna/arrowoiltinna.h	??
serveur/Inna/arrowspeedometerinna.cpp	
Classe <a href="#">ArrowSpeedometerInna</a> dérivée de la classe <a href="#">objet_virtuel</a>	197
serveur/Inna/arrowspeedometerinna.h	??
serveur/Inna/arrowtachometerinna.cpp	
Classe <a href="#">ArrowTachometerInna</a> dérivée de la classe <a href="#">objet_virtuel</a>	198
serveur/Inna/arrowtachometerinna.h	??
serveur/Inna/clignotantinna.cpp	
Classe permettant d'afficher les voyants clignotants	199
serveur/Inna/clignotantinna.h	??
serveur/Inna/compteurkminna.cpp	
Classe dérivant de <a href="#">objet_virtuel</a> permettant un affichage en temps réel de la distance parcourue depuis le redémarrage du programme	200
serveur/Inna/compteurkminna.h	??
serveur/Inna/enginetinna.cpp	
Classe <a href="#">EngineTinna</a> dérivée de la classe <a href="#">objet_virtuel</a>	201
serveur/Inna/enginetinna.h	??
serveur/Inna/fuelengineinna.cpp	
Classe dérivée de <a href="#">QGraphicsItem</a> permettant l'affichage des jauges de l'essence et de la température du moteur	202
serveur/Inna/fuelengineinna.h	??
serveur/Inna/fuellevelinna.cpp	
Classe <a href="#">FuelLevelInna</a> dérivée de la classe <a href="#">objet_virtuel</a>	202
serveur/Inna/fuellevelinna.h	??
serveur/Inna/iconinna.cpp	
Classe <a href="#">IconInna</a> dérivée de la classe <a href="#">objet_virtuel</a>	203
serveur/Inna/iconinna.h	??
serveur/Inna/icononoff.h	??
serveur/Inna/icononoffinna.cpp	
Classe <a href="#">iconOnOffInna</a> dérivée de la classe <a href="#">IconInna</a>	204
serveur/Inna/icononoffinna.h	??
serveur/Inna/infowindowinna.cpp	
Classe <a href="#">InfoWindowInna</a> dérivée de la classe <a href="#">QGraphicsItem</a>	205

serveur/Inna/ <b>infowindowinna.h</b> . . . . .	??
serveur/Inna/ <b>inna_scene.cpp</b>	
Classe dérivée de <a href="#">scene_globale</a> où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans <a href="#">scene_globale.h</a> . . . . .	206
serveur/Inna/ <b>inna_scene.h</b> . . . . .	??
serveur/Inna/ <b>oilgaugeinna.cpp</b>	
Classe dérivée de QGraphicsItem permettant l'affichage des jauges de niveau et de température d'huile . . . . .	206
serveur/Inna/ <b>oilgaugeinna.h</b> . . . . .	??
serveur/Inna/ <b>prnd_inna.cpp</b>	
Classe <a href="#">PRND_Inna</a> dérivée de la classe <a href="#">IconInna</a> pour afficher les modes de la transmission . . . . .	207
serveur/Inna/ <b>prnd_inna.h</b> . . . . .	??
serveur/Inna/ <b>speedometerinna.cpp</b>	
Classe <a href="#">SpeedometerInna</a> dérivée de la classe QGraphicsItem . . . . .	208
serveur/Inna/ <b>speedometerinna.h</b> . . . . .	??
serveur/Inna/ <b>tachometerinna.cpp</b>	
Classe <a href="#">TachometerInna</a> dérivée de la classe QGraphicsItem . . . . .	209
serveur/Inna/ <b>tachometerinna.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_compteur.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_compteurkm.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_scene.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_voyant_clignotant.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_voyant_simple.h</b> . . . . .	??
serveur/Jonas/ <b>jonas_voyant_warning.h</b> . . . . .	??
serveur/Karim/ <b>cadranEss.h</b> . . . . .	??
serveur/Karim/ <b>cadrantourparmin.h</b> . . . . .	??
serveur/Karim/ <b>CadranVitesse.h</b> . . . . .	??
serveur/Karim/ <b>clignot.h</b> . . . . .	??
serveur/Karim/ <b>karim_scene.h</b> . . . . .	??
serveur/Karim/ <b>voyants.h</b> . . . . .	??
serveur/Karim/ <b>voyants_warning.h</b> . . . . .	??
serveur/Lea/ <b>aiguille_lea.h</b> . . . . .	??
serveur/Lea/ <b>compteur_kilometre_lea.cpp</b>	
Classe dérivant de <a href="#">objet_virtuel</a> permettant un affichage en temps réel de la distance parcourue depuis le démarrage du programme . . . . .	210
serveur/Lea/ <b>compteur_kilometre_lea.h</b> . . . . .	??
serveur/Lea/ <b>fuelgauge_lea.cpp</b>	
Classe jauge d'essence, permettant la création d'une jauge paramétrables . . . . .	210
serveur/Lea/ <b>fuelgauge_lea.h</b> . . . . .	??
serveur/Lea/ <b>horloge_lea.h</b> . . . . .	??
serveur/Lea/ <b>jauge_temperature_lea.cpp</b>	
Classe jauge de température, permettant la création d'une jauge paramétrable . . . . .	211
serveur/Lea/ <b>jauge_temperature_lea.h</b> . . . . .	??
serveur/Lea/ <b>lea_clignotants.cpp</b>	
Classe dérivée de <a href="#">objet_virtuel</a> permettant l'affichage d'un voyant clignotant droite ou gauche . . . . .	212
serveur/Lea/ <b>lea_clignotants.h</b> . . . . .	??
serveur/Lea/ <b>lea_scene.cpp</b>	
Classe dérivée de <a href="#">scene_globale</a> où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans <a href="#">scene_globale.h</a> . . . . .	212
serveur/Lea/ <b>lea_scene.h</b> . . . . .	??
serveur/Lea/ <b>lea_scenedefond.h</b> . . . . .	??
serveur/Lea/ <b>mainwindow.h</b> . . . . .	??
serveur/Lea/ <b>mygraphicitem.h</b> . . . . .	??
serveur/Lea/ <b>speedometer_lea.cpp</b>	
Classe speedometer, permettant la création du compteur vitesse paramétrable avec aiguille . . . . .	213
serveur/Lea/ <b>speedometer_lea.h</b> . . . . .	??
serveur/Lea/ <b>tachometergauge_lea.cpp</b>	
Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille . . . . .	213

serveur/Lea/tachometergauge_lea.h . . . . .	??
serveur/Lea/voyant_lea.cpp	
Classe permettant l'affichage de voyants dont l'état est allumé ou éteint . . . . .	214
serveur/Lea/voyant_lea.h . . . . .	??
serveur/Lea/warning_lea.cpp	
Classe dérivée de <a href="#">objet_virtuel</a> permettant l'affichage d'un voyant warning clignotant . . . . .	215
serveur/Lea/warning_lea.h . . . . .	??
serveur/Leo/leo_display.h . . . . .	??
serveur/Leo/leo_gauge.h . . . . .	??
serveur/Leo/leo_indicator.h . . . . .	??
serveur/Leo/leo_label.h . . . . .	??
serveur/Leo/leo_object.h . . . . .	??
serveur/Leo/leo_scene.h . . . . .	??
serveur/Loto/fuel_guage.h . . . . .	??
serveur/Loto/leds.h . . . . .	??
serveur/Loto/loto_dashboard.h . . . . .	??
serveur/Loto/loto_scene.h . . . . .	??
serveur/Loto/tachometre.h . . . . .	??
serveur/Loto/tempgauge.h . . . . .	??
serveur/Maroua/cadrantcontour.h . . . . .	??
serveur/Maroua/cadrantessence.h . . . . .	??
serveur/Maroua/cadranttempmoteur.h . . . . .	??
serveur/Maroua/cadranttour.h . . . . .	??
serveur/Maroua/cadrantvitesse.h . . . . .	??
serveur/Maroua/maroua_scene.h . . . . .	??
serveur/Maroua/voyantsmaroua.h . . . . .	??
serveur/Youcef/cardialerprototype.h . . . . .	??
serveur/Youcef/carspeeddial.h . . . . .	??
serveur/Youcef/engineheatgauge.h . . . . .	??
serveur/Youcef/fuelgauge.h . . . . .	??
serveur/Youcef/tachometer.h . . . . .	??
serveur/Youcef/voyantsyoucef.h . . . . .	??
serveur/Youcef/youcef_scene.h . . . . .	??

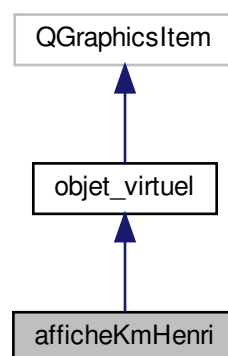


## Chapter 6

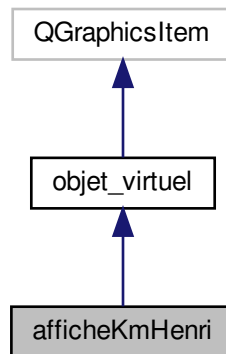
# Class Documentation

### 6.1 afficheKmHenri Class Reference

Inheritance diagram for afficheKmHenri:



Collaboration diagram for afficheKmHenri:



## Public Member Functions

- [afficheKmHenri](#) ([objet\\_virtuel](#) \*parent=nullptr)  
[afficheKmHenri::afficheKmHenri](#)
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`  
[afficheKmHenri::paint](#)

## Additional Inherited Members

### 6.1.1 Constructor & Destructor Documentation

#### 6.1.1.1 afficheKmHenri()

```
afficheKmHenri::afficheKmHenri (
    objet\_virtuel * parent = nullptr )
```

[afficheKmHenri::afficheKmHenri](#)

Initialise la variable value de la classe mère à 0

### 6.1.2 Member Function Documentation

#### 6.1.2.1 paint()

```
void afficheKmHenri::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[afficheKmHenri::paint](#)

## Parameters

<i>painter</i>	
----------------	--

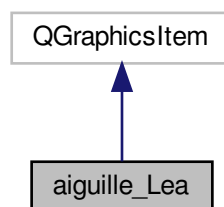
La fonction paint va permettre l'affichage des kilometres totaux et actuels

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

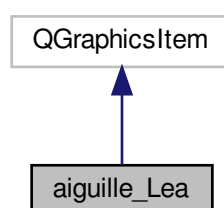
- serveur/Henri/affichekmhenri.h
- serveur/Henri/[affichekmhenri.cpp](#)

## 6.2 aiguille\_Lea Class Reference

Inheritance diagram for aiguille\_Lea:



Collaboration diagram for aiguille\_Lea:



### Public Member Functions

- **aiguille\_Lea** (`QGraphicsItem *parent=nullptr`)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)
- int **getValueMax** ()
- void **setValue** (int)
- virtual void **parametrage** (double param\_x, double param\_y, double param\_r, double param\_t, int param\_↔start, int param\_end, int param\_value)

## Public Attributes

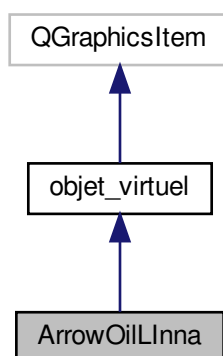
- double **x**
- double **y**
- double **r**
- double **taille\_aiguille**
- int **vitesse**
- int **vitesse\_max**
- int **angle\_depart** =0
- int **angle\_fin** =0
- int **angle** =0
- const double **pi** =3.14159265359

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

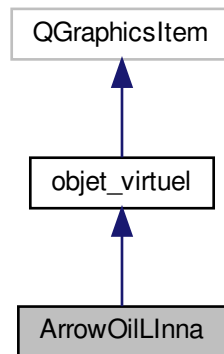
- serveur/Lea/aiguille\_lea.h
- serveur/Lea/aiguille\_lea.cpp

## 6.3 ArrowOilInna Class Reference

Inheritance diagram for ArrowOilInna:



Collaboration diagram for ArrowOilInna:



### Public Member Functions

- **ArrowOilInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **ArrowOilInna** (int LMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **lmax**
- double **l**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- double **k**
- int **dx**
- int **dy**

### Additional Inherited Members

#### 6.3.1 Member Function Documentation

### 6.3.1.1 paint()

```
void ArrowOilInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Dessin du circle central.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

Dessin de la flèche.

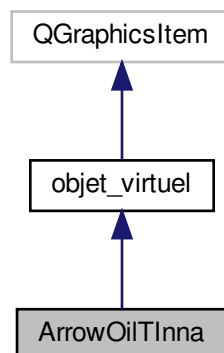
On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

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

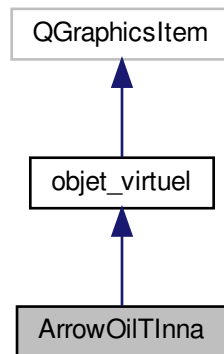
- serveur/lnna/arwoillinna.h
- serveur/lnna/[arwoillinna.cpp](#)

## 6.4 ArrowOilTlnna Class Reference

Inheritance diagram for ArrowOilTlnna:



Collaboration diagram for ArrowOilTInna:



### Public Member Functions

- **ArrowOilTInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **ArrowOilTInna** (int TMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **tmax**
- int **t**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- double **k**
- int **dx**
- int **dy**

### Additional Inherited Members

#### 6.4.1 Member Function Documentation

#### 6.4.1.1 paint()

```
void ArrowOilTinna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Dessin du circle central.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

Dessin de la flèche.

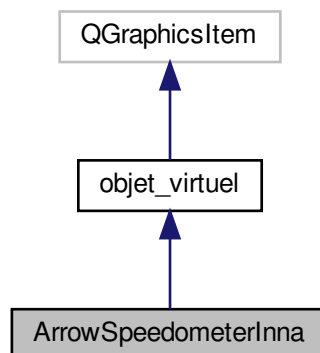
On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

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

- serveur/lnna/arrowoiltinna.h
- serveur/lnna/[arrowoiltinna.cpp](#)

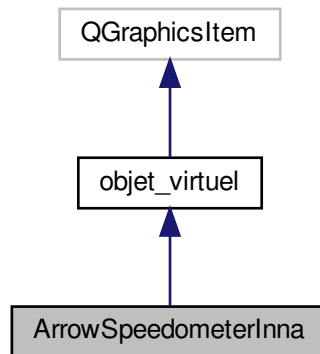
## 6.5 ArrowSpeedometerInna Class Reference

Inheritance diagram for ArrowSpeedometerInna:





Collaboration diagram for ArrowSpeedometerInna:



### Public Member Functions

- **ArrowSpeedometerInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **ArrowSpeedometerInna** (int VMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **vmax**
- int **v**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- double **k**
- int **dx**
- int **dy**

### Additional Inherited Members

#### 6.5.1 Member Function Documentation

### 6.5.1.1 paint()

```
void ArrowSpeedometerInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Dessin du cercle central.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

dessin de la flèche.

On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

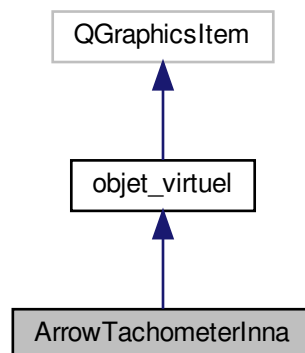
Affichage de la vitesse actuelle

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

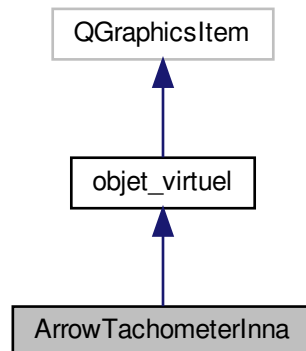
- serveur/lnna/arrowspeedometerinna.h
- serveur/lnna/[arrowspeedometerinna.cpp](#)

## 6.6 ArrowTachometerInna Class Reference

Inheritance diagram for ArrowTachometerInna:



Collaboration diagram for ArrowTachometerInna:



### Public Member Functions

- **ArrowTachometerInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **ArrowTachometerInna** (int VMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **vmax**
- int **v**
- QFont **font** = QFont("Chandas",22,QFont::ExtraBold)
- double **k**
- int **dx**
- int **dy**

### Additional Inherited Members

#### 6.6.1 Member Function Documentation

### 6.6.1.1 paint()

```
void ArrowTachometerInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Dessin du circle central.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

dessin de la flèche.

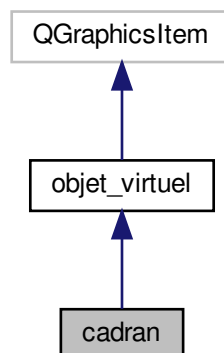
On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

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

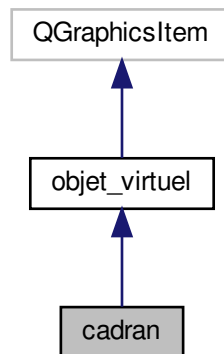
- serveur/lnna/arrowtachometerinna.h
- serveur/lnna/[arrowtachometerinna.cpp](#)

## 6.7 cadran Class Reference

Inheritance diagram for cadran:



Collaboration diagram for cadran:



### Public Member Functions

- **cadran** (QStringList, QString, QString, float, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- void **ellipse** (QPainter \*painter, QRectF)
- void **aiguille** (QPainter \*painter, QRectF)
- void **arc** (QPainter \*painter, QRectF)
- void **graduation** (QPainter \*painter, QRectF)

### Protected Attributes

- QStringList **gradVitesse**
- QStringList **gradRPM**
- QStringList **grad**
- QString **text**
- QString **Vtext**
- QString **RPMtext**
- QString **text1**
- QString **Temptext1**
- QString **Essencetext1**
- float **valueVitesse**
- float **valueRPM**
- float **ANGLE** =0
- float **angled**
- int **spanAngle0**

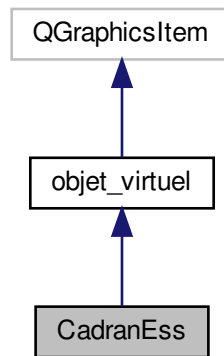
### Additional Inherited Members

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

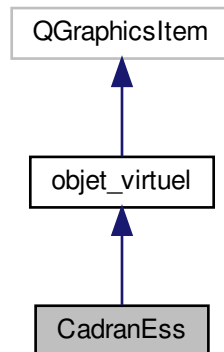
- serveur/Harout/cadran.h
- serveur/Harout/cadran.cpp

## 6.8 CadranEss Class Reference

Inheritance diagram for CadranEss:



Collaboration diagram for CadranEss:



### Public Member Functions

- **CadranEss** (QGraphicsItem \*parent=0)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

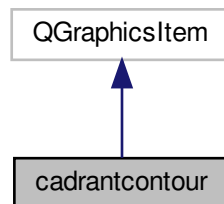
### Additional Inherited Members

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

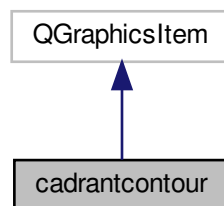
- serveur/Karim/cadranEss.h
- serveur/Karim/cadranEss.cpp

## 6.9 cadrantcontour Class Reference

Inheritance diagram for cadrantcontour:



Collaboration diagram for cadrantcontour:



### Public Member Functions

- **cadrantcontour** (`QGraphicsItem *parent=nullptr`)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

#### 6.9.1 Member Function Documentation

### 6.9.1.1 paint()

```
void cadrantcontour::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

la fonction antialiasing permet d'activer le mode haute définition

création du rectangle avec arrière plan dégradé en noir et bleu

création du cadrant vitesse

création des graduations de deux gauges a cotés pour le tour minute, l'essence et la température moteur

création contours arc pour la gauge vitesse

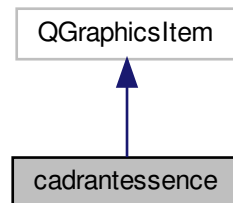
création de gradient transparent

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

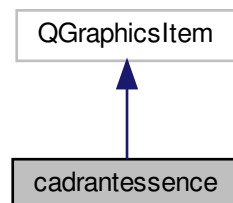
- serveur/Maroua/cadrantcontour.h
- serveur/Maroua/cadrantcontour.cpp

## 6.10 cadrantessence Class Reference

Inheritance diagram for cadrantessence:



Collaboration diagram for cadrantessence:





## Public Member Functions

- **cadrantessence** (QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

## Public Attributes

- int **niveauescence**

### 6.10.1 Member Function Documentation

#### 6.10.1.1 paint()

```
void Cadrantessence::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

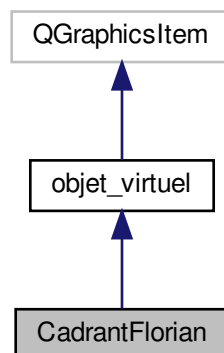
création des traits de gauge essence

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

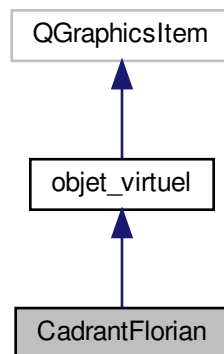
- serveur/Maroua/cadrantessence.h
- serveur/Maroua/cadrantessence.cpp

## 6.11 CadrantFlorian Class Reference

Inheritance diagram for CadrantFlorian:



Collaboration diagram for CadrantFlorian:



## Public Member Functions

- [CadrantFlorian](#) (bool hasText=true, bool hasSubTrait=true, int invertAiguille=1, int pas=10, int angleB=220, int angleE=260, int valeurMax=260, bool hasGradient=true, QGraphicsItem \*parent=NULL)  
[CadrantFlorian::CadrantFlorian.](#)
- QRectF [boundingRect](#) () const  
[CadrantFlorian::boundingRect.](#)
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[CadrantFlorian::paint.](#)
- float [generateAngle](#) ()  
[CadrantFlorian::generateAngle.](#)
- int [getSpeedMax](#) ()

## Additional Inherited Members

### 6.11.1 Constructor & Destructor Documentation

#### 6.11.1.1 CadrantFlorian()

```

CadrantFlorian::CadrantFlorian (
    bool hasText = true,
    bool hasSubTrait = true,
    int invertAiguille = 1,
    int pas = 10,
    int angleB = 220,
    int angleE = 260,
    int valeurMax = 260,
    bool hasGradient = true,
    QGraphicsItem * parent = NULL )
  
```

[CadrantFlorian::CadrantFlorian.](#)

## Parameters

<i>hasText</i>	
<i>hasSubTrait</i>	
<i>invertAiguille</i>	
<i>pas</i>	
<i>angleB</i>	
<i>angleE</i>	
<i>valeurMax</i>	
<i>parent</i>	

## 6.11.2 Member Function Documentation

### 6.11.2.1 boundingRect()

```
QRectF CadrantFlorian::boundingRect ( ) const
```

[CadrantFlorian::boundingRect.](#)

## Returns

retourne un rectangle qui encadre l'objet

### 6.11.2.2 generateAngle()

```
float CadrantFlorian::generateAngle ( )
```

[CadrantFlorian::generateAngle.](#)

## Returns

la valeur de l'angle en fonction de la valeur et de la valeur max

### 6.11.2.3 paint()

```
void CadrantFlorian::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[CadrantFlorian::paint.](#)

**Parameters**

<i>painter</i>	
----------------	--

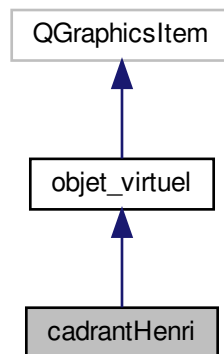
dessine les différents éléments du compteur : aiguille, compteur, chiffre...

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

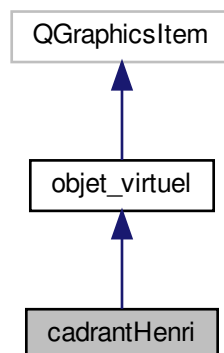
- serveur/Florian/cadrantflorian.h
- serveur/Florian/cadrantflorian.cpp

## 6.12 **cadrantHenri** Class Reference

Inheritance diagram for `cadrantHenri`:



Collaboration diagram for `cadrantHenri`:



### Public Member Functions

- **cadranthHenri** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **epaisseurTraitVitesse**
- int **tailleTexteVitesse**
- int **valeurMaxNombreCompteur**

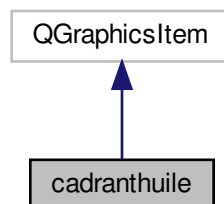
### Additional Inherited Members

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

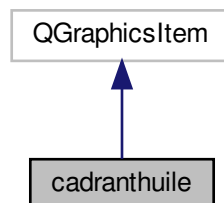
- serveur/Henri/cadranthenri.h
- serveur/Henri/cadranthenri.cpp

## 6.13 cadranthuille Class Reference

Inheritance diagram for cadranthuille:



Collaboration diagram for cadranthuille:



## Public Member Functions

- **cadranthuile** (QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### 6.13.1 Member Function Documentation

#### 6.13.1.1 paint()

```
void cadranthuile::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

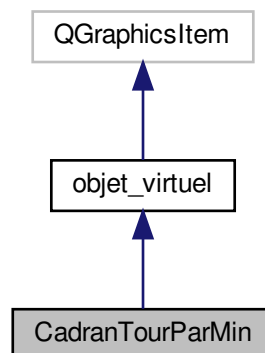
création des traits désignant la température du moteur

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

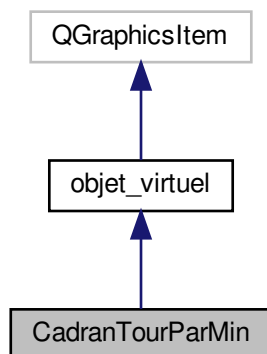
- serveur/Maroua/cadranttempmoteur.h
- serveur/Maroua/cadranttempmoteur.cpp

## 6.14 CadranTourParMin Class Reference

Inheritance diagram for CadranTourParMin:



Collaboration diagram for CadranTourParMin:



### Public Member Functions

- **CadranTourParMin** (QObject \*parent=0)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

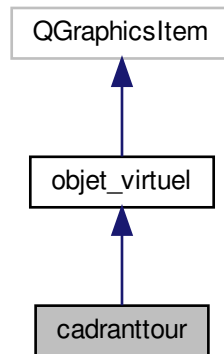
### Additional Inherited Members

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

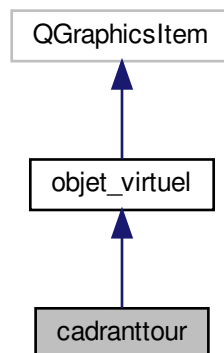
- serveur/Karim/cadrantourparmin.h
- serveur/Karim/cadrantourparmin.cpp

## 6.15 cadranttour Class Reference

Inheritance diagram for cadranttour:



Collaboration diagram for cadranttour:



### Public Member Functions

- **cadranttour** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Additional Inherited Members

#### 6.15.1 Member Function Documentation



### 6.15.1.1 paint()

```
void cadranttour::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

création des traits désignant le tour minute

création des nombres désignant les tours minute

création de l'aiguille tour minute

définition des paramètres pour dessiner l'aiguille

la création de l'aiguille

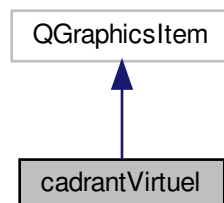
création du cercle autour de l'aiguille

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

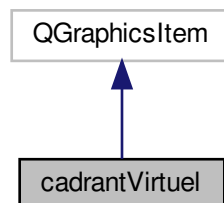
- serveur/Maroua/cadranttour.h
- serveur/Maroua/cadranttour.cpp

## 6.16 cadrantVirtuel Class Reference

Inheritance diagram for cadrantVirtuel:



Collaboration diagram for cadrantVirtuel:



## Public Member Functions

- **cadrantVirtuel** (QGraphicsItem \*parent=nullptr)
- int **getValeur** () const
- void **setValeur** (int value)
- int **getValeurMax** () const

## Protected Attributes

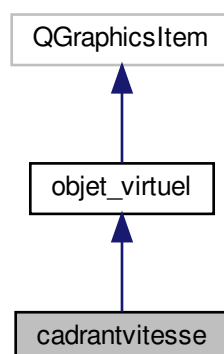
- int **valeur**
- int **valeurMax**

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

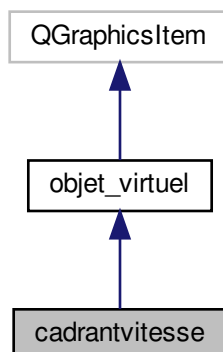
- serveur/Florian/cadrantvirtuel.h
- serveur/Florian/cadrantvirtuel.cpp

## 6.17 cadrantvitesse Class Reference

Inheritance diagram for cadrantvitesse:



Collaboration diagram for cadrantvitesse:



## Public Member Functions

- **cadrantvitesse** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

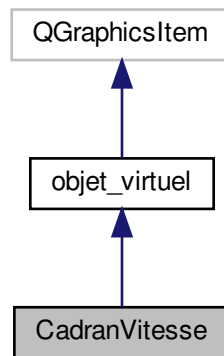
## Additional Inherited Members

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

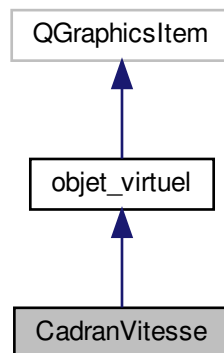
- serveur/Maroua/cadrantvitesse.h
- serveur/Maroua/cadrantvitesse.cpp

## 6.18 CadranVitesse Class Reference

Inheritance diagram for CadranVitesse:



Collaboration diagram for CadranVitesse:



### Public Member Functions

- **QRectF boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

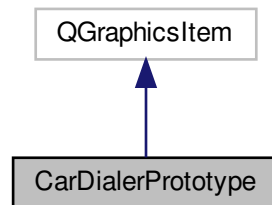
### Additional Inherited Members

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

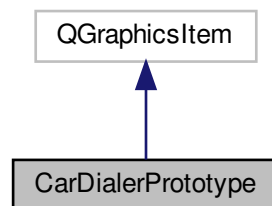
- serveur/Karim/CadranVitesse.h
- serveur/Karim/CadranVitesse.cpp

## 6.19 CarDialerPrototype Class Reference

Inheritance diagram for CarDialerPrototype:



Collaboration diagram for CarDialerPrototype:



### Public Member Functions

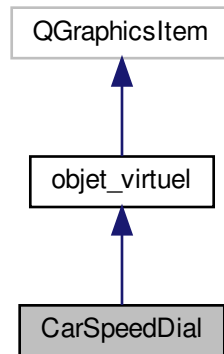
- **CarDialerPrototype** (`QGraphicsItem *parent=nullptr`)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

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

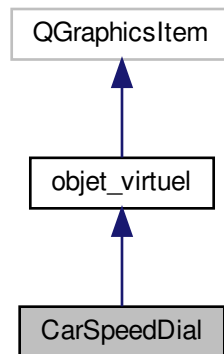
- `serveur/Youcef/cardialerprototype.h`
- `serveur/Youcef/cardialerprototype.cpp`

## 6.20 CarSpeedDial Class Reference

Inheritance diagram for CarSpeedDial:



Collaboration diagram for CarSpeedDial:



### Public Member Functions

- **CarSpeedDial** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- float **pi** = 3.14159265359

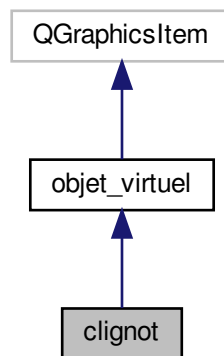
### Additional Inherited Members

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

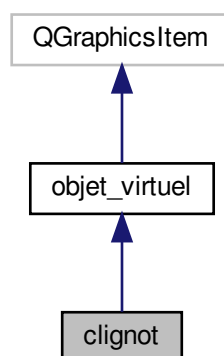
- serveur/Youcef/carspeeddial.h
- serveur/Youcef/carspeeddial.cpp

## 6.21 clignot Class Reference

Inheritance diagram for clignot:



Collaboration diagram for clignot:



## Public Member Functions

- **clignot** (QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- void **MAJ** ()

## Public Attributes

- int **cligno**

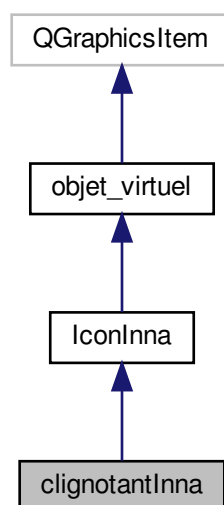
## Additional Inherited Members

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

- serveur/Karim/clignot.h
- serveur/Karim/clignot.cpp

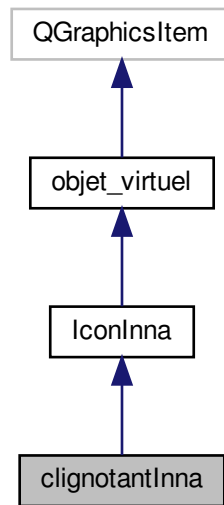
## 6.22 clignantInna Class Reference

Inheritance diagram for clignantInna:





Collaboration diagram for clignotantInna:



## Public Member Functions

- `clignotantInna (IconInna *parent=nullptr)`  
*`clignotantInna::clignotantInna` - Constructeur de la classe.*
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
- `void MAJ ()`  
*`clignotantInna::MAJ()` Fonction de mise à jour de l'affichage. La valeur cligno controle l'opacité du painter, permettant de le rendre visible ou non*

## Public Attributes

- `int cligno`
- `int mode`

## Additional Inherited Members

### 6.22.1 Constructor & Destructor Documentation

### 6.22.1.1 clignotantInna()

```
clignotantInna::clignotantInna (
    IconInna * parent = nullptr )
```

[clignotantInna::clignotantInna](#) - Constructeur de la classe.

Constructeur de la classe initialisant la variable *value* de la classe mère, et la variable *cligno* qui permet de gérer l'affichage alternatif des voyants.

## 6.22.2 Member Function Documentation

### 6.22.2.1 paint()

```
void clignotantInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

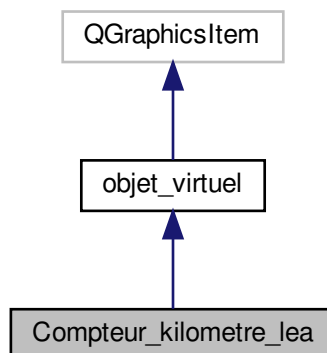
Va chercher la valeur de la variable *value* de la classe [clignotantInna](#). *value* = 1 correspond à l'affichage du clignotant droit *value* = -1 correspond à l'affichage du clignotant gauche *value* = 2 correspond à l'affichage des clignotants droit et gauche au même temps

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

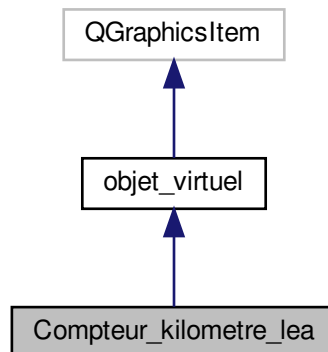
- serveur/Inna/clignotantinna.h
- serveur/Inna/[clignotantinna.cpp](#)

## 6.23 Compteur\_kilometre\_lea Class Reference

Inheritance diagram for Compteur\_kilometre\_lea:



Collaboration diagram for Compteur\_kilometre\_lea:



## Public Member Functions

- [Compteur\\_kilometre\\_lea](#) (int, int)  
*Compteur\_kilometre\_lea::Compteur\_kilometre\_lea* Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0.
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`  
*Compteur\_kilometre\_lea::paint.*

## Public Attributes

- `int x`
- `int y`

## Additional Inherited Members

### 6.23.1 Constructor & Destructor Documentation

#### 6.23.1.1 Compteur\_kilometre\_lea()

```
Compteur_kilometre_lea::Compteur_kilometre_lea (
    int param_x,
    int param_y )
```

[Compteur\\_kilometre\\_lea::Compteur\\_kilometre\\_lea](#) Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0.

## Parameters

<i>param</i> ↔ _x	position horizontale du coin en haut à gauche du rectangle dans lequel apparaîtra le compteur kilométrique.
<i>param</i> ↔ _y	position verticale du coin en haut à gauche du rectangle dans lequel apparaîtra le compteur de kilométrique.

## 6.23.2 Member Function Documentation

### 6.23.2.1 paint()

```
void Compteur_kilometre_lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[Compteur\\_kilometre\\_lea::paint.](#)

## Parameters

<i>painter</i>	
----------------	--

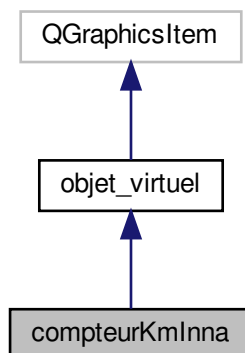
La fonction paint va permettre de réaliser un affichage pour les compteurs kilometrique totaux et du voyage commencé, de récupérer le kilometrage avec la fonction [getValue\(\)](#) et d'afficher l'heure.

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

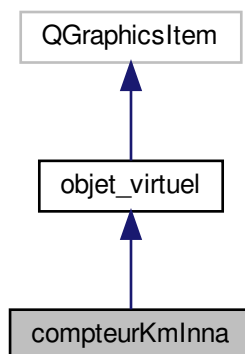
- serveur/Lea/compteur\_kilometre\_lea.h
- serveur/Lea/[compteur\\_kilometre\\_lea.cpp](#)

## 6.24 compteurKmlInna Class Reference

Inheritance diagram for compteurKmlInna:



Collaboration diagram for compteurKmlInna:



### Public Member Functions

- **compteurKmlInna** ([objet\\_virtuel](#) \*parent=nullptr)
- `QRectF` **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[compteurKmlInna::paint](#)

## Additional Inherited Members

### 6.24.1 Member Function Documentation

#### 6.24.1.1 `paint()`

```
void compteurKmInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[compteurKmInna::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

La fonction `paint` permet de réaliser un affichage du compteur kilometrique.La valeur ***value*** est envoyer par la fonction `MainWindow::update_km()` dans le fichier `mainwindow.cpp`. Affichage de la valeur du kilométrage.

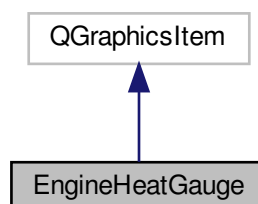
Affichage d'un rectangle avec gradient.

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

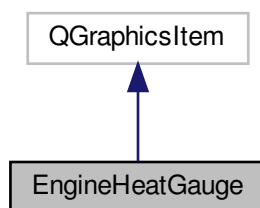
- `serveur/lnna/compteurkminna.h`
- `serveur/lnna/compteurkminna.cpp`

## 6.25 EngineHeatGauge Class Reference

Inheritance diagram for `EngineHeatGauge`:



Collaboration diagram for EngineHeatGauge:



### Public Member Functions

- **EngineHeatGauge** (`QGraphicsItem *parent=nullptr`)
- `QRectF boundingRect ()` const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

### Public Attributes

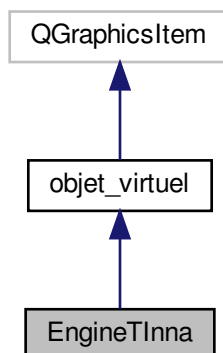
- float **pi** = 3.14159265359

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

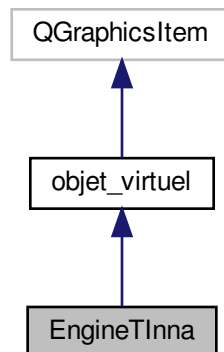
- `serveur/Youcef/engineheatgauge.h`
- `serveur/Youcef/engineheatgauge.cpp`

## 6.26 EngineTInna Class Reference

Inheritance diagram for EngineTInna:



Collaboration diagram for EngineTInna:



### Public Member Functions

- **EngineTInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **EngineTInna** (int TMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **tmax**
- int **t**
- double **k**

### Additional Inherited Members

#### 6.26.1 Member Function Documentation



### 6.26.1.1 paint()

```
void EngineTinna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Dessin du circle central.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

Dessin de la flèche

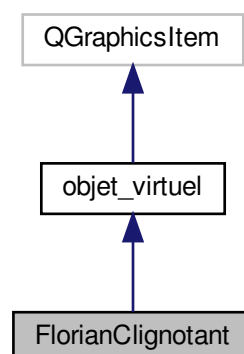
On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

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

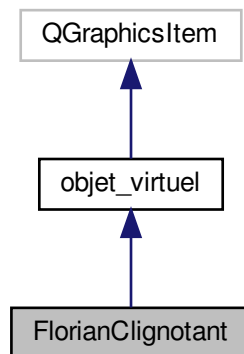
- serveur/lnna/enginetinna.h
- serveur/lnna/[enginetinna.cpp](#)

## 6.27 FlorianClignotant Class Reference

Inheritance diagram for FlorianClignotant:



Collaboration diagram for FlorianClignotant:



## Public Member Functions

- [FlorianClignotant](#) (QGraphicsItem \*parent=nullptr)  
*FlorianClignotant::FlorianClignotant.*
- QRectF [boundingRect](#) () const  
*FlorianClignotant::boundingRect.*
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- void **clignoter** ()

## Additional Inherited Members

### 6.27.1 Constructor & Destructor Documentation

#### 6.27.1.1 FlorianClignotant()

```

FlorianClignotant::FlorianClignotant (
    QGraphicsItem * parent = nullptr )

```

[FlorianClignotant::FlorianClignotant.](#)

#### Parameters

<i>parent</i>	
---------------	--

## 6.27.2 Member Function Documentation

### 6.27.2.1 boundingRect()

```
QRectF FlorianClignotant::boundingRect ( ) const
```

[FlorianClignotant::boundingRect.](#)

#### Returns

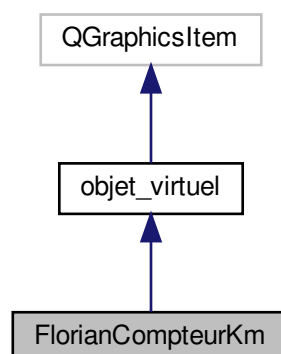
retourne un rectangle qui encadre l'objet

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

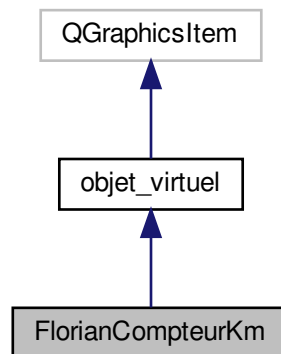
- serveur/Florian/florianclignotant.h
- serveur/Florian/florianclignotant.cpp

## 6.28 FlorianCompteurKm Class Reference

Inheritance diagram for FlorianCompteurKm:



Collaboration diagram for FlorianCompteurKm:



## Public Member Functions

- [FlorianCompteurKm](#) (QGraphicsItem \*parent=nullptr)  
*FlorianCompteurKm::FlorianCompteurKm.*
- QRectF [boundingRect](#) () const  
*FlorianCompteurKm::boundingRect.*
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*FlorianCompteurKm::paint.*

## Additional Inherited Members

### 6.28.1 Constructor & Destructor Documentation

#### 6.28.1.1 FlorianCompteurKm()

```
FlorianCompteurKm::FlorianCompteurKm (  
    QGraphicsItem * parent = nullptr )
```

[FlorianCompteurKm::FlorianCompteurKm.](#)

#### Parameters

<i>parent</i>	
---------------	--

## 6.28.2 Member Function Documentation

### 6.28.2.1 boundingRect()

```
QRectF FlorianCompteurKm::boundingRect ( ) const
```

[FlorianCompteurKm::boundingRect.](#)

#### Returns

retourne un rectangle qui encadre l'objet

### 6.28.2.2 paint()

```
void FlorianCompteurKm::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[FlorianCompteurKm::paint.](#)

#### Parameters

<i>painter</i>	
----------------	--

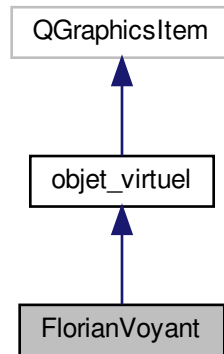
dessine le compteur km avec une police d'écriture digital

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

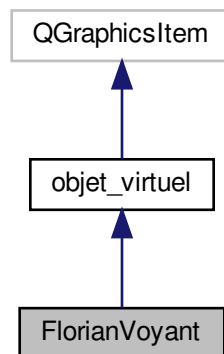
- serveur/Florian/floriancompteurkm.h
- serveur/Florian/floriancompteurkm.cpp

## 6.29 FlorianVoyant Class Reference

Inheritance diagram for FlorianVoyant:



Collaboration diagram for FlorianVoyant:



### Public Member Functions

- **FlorianVoyant** (QPixmap map, QGraphicsItem \*parent=nullptr)
- QRectF [boundingRect](#) () const  
*FlorianVoyant::boundingRect.*
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*FlorianVoyant::paint.*

## Additional Inherited Members

### 6.29.1 Member Function Documentation

#### 6.29.1.1 boundingRect()

```
QRectF FlorianVoyant::boundingRect ( ) const
```

[FlorianVoyant::boundingRect.](#)

#### Returns

retourne le rectangle qui encadre l'objet

#### 6.29.1.2 paint()

```
void FlorianVoyant::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[FlorianVoyant::paint.](#)

#### Parameters

<i>painter</i>	<input type="checkbox"/>
----------------	--------------------------

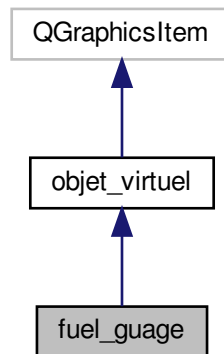
si value == 1, dessine l'image qui est en attribut

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

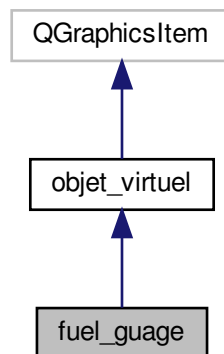
- serveur/Florian/florianvoyant.h
- serveur/Florian/florianvoyant.cpp

## 6.30 fuel\_guage Class Reference

Inheritance diagram for fuel\_guage:



Collaboration diagram for fuel\_guage:



### Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- double **getrotationAngle** (double tankLevelLitres)

### Public Attributes

- double **tankLevelLitres** =getValue()
- const double **pi** =3.142



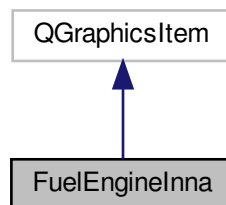
### Additional Inherited Members

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

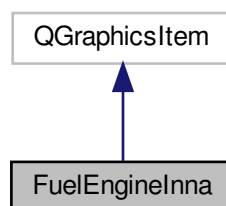
- serveur/Loto/fuel\_guage.h
- serveur/Loto/fuel\_guage.cpp

## 6.31 FuelEngineInna Class Reference

Inheritance diagram for FuelEngineInna:



Collaboration diagram for FuelEngineInna:



### Public Member Functions

- **FuelEngineInna** (`QGraphicsItem *parent=nullptr`)
- **FuelEngineInna** (`int TMAX, int ALPHA0, int ALPHAMAX`)
- `QRectF boundingRect () const`
- void [paint](#) (`QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget`)  
*[FuelEngineInna::paint.](#)*

## Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **tmax**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- int **dx**
- int **dy**
- double **k**

## 6.31.1 Member Function Documentation

### 6.31.1.1 paint()

```
void FuelEngineInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[FuelEngineInna::paint.](#)

#### Parameters

<i>painter</i>	
----------------	--

La fonction paint va permettre de réaliser un affichage des objets fixes de la représentation des jauges de l'essence et de la température du moteur. Remplissage du fond et dessin de l'arche encadrante

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une arche encadrante avec un gradient de couleur.

Affichage des traits et du texte pour la jauge de la température du moteur.

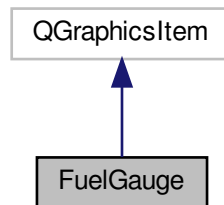
Affichage des traits et du texte pour le niveau d'essence.

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

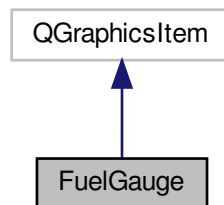
- serveur/lnna/fuelengineinna.h
- serveur/lnna/[fuelengineinna.cpp](#)

## 6.32 FuelGauge Class Reference

Inheritance diagram for FuelGauge:



Collaboration diagram for FuelGauge:



### Public Member Functions

- **FuelGauge** (QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

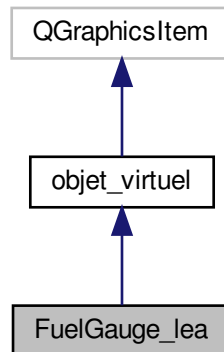
- float **pi** = 3.14159265359

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

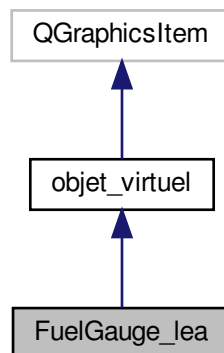
- serveur/Youcef/fuelgauge.h
- serveur/Youcef/fuelgauge.cpp

### 6.33 FuelGauge\_lea Class Reference

Inheritance diagram for FuelGauge\_lea:



Collaboration diagram for FuelGauge\_lea:



#### Public Member Functions

- [FuelGauge\\_lea](#) (double, double, double, double)  
*FuelGauge\_lea::FuelGauge\_lea*, Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 30.
- `QRectF boundingRect () const`
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*FuelGauge\_lea::paint* Fonction permettant l'affichage du compteur.

## Public Attributes

- double **x**
- double **y**
- double **x2**
- double **y2**
- double **x3**
- double **y3**
- double **width**
- double **height**

## Additional Inherited Members

### 6.33.1 Constructor & Destructor Documentation

#### 6.33.1.1 FuelGauge\_lea()

```
FuelGauge_lea::FuelGauge_lea (
    double param_x,
    double param_y,
    double param_width,
    double param_height )
```

[FuelGauge\\_lea::FuelGauge\\_lea](#), Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 30.

#### Parameters

<i>param_x</i>	position horizontale du coin en haut à gauche du rectangle définissant la jauge
<i>param_y</i>	position verticale coin en haut à gauche du rectangle définissant la jauge
<i>param_width</i>	largeur voulue de la jauge
<i>param_height</i>	hauteur voulue de la jauge

### 6.33.2 Member Function Documentation

#### 6.33.2.1 paint()

```
void FuelGauge_lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[FuelGauge\\_lea::paint](#) Fonction permettant l'affichage du compteur.

Cette fonction construit un compteur en plusieurs étapes : création du cadre de la jauge, des rectangles indiquant le niveau d'essence, du E et du F ainsi que l'icone/voyant d'essence.

#### Parameters

<i>painter</i>	
----------------	--

Création du cadre.

Le cadre est basé sur un rectangle obtenu avec un drawRect paramétrable.

Création des graduations du cadre.

Les graduations sont obtenus avec un drawLine d'un vecteur(pour automatiser le dessin avec un interval constant), ici 6 graduations seont dessinées représentant le 0%, 20%, 40%, 60%, 80% et 100%.

Création des rectangle amplissant le cadre.

Les rectangles sont obtenus avec un drawRect d'un vecteur(pour automatiser le dessin avec un interval constant), ici 10 rectangles sont dessinées représentant chacun 10% du niveau d'essence.

Création de l'icone d'essence de couleur blanche et devenant rouge l'orsqu'on passe en dessous de 25% d'essence.

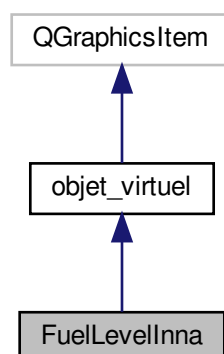
Les voyants sont obtenus via un drawPixmap dans une boucle if pour changer le pixmap en fonction de la "value" d'essence(devient rouge quand le niveau passe en dessous de 25%).

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

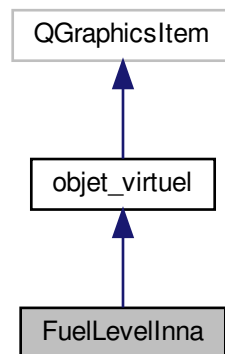
- serveur/Lea/fuelgauge\_lea.h
- serveur/Lea/[fuelgauge\\_lea.cpp](#)

## 6.34 FuelLevelInna Class Reference

Inheritance diagram for FuelLevelInna:



Collaboration diagram for FuelLevelInna:



### Public Member Functions

- **FuelLevelInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **FuelLevelInna** (int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- double **l**
- double **k**

### Additional Inherited Members

#### 6.34.1 Member Function Documentation

#### 6.34.1.1 paint()

```
void FuelLevelInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

dessin du cercle central

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une ellipse avec un gradient de couleur.

Dessin de la flèche.

On crée le gradient linear et un triangle, qu'on remplit avec un gradient de couleur. La position de la pointe de flèche est mathématiquement calculée.

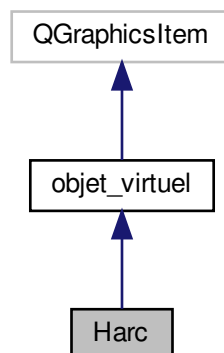
dessin de l'icone du niveau d'essence

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

- serveur/lnna/fuellevelinna.h
- serveur/lnna/[fuellevelinna.cpp](#)

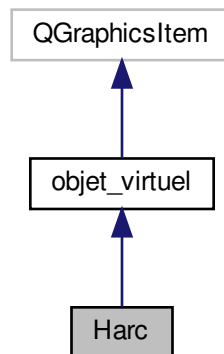
## 6.35 Harc Class Reference

Inheritance diagram for Harc:





Collaboration diagram for Harc:



### Public Member Functions

- **Harc** (int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

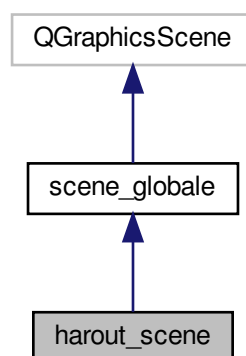
### Additional Inherited Members

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

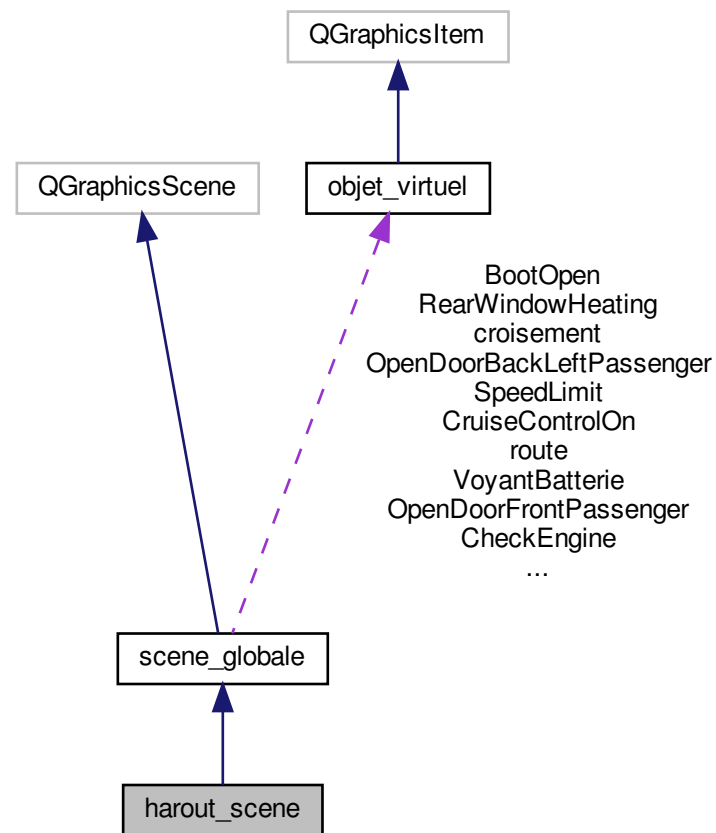
- serveur/Harout/harc.h
- serveur/Harout/harc.cpp

## 6.36 harout\_scene Class Reference

Inheritance diagram for harout\_scene:



Collaboration diagram for harout\_scene:



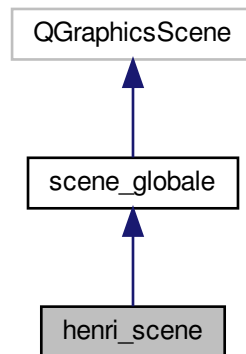
### Additional Inherited Members

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

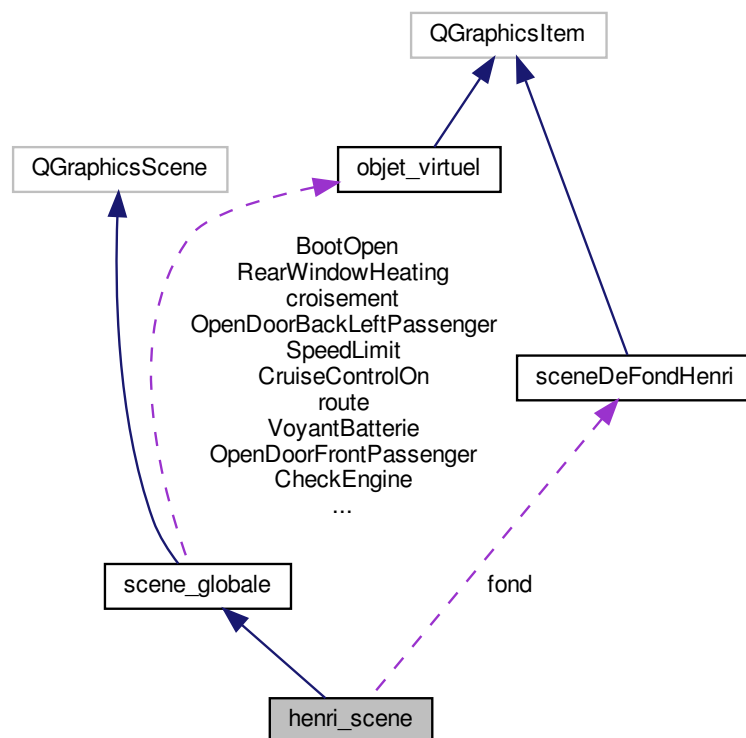
- `serveur/Harout/harout_scene.h`
- `serveur/Harout/harout_scene.cpp`

## 6.37 henri\_scene Class Reference

Inheritance diagram for henri\_scene:



Collaboration diagram for henri\_scene:



### Public Member Functions

- **henri\_scene** ([scene\\_globale](#) \*parent=nullptr)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const

### Public Attributes

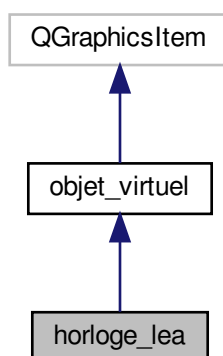
- [sceneDeFondHenri](#) \* **fond**

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

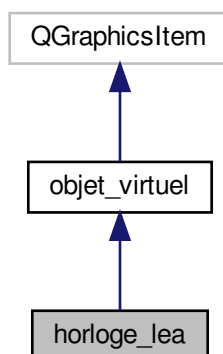
- serveur/Henri/henri\_scene.h
- serveur/Henri/henri\_scene.cpp

## 6.38 horloge\_lea Class Reference

Inheritance diagram for horloge\_lea:



Collaboration diagram for horloge\_lea:



### Public Member Functions

- **horloge\_lea** (int, int, int, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **x**
- int **y**
- int **width**
- int **height**

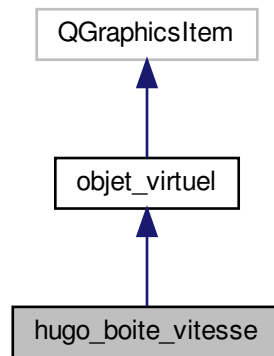
### Additional Inherited Members

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

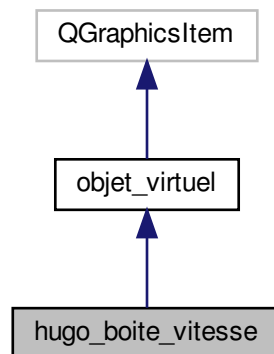
- serveur/Lea/horloge\_lea.h
- serveur/Lea/horloge\_lea.cpp

## 6.39 hugo\_boite\_vitesse Class Reference

Inheritance diagram for hugo\_boite\_vitesse:



Collaboration diagram for hugo\_boite\_vitesse:



### Public Member Functions

- [hugo\\_boite\\_vitesse](#) ()  
*hugo\_boite\_vitesse::hugo\_boite\_vitesse*
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*hugo\_ecran::paint*

## Additional Inherited Members

### 6.39.1 Constructor & Destructor Documentation

#### 6.39.1.1 hugo\_boite\_vitesse()

```
hugo_boite_vitesse::hugo_boite_vitesse ( )
```

[hugo\\_boite\\_vitesse::hugo\\_boite\\_vitesse](#)

Initialise la variable value de la classe mère à 1 i.e. au mode P (Park)

### 6.39.2 Member Function Documentation

#### 6.39.2.1 paint()

```
void hugo_boite_vitesse::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[hugo\\_ecran::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

La fonction paint va permettre l'affichage des différents modes P,R,N,D,3,2,L suivant un arc de cercle autour du compteur vitesse. L'affichage se fait par l'intermédiaire d'une boucle. Si l'itérateur correspond à value, le mode de transmission est affiché en vert avec un effet de halo autour.

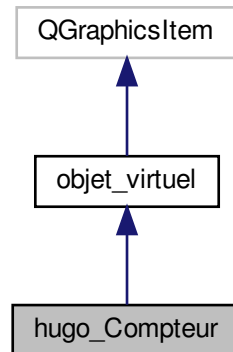
The documentation for this class was generated from the following files:

- serveur/Hugo/hugo\_boite\_vitesse.h
- serveur/Hugo/[hugo\\_boite\\_vitesse.cpp](#)

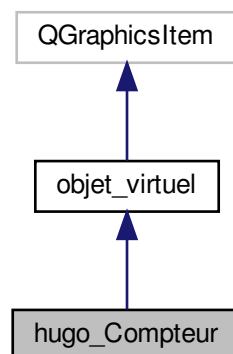


## 6.40 hugo\_Compteur Class Reference

Inheritance diagram for hugo\_Compteur:



Collaboration diagram for hugo\_Compteur:



### Public Member Functions

- [`hugo\_Compteur`](#) (int, int, int, int, int, QStringList, int, int, int, int, int, int, int param\_critique=100, int red2=100, int green2=100, int blue2=100)  
*[`hugo\_Compteur::hugo\_Compteur`](#) Constructeur de la classe, permet d'initialiser tous les paramètres*
- `QRectF boundingRect () const`
- void [`paint`](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*[`hugo\_Compteur::paint`](#) Fonction permettant l'affichage du compteur*

## Protected Attributes

- int **x** =0
- int **y** =0
- int **r** =100
- int **start\_angle** =0
- int **end\_angle** =360
- int **nbre\_graduations** =12
- int **critique**
- int **angle** =0
- int **value2** =0
- int **r\_verre**
- int **direction\_grad** =1
- QColor **couleur** =QColor(100,100,100)
- QColor **couleur2** =QColor(100,100,100)
- QColor **couleurgrad** =QColor(100,100,100,50)
- QColor **couleurgrad2** =QColor(100,100,100,50)
- QStringList **graduations**

## Additional Inherited Members

### 6.40.1 Constructor & Destructor Documentation

#### 6.40.1.1 hugo\_Compteur()

```
hugo_Compteur::hugo_Compteur (
    int param_x,
    int param_y,
    int param_r,
    int param_start_angle,
    int param_end_angle,
    QStringList param_graduations,
    int param_value,
    int param_r_verre,
    int param_direction_grad,
    int red,
    int green,
    int blue,
    int param_critique = 100,
    int red2 = 100,
    int green2 = 100,
    int blue2 = 100 )
```

[hugo\\_Compteur::hugo\\_Compteur](#) Constructeur de la classe, permet d'initialiser tous les paramètres

#### Parameters

<i>param_x</i>	position horizontale du centre du compteur
<i>param_y</i>	position verticale du centre du compteur
<i>param_r</i>	rayon
<i>param_start_angle</i>	Angle de départ pour le tracé de l'arc de cercle

## Parameters

<i>param_end_angle</i>	Angle de fin pour le tracé de l'arc de cercle
<i>param_graduations</i>	QStringList des textes à afficher sur les graduations. Le nombre d'éléments correspondra au nombre de grandes graduations
<i>param_value</i>	valeur max de la quantité représentée par le compteur, utile pour le calcul du rapport angle de l'aiguille / valeur à afficher
<i>param_r_verre</i>	permet de varier la surface du disc pour l'effet de verre
<i>param_direction_grad</i>	permet de varier la direction du gradient : +1 = blanc->noir, -1= noir->blanc
<i>red</i>	Couleurs rgb de l'arc de cercle et des graduations
<i>green</i>	
<i>blue</i>	
<i>param_critique</i>	Parmètre optionnel, graduation à partir de laquelle un changement de couleur doit être effectué sur le compteur
<i>red2</i>	Couleur optionnelle,
<i>green2</i>	
<i>blue2</i>	

## 6.40.2 Member Function Documentation

## 6.40.2.1 paint()

```
void hugo_Compteur::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[hugo\\_Compteur::paint](#) Fonction permettant l'affichage du compteur

Cette fonction construit un compteur en plusieurs étapes : création du cadran, des graduations, ajout du texte sur les graduations, ajout de l'aiguille, superposition d'un cercle avec effet de reflet.

## Parameters

<i>painter</i>	
----------------	--

Création du cadran.

Le cadran est basé sur un arc de cercle dessiné à l'aide de la fonction drawArc. Cette méthode est répétée dans une boucle pour obtenir un effet de gradient. Si lors de la création de l'objet, le paramètre param\_critique a été défini à une autre valeur qu'à 100, une partie du cadran serait redessinée d'une autre couleur.

Création des graduations.

Les graduations sont créées en utilisant la fonction drawLine. 2 boucles sont nécessaires : l'une pour les petites graduations, l'autre pour les grandes.

Ajout du texte sur les graduations\$

Le texte est positionné de la même façon que les graduations i.e. sur un arc de cercle. Le cercle a ensuite été translaté de manière à correspondre le plus possible aux positions des graduations.

Création de l'aiguille.

Dans un premier temps, affichage d'un petit cercle à la base de l'aiguille. Puis affichage de l'aiguille à l'aide de la fonction `drawPolygon`. Le paramètre `angle`, utilisant la fonction `getValue()` de la classe `objet_virtuel`, positionne l'aiguille au bon endroit.

Effet de verre

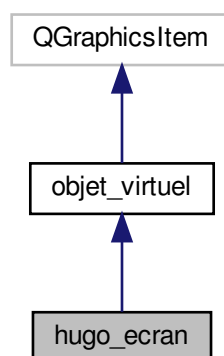
L'effet de verre est obtenu en superposant un cercle semi-transparent au compteur. L'option `direction_grad` permet de définir le sens du gradient.

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

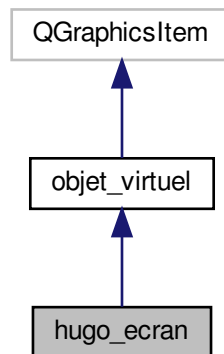
- `serveur/Hugo/hugo_compteur.h`
- `serveur/Hugo/hugo_compteur.cpp`

## 6.41 hugo\_ecran Class Reference

Inheritance diagram for `hugo_ecran`:



Collaboration diagram for hugo\_ecran:



### Public Member Functions

- [hugo\\_ecran\(\)](#)  
[hugo\\_ecran::hugo\\_ecran](#)
- `QRectF boundingRect() const`
- `void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`  
[hugo\\_ecran::paint](#)

### Additional Inherited Members

#### 6.41.1 Constructor & Destructor Documentation

##### 6.41.1.1 hugo\_ecran()

```
hugo_ecran::hugo_ecran ( )
```

[hugo\\_ecran::hugo\\_ecran](#)

Constructeur sans paramètres spécifiques. Initialise simplement la valeur value de la classe mère à 0.

#### 6.41.2 Member Function Documentation

##### 6.41.2.1 paint()

```
void hugo_ecran::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[hugo\\_ecran::paint](#)

## Parameters

<i>painter</i>	
----------------	--

La fonction paint va permettre de réaliser un affichage pour le compteur kilometrique et l'heure. Dans un premier temps, affichage des objets fixes, puis on récupère et affiche les différentes variables. Affichage d'un rectangle avec gradient, et effet de luminosité

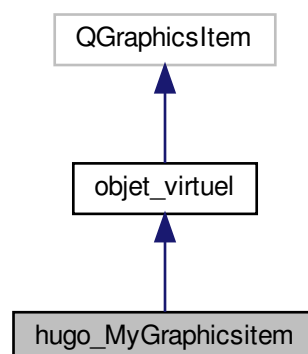
Affichage des différents textes. Le kilométrage est récupéré grâce à la fonction getValue.

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

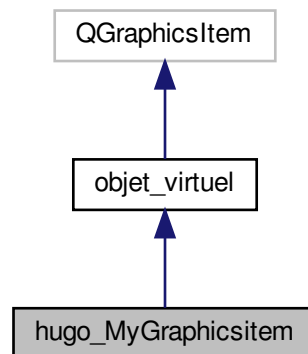
- serveur/Hugo/hugo\_ecran.h
- serveur/Hugo/[hugo\\_ecran.cpp](#)

## 6.42 hugo\_MyGraphicsitem Class Reference

Inheritance diagram for hugo\_MyGraphicsitem:



Collaboration diagram for hugo\_MyGraphicsitem:



## Public Member Functions

- [hugo\\_MyGraphicsitem \(\)](#)  
*[hugo\\_MyGraphicsitem::hugo\\_MyGraphicsitem](#)*
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`  
*[hugo\\_MyGraphicsitem::paint](#)*

## Public Attributes

- `int current_speed =30`
- `QString station =""`
- `float km =0.0`

## Additional Inherited Members

### 6.42.1 Constructor & Destructor Documentation

#### 6.42.1.1 hugo\_MyGraphicsitem()

```
hugo_MyGraphicsitem::hugo_MyGraphicsitem ( )
```

[hugo\\_MyGraphicsitem::hugo\\_MyGraphicsitem](#)

Constructeur sans paramètres spécifiques. Initialise simplement la valeur value de la classe mère à 0.

## 6.42.2 Member Function Documentation

### 6.42.2.1 paint()

```
void hugo_MyGraphicsitem::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[hugo\\_MyGraphicsitem::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

<Antialiasing//

Création du fond (gradient ou image)

Création des arcs de cercles gris, et d'un fond noir pour poser les objects

Création de l'arc de cercle fermé bleu qui entoure le compteur de vitesse

Affichage de texte dans les différents compteurs

Afficheur pour les stations de radios

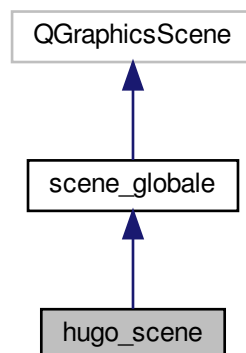
Affichage d'une icone jauge d'essence

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

- serveur/Hugo/hugo\_mygraphicsitem.h
- serveur/Hugo/hugo\_mygraphicsitem.cpp

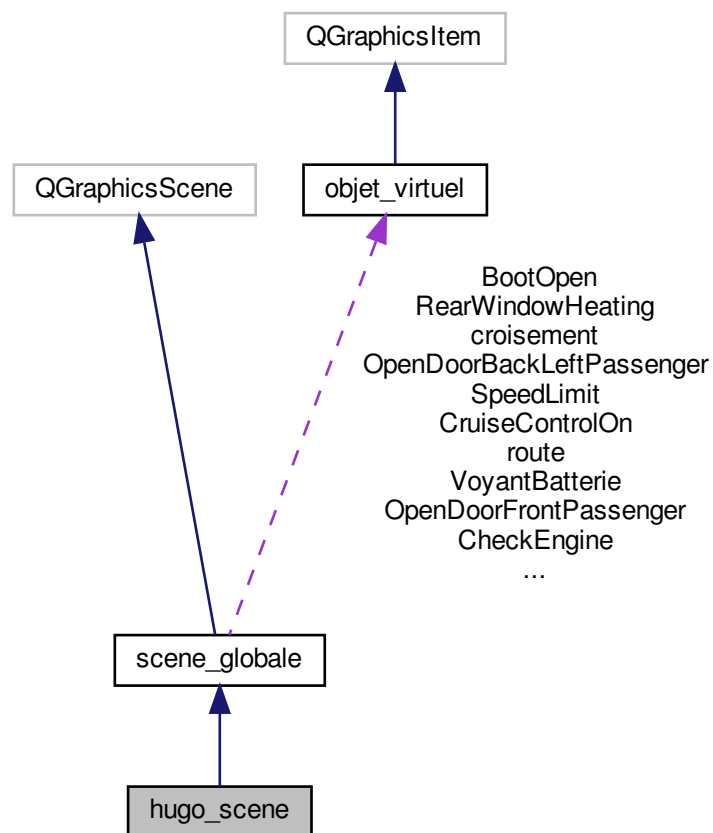
## 6.43 hugo\_scene Class Reference

Inheritance diagram for hugo\_scene:





Collaboration diagram for hugo\_scene:



## Public Member Functions

- **hugo\_scene** ([scene\\_globale](#) \*parent=nullptr)

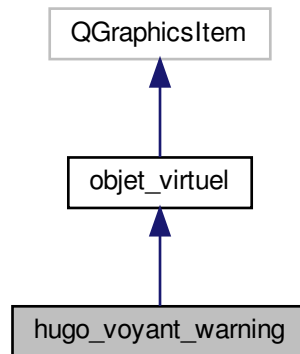
## Additional Inherited Members

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

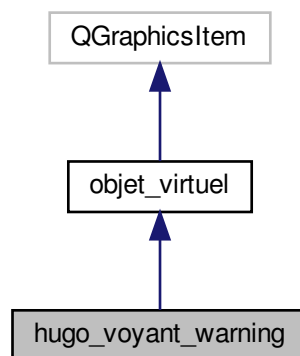
- serveur/Hugo/hugo\_scene.h
- serveur/Hugo/hugo\_scene.cpp

## 6.44 hugo\_voyant\_warning Class Reference

Inheritance diagram for hugo\_voyant\_warning:



Collaboration diagram for hugo\_voyant\_warning:



### Public Member Functions

- [hugo\\_voyant\\_warning](#) ()  
[hugo\\_voyant\\_warning::hugo\\_voyant\\_warning](#)
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const
- void [MAJ](#) ()  
[hugo\\_voyants\\_clignotant::MAJ](#) Fonction de mise à jour de l'affichage.

## Public Attributes

- int **cligno**

## Additional Inherited Members

### 6.44.1 Constructor & Destructor Documentation

#### 6.44.1.1 hugo\_voyant\_warning()

```
hugo_voyant_warning::hugo_voyant_warning ( )
```

[hugo\\_voyant\\_warning::hugo\\_voyant\\_warning](#)

Constructeur de la classe initialisant la variable value de la classe mère, et la variable cligno qui permet de gérer l'affichage alternatif des voyants.

### 6.44.2 Member Function Documentation

#### 6.44.2.1 MAJ()

```
void hugo_voyant_warning::MAJ ( )
```

[hugo\\_voyants\\_clignotant::MAJ](#) Fonction de mise à jour de l'affichage.

La valeur cligno controle l'opacité du painter, permettant de le rendre visible ou non

#### 6.44.2.2 paint()

```
void hugo_voyant_warning::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Va chercher la valeur de la variable "valeur" de la classe hugo\_voyants. 1 correspond à l'affichage du voyant.

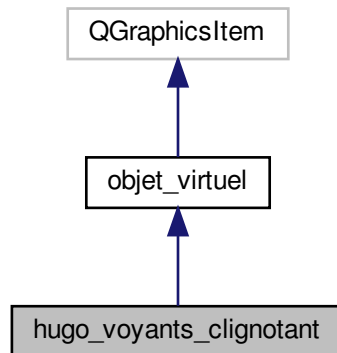
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

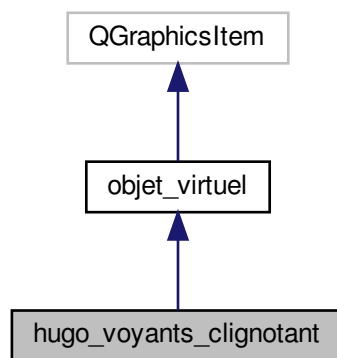
- serveur/Hugo/hugo\_voyant\_warning.h
- serveur/Hugo/[hugo\\_voyant\\_warning.cpp](#)

## 6.45 hugo\_voyants\_clignotant Class Reference

Inheritance diagram for hugo\_voyants\_clignotant:



Collaboration diagram for hugo\_voyants\_clignotant:



### Public Member Functions

- [hugo\\_voyants\\_clignotant \(\)](#)  
*[hugo\\_voyants\\_clignotant::hugo\\_voyants\\_clignotant](#). Constructeur de la classe.*
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const
- void [MAJ](#) ()  
*[hugo\\_voyants\\_clignotant::MAJ](#) Fonction de mise à jour de l'affichage. La valeur cligno controle l'opacité du painter, permettant de le rendre visible ou non*

## Public Attributes

- int **cligno**

## Additional Inherited Members

### 6.45.1 Constructor & Destructor Documentation

#### 6.45.1.1 hugo\_voyants\_clignotant()

```
hugo_voyants_clignotant::hugo_voyants_clignotant ( )
```

[hugo\\_voyants\\_clignotant::hugo\\_voyants\\_clignotant](#). Constructeur de la classe.

Constructeur de la classe initialisant la variable *value* de la classe mère, et la variable *cligno* qui permet de gérer l'affichage alternatif des voyants.

### 6.45.2 Member Function Documentation

#### 6.45.2.1 paint()

```
void hugo_voyants_clignotant::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

<Antialiasing//

Va chercher la valeur de la variable "valeur" de la classe *hugo\_voyants*. 1 correspond à l'affichage du clignotant droit, -1 à celui du clignotant gauche.

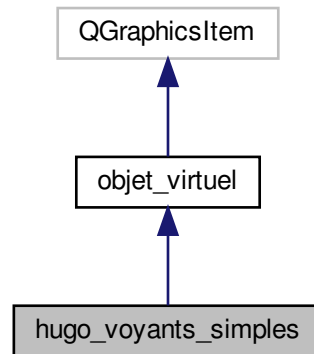
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné

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

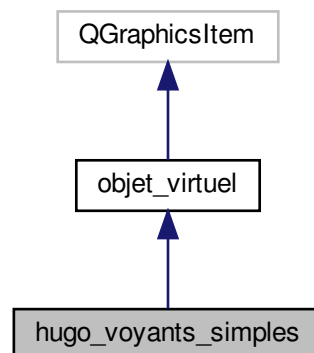
- serveur/Hugo/hugo\_voyants\_clignotant.h
- serveur/Hugo/hugo\_voyants\_clignotant.cpp

## 6.46 hugo\_voyants\_simples Class Reference

Inheritance diagram for hugo\_voyants\_simples:



Collaboration diagram for hugo\_voyants\_simples:



### Public Member Functions

- [hugo\\_voyants\\_simples](#) (int, int, QString, int red=255, int green=0, int blue=0, int param\_size=30)  
*hugo\_voyants\_simples::hugo\_voyants\_simples.* Constructeur permettant de paramétrer la position et le halo des voyants
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const

## Protected Attributes

- int **x**
- int **y**
- int **size** =30
- QString **chemin**
- QColor **couleur**

## Additional Inherited Members

### 6.46.1 Constructor & Destructor Documentation

#### 6.46.1.1 hugo\_voyants\_simples()

```
hugo_voyants_simples::hugo_voyants_simples (
    int param_x,
    int param_y,
    QString param_chemin,
    int red = 255,
    int green = 0,
    int blue = 0,
    int param_size = 30 )
```

[hugo\\_voyants\\_simples::hugo\\_voyants\\_simples](#). Constructeur permettant de paramétrer la position et le halo des voyants

#### Parameters

<i>param_x</i>	position horizontale
<i>param_y</i>	position verticale
<i>param_chemin</i>	Nom de la ressource
<i>red</i>	Paramètre optionnel pour gérer la couleur du halo donnant l'effet de brillance
<i>green</i>	
<i>blue</i>	
<i>param_size</i>	Paramètre optionnel pour gérer la taille du voyant

### 6.46.2 Member Function Documentation

#### 6.46.2.1 paint()

```
void hugo_voyants_simples::paint (
    QPainter * painter,
```

```
const QStyleOptionGraphicsItem * option,
QWidget * widget )
```

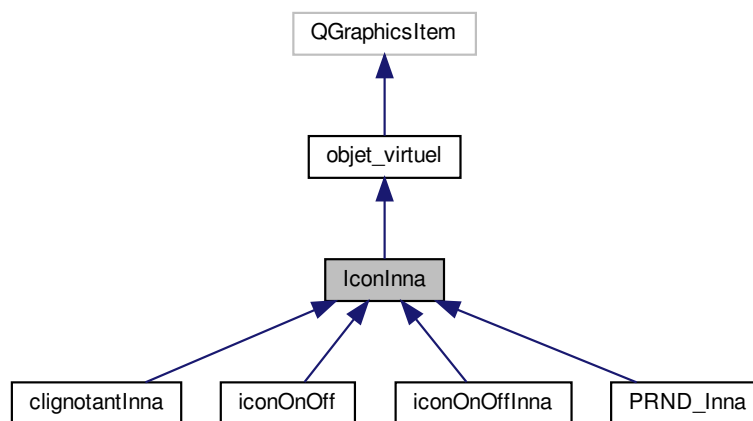
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché.

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

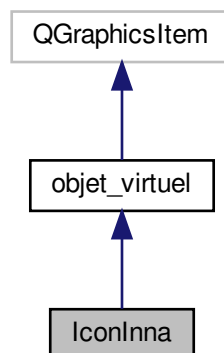
- serveur/Hugo/hugo\_voyants\_simples.h
- serveur/Hugo/[hugo\\_voyants\\_simples.cpp](#)

## 6.47 IconInna Class Reference

Inheritance diagram for IconInna:



Collaboration diagram for IconInna:





## Public Member Functions

- **IconInna** ([objet\\_virtuel](#) \*parent=nullptr)
- **IconInna** (QPoint p, QSize s, QString pth)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- void [init](#) (QPoint p, QSize s, QString pth)

## Public Attributes

- QString **imagePath**
- QPoint **position**
- QSize **size**
- int **val**

## Additional Inherited Members

### 6.47.1 Member Function Documentation

#### 6.47.1.1 [init\(\)](#)

```
void IconInna::init (  
    QPoint p,  
    QSize s,  
    QString pth )
```

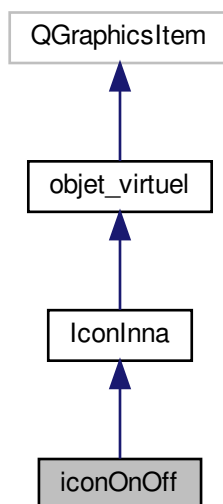
La fonction [init\(QPoint p, QSize s, QString pth\)](#) permet de reinitialiser un voyant.

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

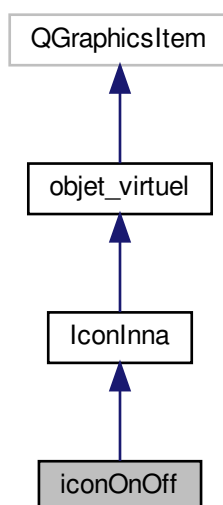
- serveur/Inna/iconinna.h
- serveur/Inna/[iconinna.cpp](#)

## 6.48 iconOnOff Class Reference

Inheritance diagram for iconOnOff:



Collaboration diagram for iconOnOff:



## Public Member Functions

- **iconOnOff** ([IconInna](#) \*parent=nullptr)
- **iconOnOff** (QPoint p, QSize s, QString pth, double Z)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

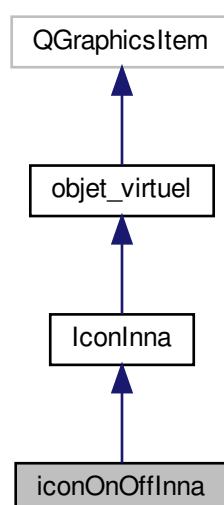
## Additional Inherited Members

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

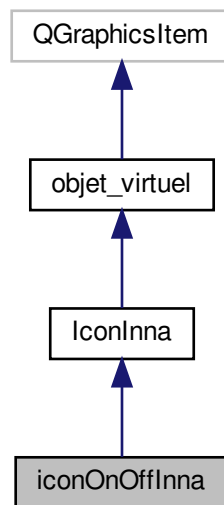
- serveur/lnna/icononoff.h
- serveur/lnna/icononoff.cpp

## 6.49 iconOnOffInna Class Reference

Inheritance diagram for iconOnOffInna:



Collaboration diagram for iconOnOffInna:



## Public Member Functions

- **iconOnOffInna** ([IconInna](#) \*parent=nullptr)
- **iconOnOffInna** (QPoint p, QSize s, QString pth, double Z)  
*[iconOnOffInna::iconOnOffInna\(QPoint p, QSize s, QString pth, double Z\)](#) - le constructeur multiparametrique*
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

## Additional Inherited Members

### 6.49.1 Constructor & Destructor Documentation

#### 6.49.1.1 iconOnOffInna()

```

iconOnOffInna::iconOnOffInna (
    QPoint p,
    QSize s,
    QString pth,
    double Z )
  
```

[iconOnOffInna::iconOnOffInna\(QPoint p, QSize s, QString pth, double Z\)](#) - le constructeur multiparametrique

Le constructeur permet d'instancier un voyant a la position p, de la taille s,

avec une image donnée par le chemin pth et au niveau Z dans l'ordre de l'affichage des elements.

## 6.49.2 Member Function Documentation

### 6.49.2.1 paint()

```
void iconOnOffInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[iconOnOffInna::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

La fonction permet l'affichage d'un voyant instancié un fonction de la valeur *value* :

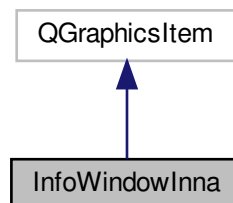
Si *value* = 1 => l'etait *allumé* , sinon l'etait *etaint*.

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

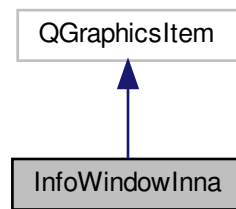
- serveur/Inna/icononoffinna.h
- serveur/Inna/[icononoffinna.cpp](#)

## 6.50 InfoWindowInna Class Reference

Inheritance diagram for InfoWindowInna:



Collaboration diagram for InfoWindowInna:



## Public Member Functions

- **InfoWindowInna** (QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[InfoWindowInna::paint.](#)

## Public Attributes

- int **r**

## 6.50.1 Member Function Documentation

### 6.50.1.1 paint()

```

void InfoWindowInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
  
```

[InfoWindowInna::paint.](#)

#### Parameters

<i>painter</i>	
----------------	--

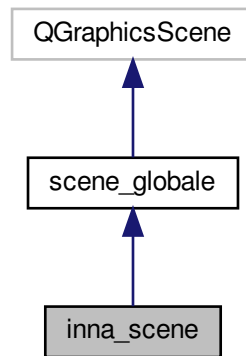
La fonction crée un objet static qui sert comme un fond pour des voyants et des messages d'info.

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

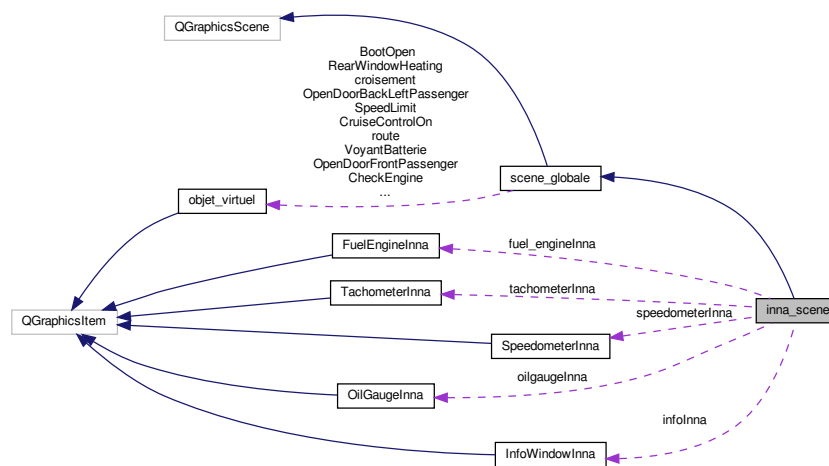
- serveur/Inna/infowindowinna.h
- serveur/Inna/[infowindowinna.cpp](#)

## 6.51 inna\_scene Class Reference

Inheritance diagram for inna\_scene:



Collaboration diagram for inna\_scene:



### Public Member Functions

- `inna_scene` (`scene_globale` \*parent=nullptr)
- `QRectF boundingRect ()` const
- `void paint` (`QPainter` \*painter, const `QStyleOptionGraphicsItem` \*option, `QWidget` \*widget)

### Public Attributes

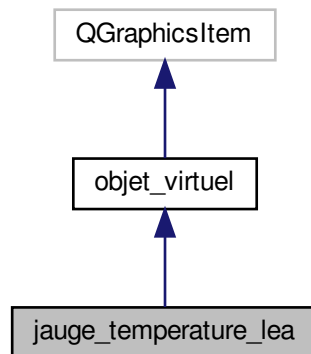
- [SpeedometerInna](#) \* **speedometerInna**
- [TachometerInna](#) \* **tachometerInna**
- [OilGaugeInna](#) \* **oilgaugeInna**
- [InfoWindowInna](#) \* **infoInna**
- [FuelEngineInna](#) \* **fuel\_engineInna**

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

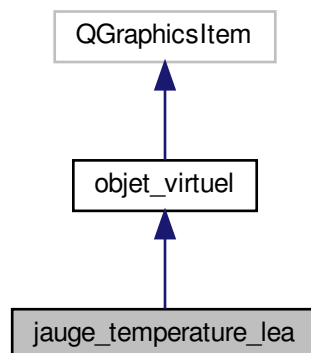
- [serveur/Inna/inna\\_scene.h](#)
- [serveur/Inna/inna\\_scene.cpp](#)

## 6.52 `jauge_temperature_lea` Class Reference

Inheritance diagram for `jauge_temperature_lea`:



Collaboration diagram for `jauge_temperature_lea`:





## Public Member Functions

- [jauge\\_temperature\\_lea](#) (double, double, double, int, int, int)  
*[jauge\\_temperature\\_lea::jauge\\_temperature\\_lea](#), Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 50.*
- `QRectF boundingRect ()` const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*[jauge\\_temperature\\_lea::paint](#)*

## Public Attributes

- const double **pi** =3.14159265359
- double **x**
- double **y**
- double **r**
- int **angle\_debut**
- int **span\_angle**
- int **temperature\_max** =130
- int **v**

## Additional Inherited Members

### 6.52.1 Constructor & Destructor Documentation

#### 6.52.1.1 `jauge_temperature_lea()`

```
jauge_temperature_lea::jauge_temperature_lea (
    double param_x,
    double param_y,
    double param_r,
    int param_start,
    int param_spanAngle,
    int param_tempMax )
```

[jauge\\_temperature\\_lea::jauge\\_temperature\\_lea](#), Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 50.

#### Parameters

<i>param_x</i>	position horizontale du centre du compteur
<i>param_y</i>	position verticale du centre du compteur
<i>param_r</i>	rayon du compteur et taille de l'aiguille
<i>param_start</i>	Angle de départ pour le tracé de l'arc de cercle
<i>param_spanAngle</i>	angle total du cadran
<i>param_tempMax</i>	determine la température max à laquelle va le cadran

## 6.52.2 Member Function Documentation

### 6.52.2.1 `paint()`

```
void jauge_temperature_lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

#### [jauge\\_temperature\\_lea::paint](#)

Cette fonction construit un compteur en plusieurs étapes : création du cadran, des graduations, ajout du texte sur les graduations, ajout de l'aiguille et d'un pixmap représentant le voyant de température.

#### Parameters

<i>painter</i>	
----------------	--

calibration de "value"

La variable "value" donnée par la fonction [getValue\(\)](#) de la classe [objet\\_virtuel](#) est mise à 0 si la valeur entrée est inférieure à 0 et mise à "valueMax" si la valeur entrée est plus grande.

Création du cadran.

Le cadran est basé sur un arc de cercle dessiné à l'aide de la fonction `drawArc`. Le cercle est dessiné avec un `QPen` à qui a été donné une `QBrush` contenant un gradient radial de couleur bleu.

Création des graduations du cadran.

Les graduations sont créées en utilisant la fonction `drawLine` et une boucle `if` pour changer la couleur des graduations en rouge quand on dépasse les 100° et en bleu quand on est en dessous de 70°. Les graduations sont dessinées tous les 20°.

Ajout du texte sur les graduations

Le texte est positionné de la même façon que les graduations, tous les 40°, avec une translation de manière à correspondre le plus possible aux positions des graduations.

Création de l'aiguille.

Dans un premier temps, affichage de l'aiguille à l'aide de la fonction `drawConvexPolygon`. Puis affichage d'un petit cercle à la base de l'aiguille.

Création de l'icone de température de couleur blanche et devenant rouge lorsqu'on passe au dessus des 100°.

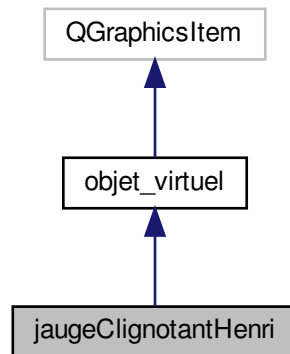
Les voyants sont obtenus via un `drawPixmap` dans une boucle `if` pour changer le pixmap en fonction de la "value" de la température.

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

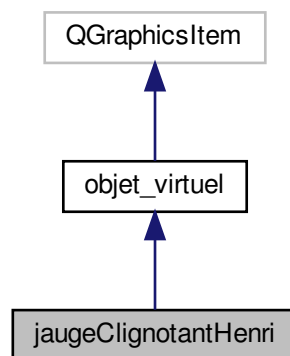
- `serveur/Lea/jauge_temperature_lea.h`
- `serveur/Lea/jauge_temperature_lea.cpp`

## 6.53 `jaugeClignotantHenri` Class Reference

Inheritance diagram for `jaugeClignotantHenri`:



Collaboration diagram for `jaugeClignotantHenri`:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const
- void **MAJ** ()
- void **MAJ2** ()

### Public Attributes

- int **cligno**

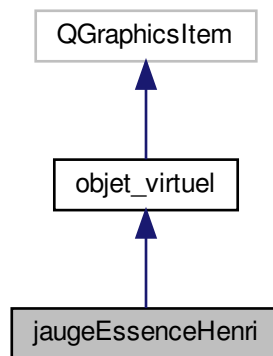
## Additional Inherited Members

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

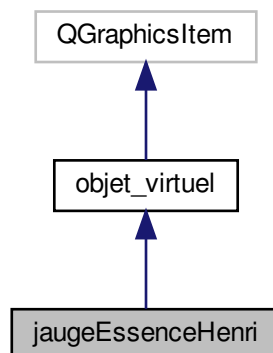
- serveur/Henri/jaugeclignotanthenri.h
- serveur/Henri/jaugeclignotanthenri.cpp

## 6.54 jaugeEssenceHenri Class Reference

Inheritance diagram for jaugeEssenceHenri:



Collaboration diagram for jaugeEssenceHenri:



### Public Member Functions

- `jaugeEssenceHenri` (`objet_virtuel` \*parent=nullptr)
- `QRectF boundingRect` () const
- void `paint` (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

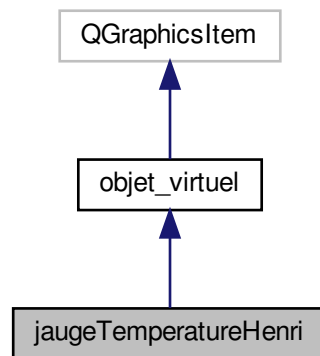
### Additional Inherited Members

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

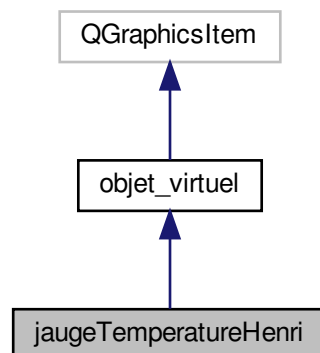
- `serveur/Henri/jaugeessencehenri.h`
- `serveur/Henri/jaugeessencehenri.cpp`

## 6.55 `jaugeTemperatureHenri` Class Reference

Inheritance diagram for `jaugeTemperatureHenri`:



Collaboration diagram for `jaugeTemperatureHenri`:



### Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

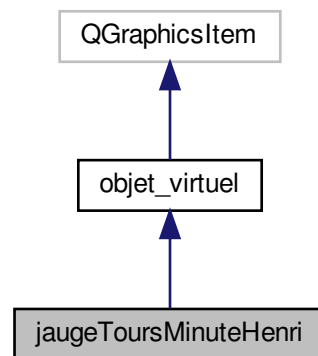
### Additional Inherited Members

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

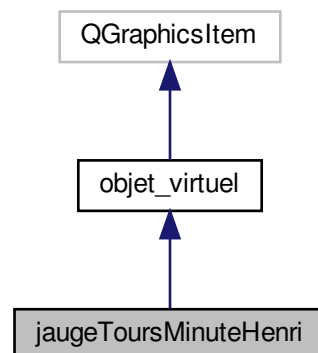
- serveur/Henri/jaugetemperaturehenri.h
- serveur/Henri/jaugetemperaturehenri.cpp

## 6.56 jaugeToursMinuteHenri Class Reference

Inheritance diagram for jaugeToursMinuteHenri:



Collaboration diagram for jaugeToursMinuteHenri:



### Public Member Functions

- `jaugeToursMinuteHenri` (`objet_virtuel *parent=NULLptr`)
- `QRectF boundingRect ()` const
- void `paint` (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

### Public Attributes

- int `epaisseurTraitToursMinure`
- int `tailleTextetoursMinute`
- `QString styleTexte`

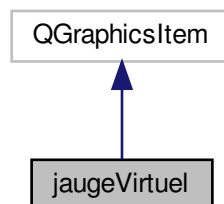
### Additional Inherited Members

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

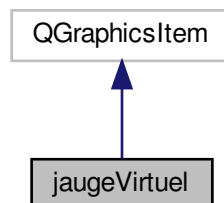
- `serveur/Henri/jaugetoursminutehenri.h`
- `serveur/Henri/jaugetoursminutehenri.cpp`

## 6.57 `jaugeVirtuel` Class Reference

Inheritance diagram for `jaugeVirtuel`:



Collaboration diagram for `jaugeVirtuel`:



## Public Member Functions

- **jaugeVirtuel** (QGraphicsItem \*parent=nullptr)
- int **getValeur** () const
- void **setValeur** (int value)
- int **getValeurMax** () const

## Protected Attributes

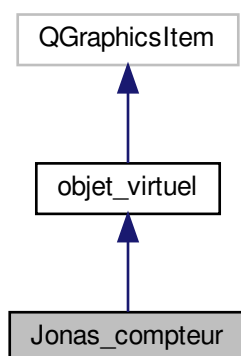
- int **valeur**
- int **valeurMax**

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

- serveur/Florian/jaugevirtuel.h
- serveur/Florian/jaugevirtuel.cpp

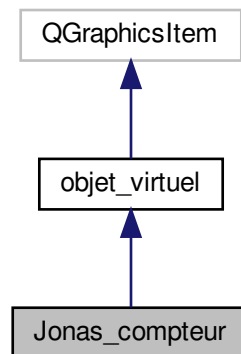
## 6.58 Jonas\_compteur Class Reference

Inheritance diagram for Jonas\_compteur:





Collaboration diagram for Jonas\_compteur:



## Public Member Functions

- [Jonas\\_compteur \(\)](#)  
*Jonas\_compteur::Jonas\_compteur.*
- [Jonas\\_compteur](#) (int max, QStringList gradList, float startAngle, float endAngle, int critic, QString textCenter, int ngrad, bool line=0, int mod=2, int size=150)  
*Jonas\_compteur::Jonas\_compteur.*
- QRectF [boundingRect \(\)](#) const override  
*Jonas\_compteur::boundingRect.*
- void [paint](#) (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override  
*Jonas\_compteur::paint.*
- void [setTextLabel](#) (QString newText)
- void [setAlphaAngle](#) (float newAngle)
- void [setBetaAngle](#) (float newAngle)
- void [setGraduation](#) (int n)
- void [setHLine](#) (bool b)
- void [setMod](#) (int n)
- void [setGaugeSize](#) (int size)
- float [getAlpha](#) () const
- float [getBeta](#) () const
- int [getGaugeSize](#) () const
- QString [gettextLabel](#) () const
- float [speedToAngle](#) (float)  
*Jonas\_compteur::speedToAngle.*

## Additional Inherited Members

### 6.58.1 Constructor & Destructor Documentation

**6.58.1.1 Jonas\_compteur()** [1/2]

```
Jonas_compteur::Jonas_compteur ( )
```

[Jonas\\_compteur::Jonas\\_compteur.](#)

constructeur par défaut

**6.58.1.2 Jonas\_compteur()** [2/2]

```
Jonas_compteur::Jonas_compteur (
    int max,
    QStringList gradList,
    float startAngle,
    float endAngle,
    int critic,
    QString textCenter,
    int ngrad,
    bool line = 0,
    int modulo = 2,
    int size = 150 )
```

[Jonas\\_compteur::Jonas\\_compteur.](#)

constructeur surchargé

**Parameters**

<i>max</i>	
<i>gradList</i>	
<i>startAngle</i>	
<i>endAngle</i>	
<i>critic</i>	
<i>textCenter</i>	
<i>ngrad</i>	
<i>line</i>	
<i>modulo</i>	
<i>size</i>	

**6.58.2 Member Function Documentation****6.58.2.1 boundingRect()**

```
QRectF Jonas_compteur::boundingRect ( ) const [override]
```

[Jonas\\_compteur::boundingRect.](#)

**Returns**

retourne un rectangle qui encadre l'objet

### 6.58.2.2 paint()

```
void Jonas_compteur::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * ,
    QWidget * ) [override]
```

[Jonas\\_compteur::paint.](#)

#### Parameters

<i>painter</i>	
----------------	--

dessine les différents éléments du compteur Definition des constantes

Active l'antialiasing pour les formes géométriques

Définition des différentes options graphiques

Dessine les carrés qui vont contenir les différents éléments du cadrant

Définition des gradients

Dessine les gradients en arrière plan

Dessine les graduations du cadrant

Dessine le cadrant intérieur

Dessine le cadrant extérieur

Dessine le texte autour du cadrant

Dessine l'aiguille et met à jour sa position à chaque update

Dessine le cercle à la base de l'aiguille

### 6.58.2.3 speedToAngle()

```
float Jonas_compteur::speedToAngle (
    float speed )
```

[Jonas\\_compteur::speedToAngle.](#)

#### Parameters

<i>speed</i>	
--------------	--

#### Returns

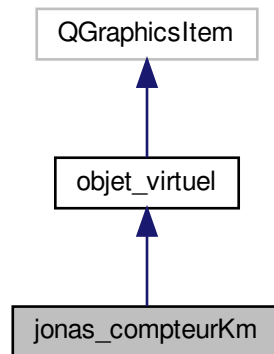
retourne l'angle (en radians) correspondant à la vitesse en entrée

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

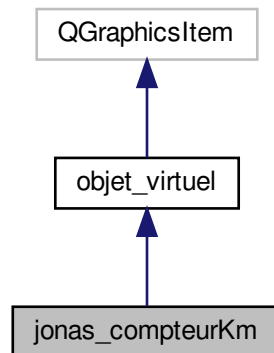
- serveur/Jonas/jonas\_compteur.h
- serveur/Jonas/jonas\_compteur.cpp

## 6.59 jonas\_compteurKm Class Reference

Inheritance diagram for jonas\_compteurKm:



Collaboration diagram for jonas\_compteurKm:



### Public Member Functions

- [jonas\\_compteurKm \(\)](#)  
[jonas\\_compteurKm::jonas\\_compteurKm](#)
- [QRectF boundingRect \(\)](#) const  
[jonas\\_compteurKm::boundingRect](#)
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[jonas\\_compteurKm::paint](#)

## Additional Inherited Members

### 6.59.1 Constructor & Destructor Documentation

#### 6.59.1.1 jonas\_compteurKm()

```
jonas_compteurKm::jonas_compteurKm ( )
```

[jonas\\_compteurKm::jonas\\_compteurKm](#)

il n'y a qu'un seul constructeur par défaut

### 6.59.2 Member Function Documentation

#### 6.59.2.1 boundingRect()

```
QRectF jonas_compteurKm::boundingRect ( ) const
```

[jonas\\_compteurKm::boundingRect](#)

##### Returns

retourne un rectangle qui encadre l'objet

#### 6.59.2.2 paint()

```
void jonas_compteurKm::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[jonas\\_compteurKm::paint](#)

##### Parameters

<i>painter</i>	
----------------	--

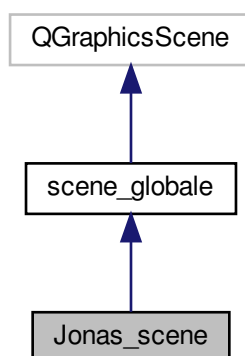
dessine un écran affichant les km parcourus

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

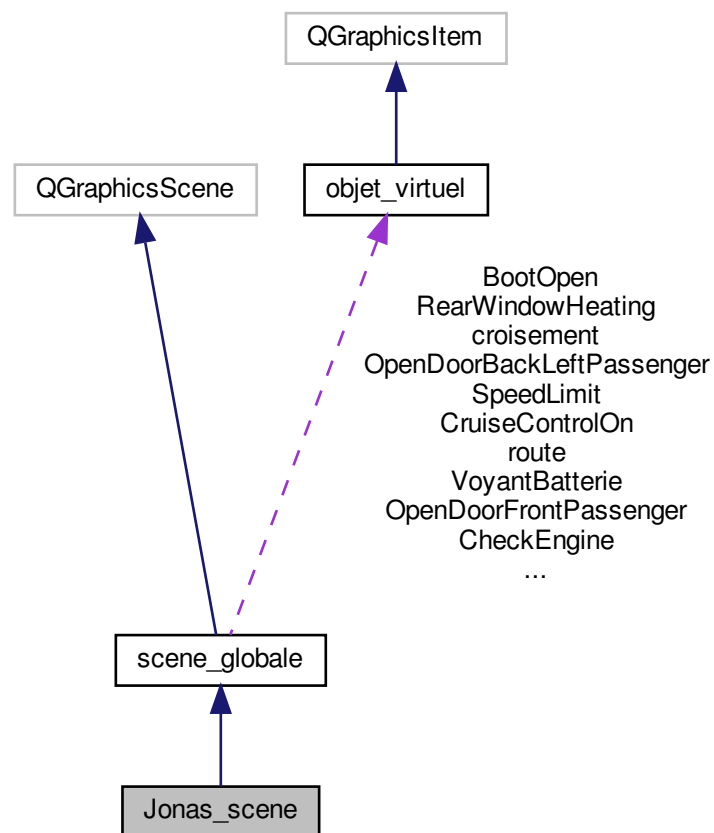
- serveur/Jonas/jonas\_compteurkm.h
- serveur/Jonas/jonas\_compteurkm.cpp

## 6.60 Jonas\_scene Class Reference

Inheritance diagram for Jonas\_scene:



Collaboration diagram for Jonas\_scene:



## Public Member Functions

- **Jonas\_scene** ([scene\\_globale](#) \*parent=nullptr)

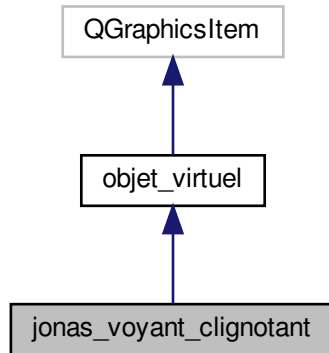
## Additional Inherited Members

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

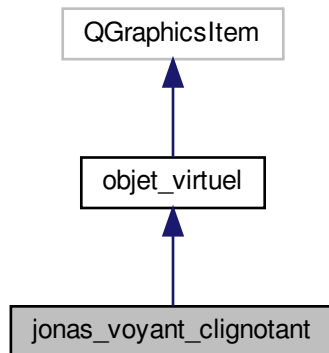
- serveur/Jonas/jonas\_scene.h
- serveur/Jonas/jonas\_scene.cpp

## 6.61 jonas\_voyant\_clignotant Class Reference

Inheritance diagram for jonas\_voyant\_clignotant:



Collaboration diagram for jonas\_voyant\_clignotant:



### Public Member Functions

- [jonas\\_voyant\\_clignotant \(\)](#)  
*[jonas\\_voyant\\_clignotant::jonas\\_voyant\\_clignotant](#)*
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*[jonas\\_voyant\\_clignotant::paint](#)*
- QRectF [boundingRect](#) () const  
*[jonas\\_voyant\\_clignotant::boundingRect](#)*
- void [updateTurn](#) ()  
*[jonas\\_voyant\\_clignotant::updateTurn](#)*



## Additional Inherited Members

### 6.61.1 Constructor & Destructor Documentation

#### 6.61.1.1 jonas\_voyant\_clignotant()

```
jonas_voyant_clignotant::jonas_voyant_clignotant ( )
```

[jonas\\_voyant\\_clignotant::jonas\\_voyant\\_clignotant](#)

constructeur par défaut

### 6.61.2 Member Function Documentation

#### 6.61.2.1 boundingRect()

```
QRectF jonas_voyant_clignotant::boundingRect ( ) const
```

[jonas\\_voyant\\_clignotant::boundingRect](#)

##### Returns

retourne un rectangle qui encadre l'objet

#### 6.61.2.2 paint()

```
void jonas_voyant_clignotant::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[jonas\\_voyant\\_clignotant::paint](#)

##### Parameters

<i>painter</i>	
----------------	--

dessine les clignotants

### 6.61.2.3 updateTurn()

```
void jonas_voyant_clignotant::updateTurn ( )
```

[jonas\\_voyant\\_clignotant::updateTurn](#)

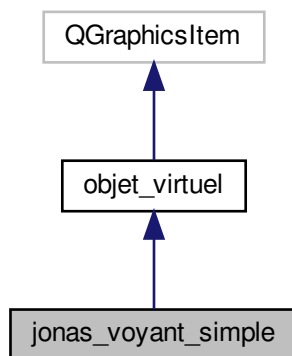
permet d'alterner l'affichage des clignotants

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

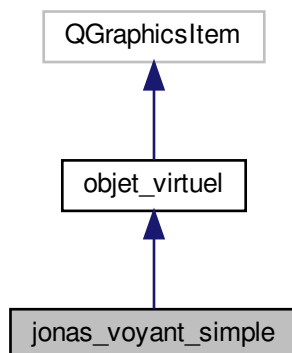
- serveur/Jonas/jonas\_voyant\_clignotant.h
- serveur/Jonas/jonas\_voyant\_clignotant.cpp

## 6.62 jonas\_voyant\_simple Class Reference

Inheritance diagram for jonas\_voyant\_simple:



Collaboration diagram for jonas\_voyant\_simple:



## Public Member Functions

- **jonas\_voyant\_simple** (QString, int, int, int)
- QRectF **boundingRect** () const override  
[jonas\\_voyant\\_simple::boundingRect](#)
- void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override  
[jonas\\_voyant\\_simple::paint](#)
- int **getValue** ()

## Additional Inherited Members

### 6.62.1 Member Function Documentation

#### 6.62.1.1 boundingRect()

```
QRectF jonas_voyant_simple::boundingRect ( ) const [override]
```

[jonas\\_voyant\\_simple::boundingRect](#)

#### Returns

retourne un rectangle qui encadre l'objet

#### 6.62.1.2 paint()

```
void jonas_voyant_simple::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * ,
    QWidget * ) [override]
```

[jonas\\_voyant\\_simple::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

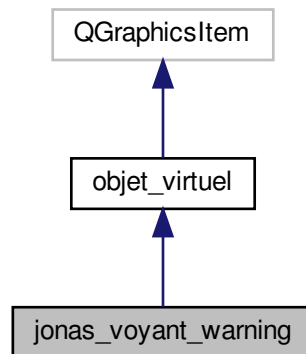
dessine les images (QPixmap) des voyants simples

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

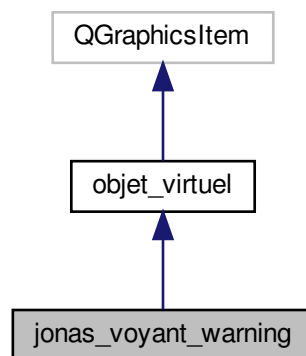
- serveur/Jonas/jonas\_voyant\_simple.h
- serveur/Jonas/jonas\_voyant\_simple.cpp

## 6.63 jonas\_voyant\_warning Class Reference

Inheritance diagram for jonas\_voyant\_warning:



Collaboration diagram for jonas\_voyant\_warning:



### Public Member Functions

- [jonas\\_voyant\\_warning\(\)](#)  
*[jonas\\_voyant\\_warning::jonas\\_voyant\\_warning](#)*
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*[jonas\\_voyant\\_warning::paint](#)*
- QRectF [boundingRect](#) () const  
*[jonas\\_voyant\\_warning::boundingRect](#)*

## Additional Inherited Members

### 6.63.1 Member Function Documentation

#### 6.63.1.1 boundingRect()

```
QRectF jonas_voyant_warning::boundingRect ( ) const
```

[jonas\\_voyant\\_warning::boundingRect](#)

#### Returns

retourne un rectangle qui encadre l'objet

#### 6.63.1.2 paint()

```
void jonas_voyant_warning::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[jonas\\_voyant\\_warning::paint](#)

#### Parameters

<i>painter</i>	
----------------	--

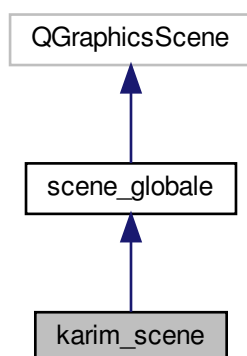
dessine l'icône des warning

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

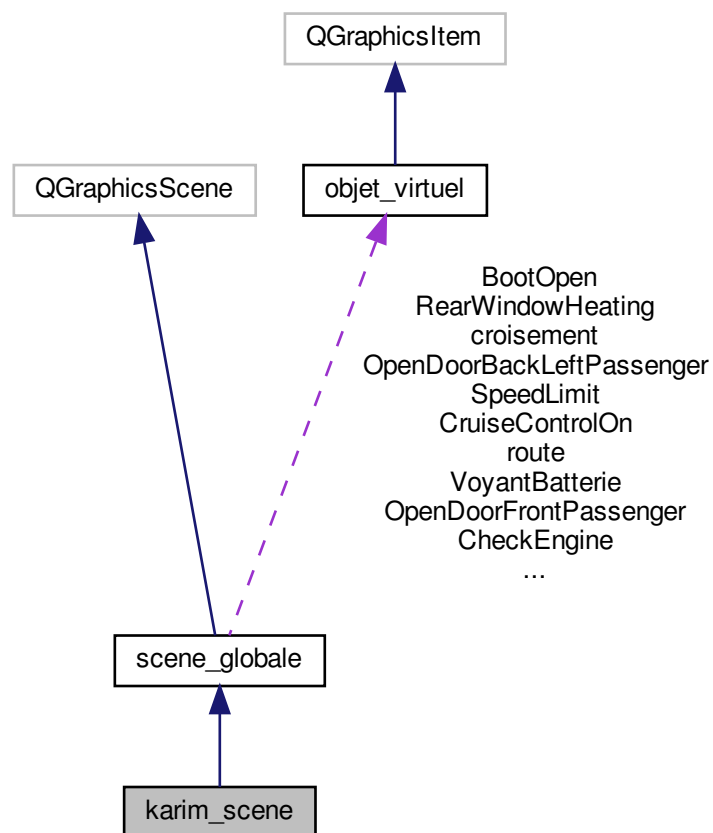
- serveur/Jonas/jonas\_voyant\_warning.h
- serveur/Jonas/jonas\_voyant\_warning.cpp

## 6.64 karim\_scene Class Reference

Inheritance diagram for karim\_scene:



Collaboration diagram for karim\_scene:



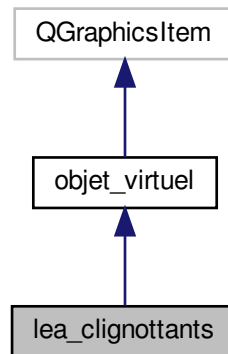
### Additional Inherited Members

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

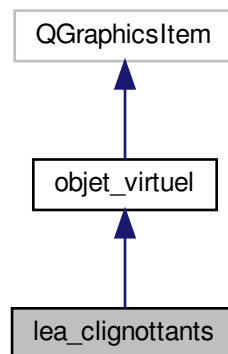
- `serveur/Karim/karim_scene.h`
- `serveur/Karim/karim_scene.cpp`

## 6.65 lea\_clignottants Class Reference

Inheritance diagram for lea\_clignottants:



Collaboration diagram for lea\_clignottants:



### Public Member Functions

- [lea\\_clignottants](#) ()  
*lea\_clignottants::lea\_clignottants* Constructeur de la classe
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const
- void [MAJ](#) ()  
*lea\_clignottants::MAJ* Fonction de mise à jour de l'affichage.



## Public Attributes

- float **cligno**

## Additional Inherited Members

### 6.65.1 Constructor & Destructor Documentation

#### 6.65.1.1 lea\_clignottants()

```
lea_clignottants::lea_clignottants ( )
```

[lea\\_clignottants::lea\\_clignottants](#) Constructeur de la classe

Constructeur de la classe initialisant la variable value de la classe mère, et la variable cligno qui permet de gérer l'affichage alternatif des voyants.

### 6.65.2 Member Function Documentation

#### 6.65.2.1 MAJ()

```
void lea_clignottants::MAJ ( )
```

[lea\\_clignottants::MAJ](#) Fonction de mise à jour de l'affichage.

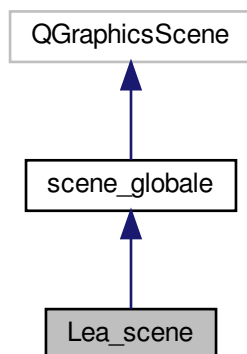
La valeur cligno controle l'opacité du painter, permettant de le rendre visible ou non. La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

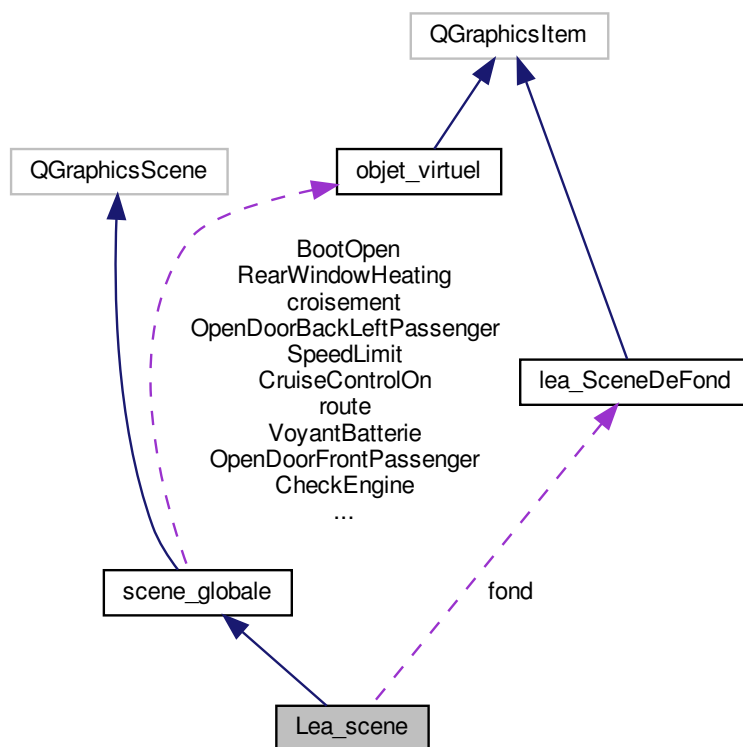
- serveur/Lea/lea\_clignottants.h
- serveur/Lea/[lea\\_clignottants.cpp](#)

## 6.66 Lea\_scene Class Reference

Inheritance diagram for Lea\_scene:



Collaboration diagram for Lea\_scene:



## Public Member Functions

- **Lea\_scene** ([scene\\_globale](#) \*parent=nullptr)

## Public Attributes

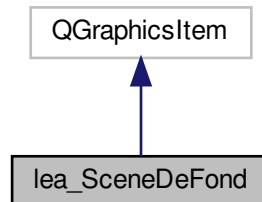
- [lea\\_SceneDeFond](#) \* **fond**

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

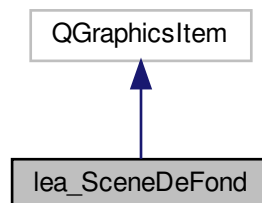
- serveur/Lea/lea\_scene.h
- serveur/Lea/[lea\\_scene.cpp](#)

## 6.67 lea\_SceneDeFond Class Reference

Inheritance diagram for lea\_SceneDeFond:



Collaboration diagram for lea\_SceneDeFond:



### Public Member Functions

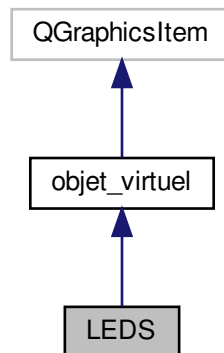
- **lea\_SceneDeFond** (QGraphicsItem \*parent=nullptr)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const

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

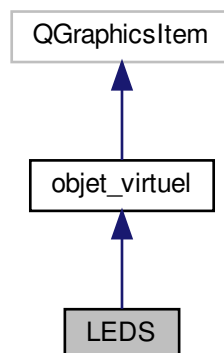
- serveur/Lea/lea\_scenedefond.h
- serveur/Lea/lea\_scenedefond.cpp

## 6.68 LEDS Class Reference

Inheritance diagram for LEDS:



Collaboration diagram for LEDS:



### Public Member Functions

- **LEDS** (int, int, QString iconPathString, int red=235, int green=0, int blue=0, int icon\_size=30)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const

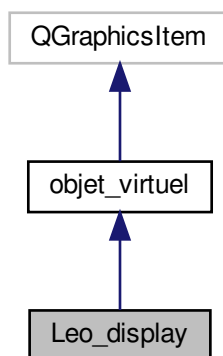
### Additional Inherited Members

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

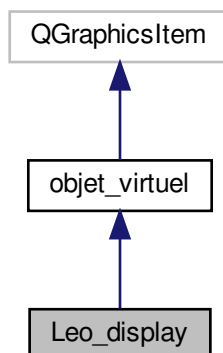
- serveur/Loto/leds.h
- serveur/Loto/leds.cpp

## 6.69 Leo\_display Class Reference

Inheritance diagram for Leo\_display:



Collaboration diagram for Leo\_display:



## Public Member Functions

- **Leo\_display** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const override
- void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override

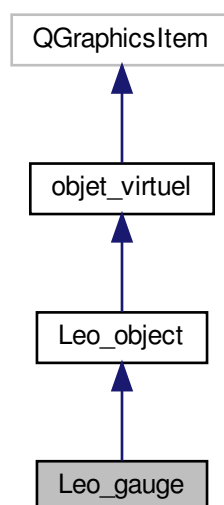
## Additional Inherited Members

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

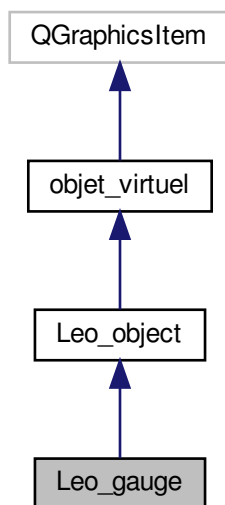
- serveur/Leo/leo\_display.h
- serveur/Leo/leo\_display.cpp

## 6.70 Leo\_gauge Class Reference

Inheritance diagram for Leo\_gauge:



Collaboration diagram for Leo\_gauge:



### Public Types

- enum **Data** {  
    **MAX, DIVS, SUBDIVS, STARTANGLE,**  
    **ARCLENGTH, WIDTH, FONT }**

### Public Member Functions

- **Leo\_gauge** (const QRectF boundingRect, QGraphicsItem \*parent=nullptr)
- void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override

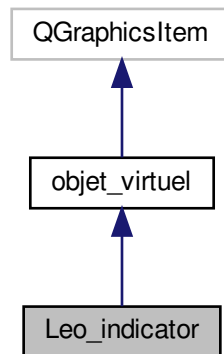
### Additional Inherited Members

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

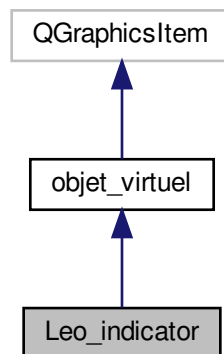
- serveur/Leo/leo\_gauge.h
- serveur/Leo/leo\_gauge.cpp

## 6.71 Leo\_indicator Class Reference

Inheritance diagram for Leo\_indicator:



Collaboration diagram for Leo\_indicator:



### Public Member Functions

- **Leo\_indicator** (QColor colorOff, QColor colorOn, [objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const override
- void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override

### Public Attributes

- QColor **ColorOff**
- QColor **ColorOn**



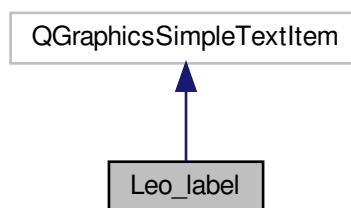
### Additional Inherited Members

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

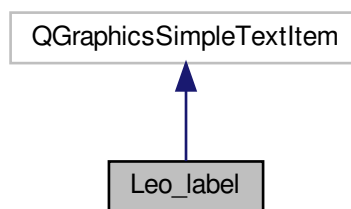
- serveur/Leo/leo\_indicator.h
- serveur/Leo/leo\_indicator.cpp

## 6.72 Leo\_label Class Reference

Inheritance diagram for Leo\_label:



Collaboration diagram for Leo\_label:



### Public Member Functions

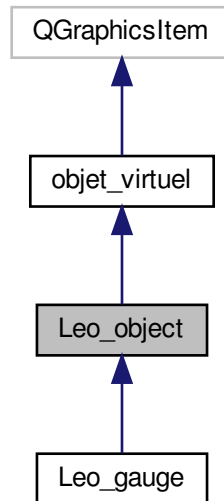
- **Leo\_label** (const QString &text, const QRectF boundingRect, QGraphicsItem \*parent=nullptr)
- QRectF **boundingRect** () const override
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*, QWidget \*) override

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

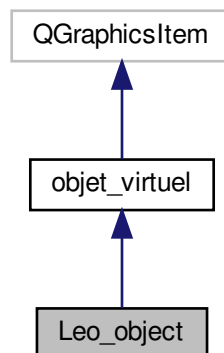
- serveur/Leo/leo\_label.h
- serveur/Leo/leo\_label.cpp

## 6.73 Leo\_object Class Reference

Inheritance diagram for Leo\_object:



Collaboration diagram for Leo\_object:



### Public Member Functions

- **Leo\_object** (const QRectF boundingRect, QGraphicsItem \*parent=nullptr)
- virtual QRectF **boundingRect** () const final override
- virtual void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override=0

### Protected Attributes

- QRectF **mBoundingRect**

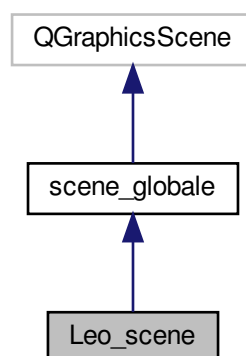
### Additional Inherited Members

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

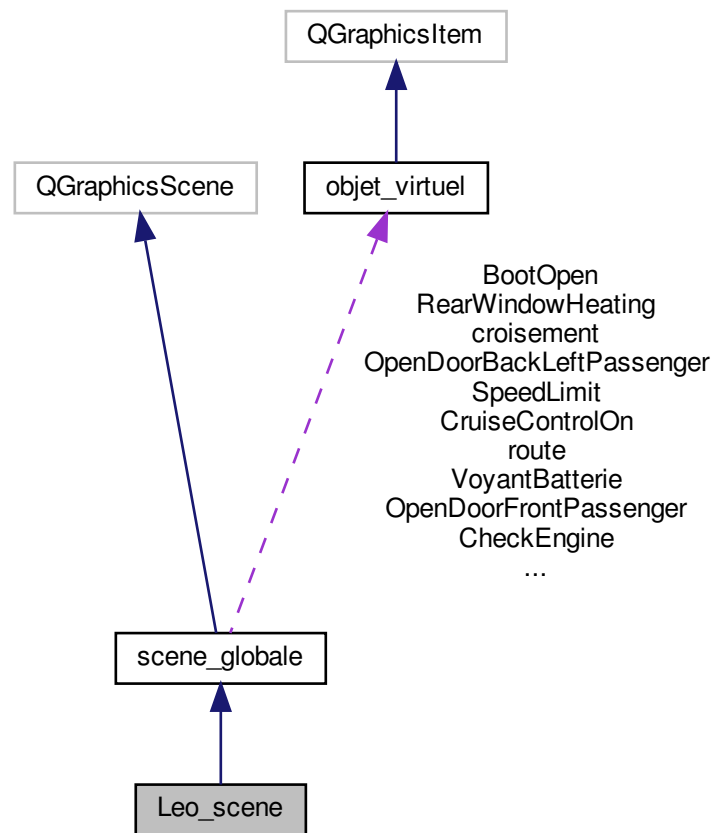
- serveur/Leo/leo\_object.h
- serveur/Leo/leo\_object.cpp

## 6.74 Leo\_scene Class Reference

Inheritance diagram for Leo\_scene:



Collaboration diagram for Leo\_scene:



## Public Member Functions

- **Leo\_scene** ([scene\\_globale](#) \*parent=nullptr)

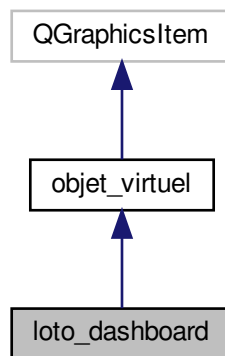
## Additional Inherited Members

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

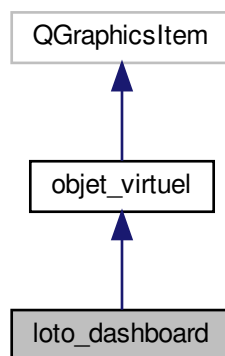
- serveur/Leo/leo\_scene.h
- serveur/Leo/leo\_scene.cpp

## 6.75 loto\_dashboard Class Reference

Inheritance diagram for loto\_dashboard:



Collaboration diagram for loto\_dashboard:



### Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- double **getSpeedAngle** (double speedValue)

### Public Attributes

- double **speedValue**
- const double **pi** =3.142

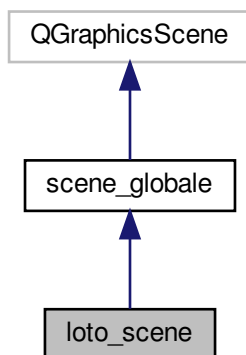
## Additional Inherited Members

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

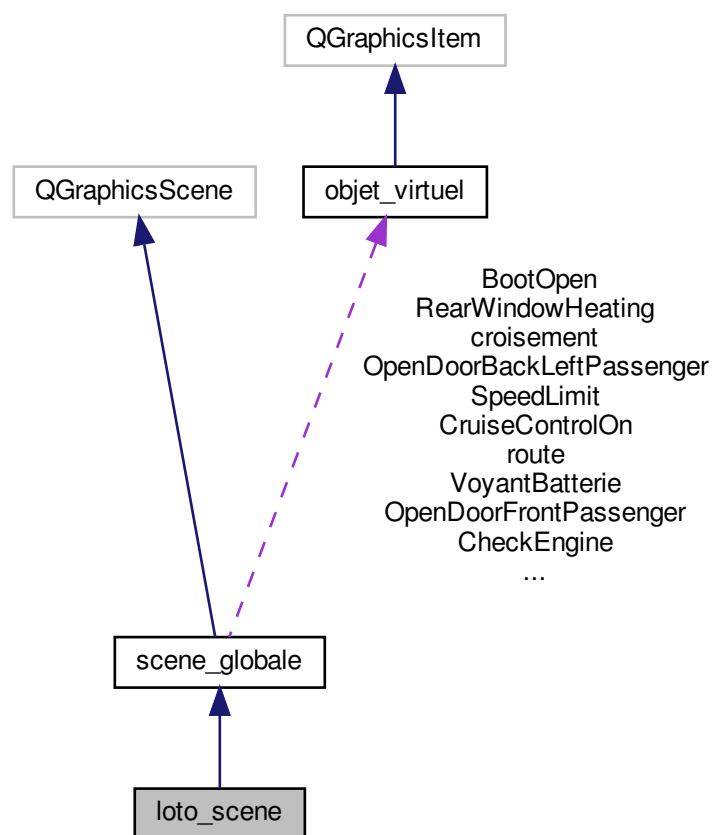
- serveur/Loto/loto\_dashboard.h
- serveur/Loto/loto\_dashboard.cpp

## 6.76 loto\_scene Class Reference

Inheritance diagram for loto\_scene:



Collaboration diagram for loto\_scene:



## Public Member Functions

- **loto\_scene** (**scene\_globale** \*parent=nullptr)

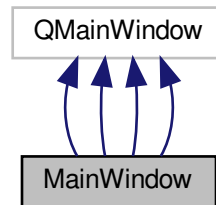
### Additional Inherited Members

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

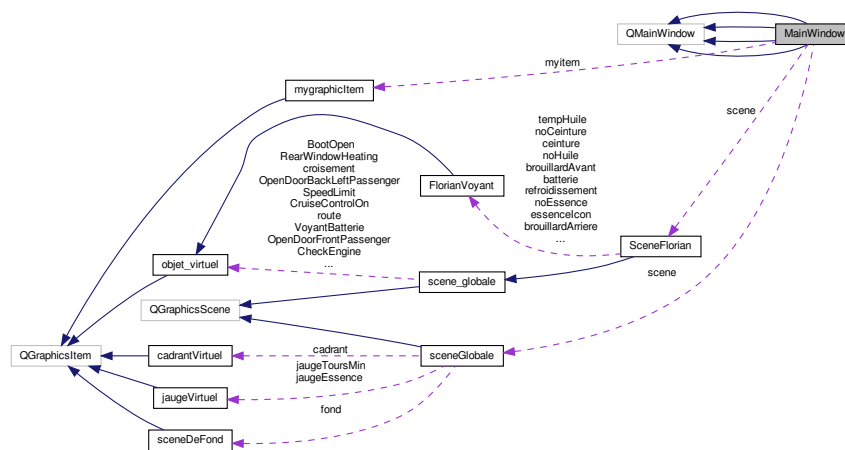
- serveur/Loto/loto\_scene.h
- serveur/Loto/loto\_scene.cpp

## 6.77 MainWindow Class Reference

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



### Public Slots

- void **Scene** ()

### Public Member Functions

- [MainWindow](#) (QWidget \*parent=0)
- **MainWindow** (QWidget \*parent=0)
- void **cli** ()
- **MainWindow** (QWidget \*parent=nullptr)
- **MainWindow** (QWidget \*parent=0)
- void [acceleration](#) (int)  
*MainWindow::acceleration* Simulation d'une accélération.
- void **simulation** (int)



## Public Attributes

- [SceneFlorian](#) \* **scene**
- [sceneGlobale](#) \* **scene**
- QGraphicsView \* **view** = new QGraphicsView()
- QGraphicsScene \* **scene** = new QGraphicsScene ()
- [mygraphicItem](#) \* **myitem** = new [mygraphicItem](#) ()

## 6.77.1 Constructor & Destructor Documentation

### 6.77.1.1 MainWindow()

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

La scène par défaut est

## 6.77.2 Member Function Documentation

### 6.77.2.1 acceleration()

```
void MainWindow::acceleration (
    int time )
```

[MainWindow::acceleration](#) Simulation d'une accélération.

Fonction simulant une accélération linéaire avec changements de rapports de vitesse

#### Parameters

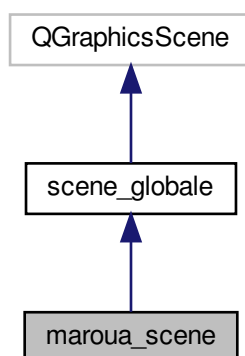
<i>time</i>	Durée de la simulation
-------------	------------------------

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

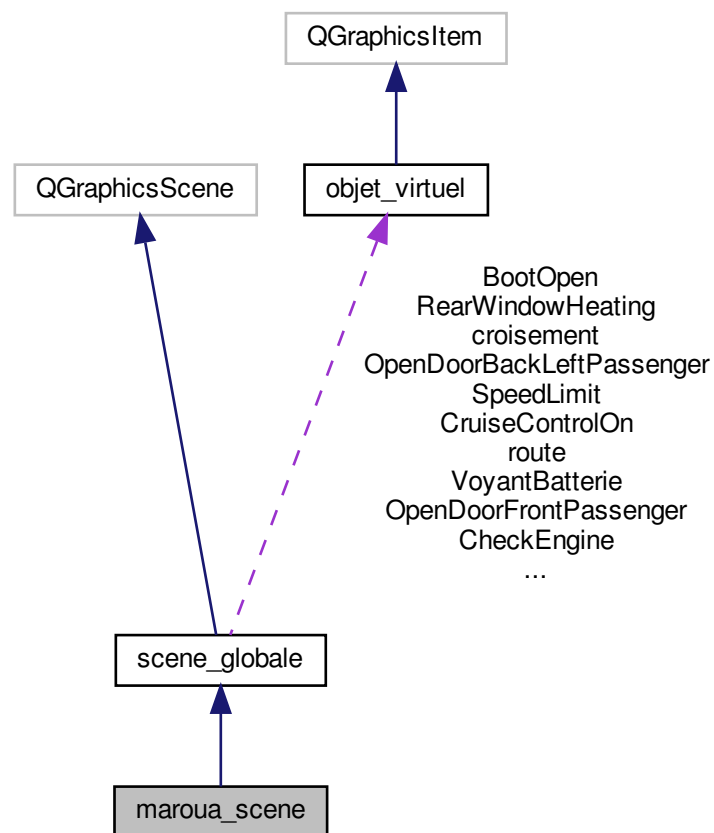
- serveur/Florian/mainwindow.h
- serveur/Florian/mainwindow.cpp

## 6.78 maroua\_scene Class Reference

Inheritance diagram for maroua\_scene:



Collaboration diagram for maroua\_scene:



## Public Member Functions

- `maroua_scene` ([scene\\_globale](#) \*parent=nullptr)

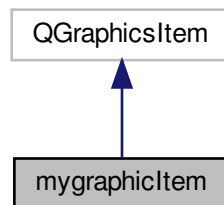
## Additional Inherited Members

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

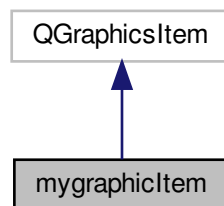
- `serveur/Maroua/maroua_scene.h`
- `serveur/Maroua/maroua_scene.cpp`

## 6.79 mygraphicItem Class Reference

Inheritance diagram for mygraphicItem:



Collaboration diagram for mygraphicItem:



### Public Member Functions

- **mygraphicItem** (`QGraphicsItem *parent=nullptr`)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

### Public Attributes

- double **xpos**
- double **ypos**
- double **xpos2**
- double **ypos2**
- double **xc**
- double **yc**
- double **xc2**
- double **yc2**
- int **v1**

- const double **pi** =3.14159265359

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

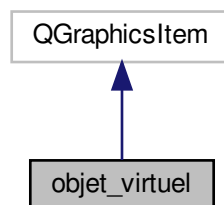
- serveur/Lea/mygraphicitem.h
- serveur/Lea/mygraphicitem.cpp

## 6.80 objet\_virtuel Class Reference

Inherits QGraphicsItem.

Inherited by [afficheKmHenri](#), [ArrowOilInna](#), [ArrowOilTInna](#), [ArrowSpeedometerInna](#), [ArrowTachometerInna](#), [cadran](#), [CadranEss](#), [CadrantFlorian](#), [cadrantHenri](#), [CadranTourParMin](#), [cadranttour](#), [cadrantvitesse](#), [CadranVitesse](#), [CarSpeedDial](#), [clignot](#), [Compteur\\_kilometre\\_lea](#), [compteurKmlInna](#), [EngineTInna](#), [FlorianClignotant](#), [FlorianCompteurKm](#), [FlorianVoyant](#), [fuel\\_guage](#), [FuelGauge\\_lea](#), [FuelLevelInna](#), [Harc](#), [horloge\\_lea](#), [hugo\\_boite\\_vitesse](#), [hugo\\_Compteur](#), [hugo\\_ecran](#), [hugo\\_MyGraphicsitem](#), [hugo\\_voyant\\_warning](#), [hugo\\_voyants\\_clignotant](#), [hugo\\_voyants\\_simples](#), [IconInna](#), [jauge\\_temperature\\_lea](#), [jaugeClignotantHenri](#), [jaugeEssenceHenri](#), [jaugeTemperatureHenri](#), [jaugeToursMinuteHenri](#), [Jonas\\_compteur](#), [jonas\\_compteurKm](#), [jonas\\_voyant\\_clignotant](#), [jonas\\_voyant\\_simple](#), [jonas\\_voyant\\_warning](#), [lea\\_clignottants](#), [LEDS](#), [Leo\\_display](#), [Leo\\_indicator](#), [Leo\\_object](#), [loto\\_dashboard](#), [onoff](#), [onoffpaintHenri](#), [portesHenri](#), [speedometer\\_Lea](#), [TachometerGauge\\_Lea](#), [tachometre](#), [tempGauge](#), [voyant\\_Lea](#), [Voyants](#), [voyants\\_warning](#), [VoyantsMaroua](#), [voyantsYoucef](#), [warning\\_lea](#), and [warninghenri](#).

Collaboration diagram for objet\_virtuel:



### Public Member Functions

- [objet\\_virtuel](#) (`QGraphicsItem *parent=nullptr`)
- float [getValue](#) () const  
*Fonction renvoyant la variable value.*
- void [setValue](#) (float set\_value)  
*[objet\\_virtuel::setValue](#). Permet de modifier la valeur de la variable value*
- int [getValueMax](#) () const  
*[objet\\_virtuel::getValueMax](#) Fonction renvoyant la valeur de valueMax*

## Public Attributes

- QString **styleTexte**

## Protected Attributes

- float **value**
- int **valueMax**

## 6.80.1 Constructor & Destructor Documentation

### 6.80.1.1 objet\_virtuel()

```
objet_virtuel::objet_virtuel (
    QGraphicsItem * parent = nullptr )
```

Classe dérivée de QGraphicsItem. Va servir de classe mère pour tous les objets du dashboardE. Elle contient un float Value, et un float valueMax, tous deux en variables protégées

## 6.80.2 Member Function Documentation

### 6.80.2.1 getValue()

```
float objet_virtuel::getValue ( ) const
```

Fonction renvoyant la variable value.

**Returns**

### 6.80.2.2 getValueMax()

```
int objet_virtuel::getValueMax ( ) const
```

[objet\\_virtuel::getValueMax](#) Fonction renvoyant la valeur de valueMax

**Returns**

### 6.80.2.3 setValue()

```
void objet_virtuel::setValue (
    float set_value )
```

[objet\\_virtuel::setValue](#). Permet de modifier la valeur de la variable value

## Parameters

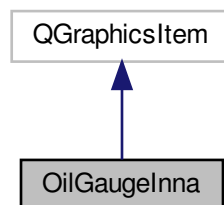
<i>set_value</i>	valeur à attribuer à value
------------------	----------------------------

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

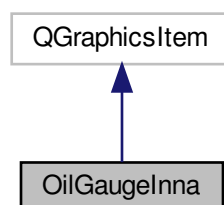
- serveur/objet\_virtuel.h
- serveur/objet\_virtuel.cpp

## 6.81 OilGaugelInna Class Reference

Inheritance diagram for OilGaugelInna:



Collaboration diagram for OilGaugelInna:



### Public Member Functions

- **OilGaugelInna** (QGraphicsItem \*parent=nullptr)
- **OilGaugelInna** (int TMAX, int LMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[FuelEngineInna::paint.](#)

## Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **tmax**
- int **t**
- int **lmax**
- double **l**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- double **k**
- int **dx**
- int **dy**

## 6.81.1 Member Function Documentation

### 6.81.1.1 paint()

```
void OilGaugeInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[FuelEngineInna::paint.](#)

#### Parameters

<i>painter</i>	
----------------	--

La fonction paint va permettre de réaliser un affichage des objets fixes de la représentation des jauges de niveau et de température d'huile. Remplissage du fond et dessin de l'arche encadrante

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner une arche encadrante avec un gradient de couleur.

Affichage des traits et du texte pour la jauge de la température d'huile.

Affichage des traits et du texte pour le niveau d'huile.

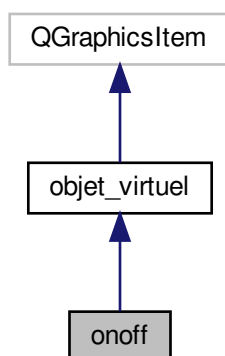
The documentation for this class was generated from the following files:

- serveur/Inna/oilgaugeinna.h
- serveur/Inna/oilgaugeinna.cpp

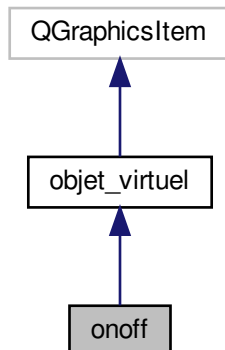


## 6.82 onoff Class Reference

Inheritance diagram for onoff:



Collaboration diagram for onoff:



### Public Member Functions

- **onoff** ([objet\\_virtuel](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

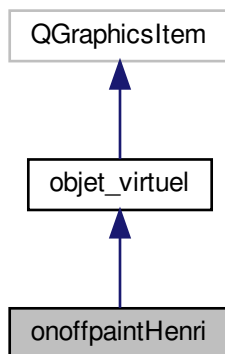
### Additional Inherited Members

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

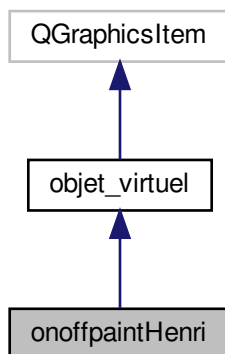
- serveur/Henri/onoff.h
- serveur/Henri/onoff.cpp

## 6.83 onoffpaintHenri Class Reference

Inheritance diagram for onoffpaintHenri:



Collaboration diagram for onoffpaintHenri:



### Public Member Functions

- **onoffpaintHenri** (int xx, int yy, int ww, int hh, QPixmap pixm)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- int **x**
- int **y**
- int **w**
- int **h**
- QPixmap **pix**

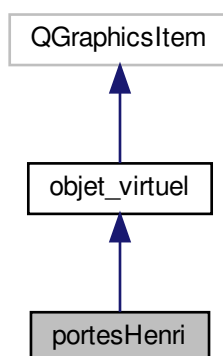
### Additional Inherited Members

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

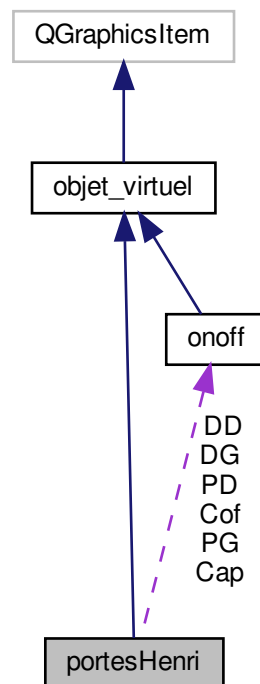
- serveur/Henri/onoffpainthenri.h
- serveur/Henri/onoffpainthenri.cpp

## 6.84 portesHenri Class Reference

Inheritance diagram for portesHenri:



Collaboration diagram for portesHenri:



### Public Member Functions

- **portesHenri** (`objet_virtuel` \*parent=nullptr)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter` \*painter, const `QStyleOptionGraphicsItem` \*option, `QWidget` \*widget)

### Public Attributes

- `onoff` \* **DD**
- `onoff` \* **DG**
- `onoff` \* **PG**
- `onoff` \* **PD**
- `onoff` \* **Cap**
- `onoff` \* **Cof**

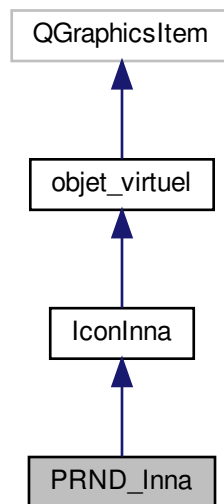
### Additional Inherited Members

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

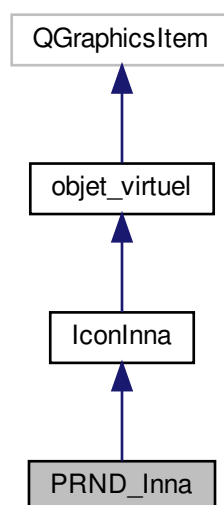
- serveur/Henri/porteshenri.h
- serveur/Henri/porteshenri.cpp

## 6.85 PRND\_Inna Class Reference

Inheritance diagram for PRND\_Inna:



Collaboration diagram for PRND\_Inna:



## Public Member Functions

- **PRND\_Inna** ([IconInna](#) \*parent=nullptr)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
[PRND\\_Inna::paint](#).

## Additional Inherited Members

### 6.85.1 Member Function Documentation

#### 6.85.1.1 paint()

```
void PRND_Inna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[PRND\\_Inna::paint](#).

#### Parameters

<i>painter</i>	
----------------	--

La fonction permet de l'afficher l'image correspondante au mode de la transmission engagée.

Le choix est fait en fonction de la valeur *value* :

1 - Park;

2 - Reverse;

3 - Neutral;

4 - Drive;

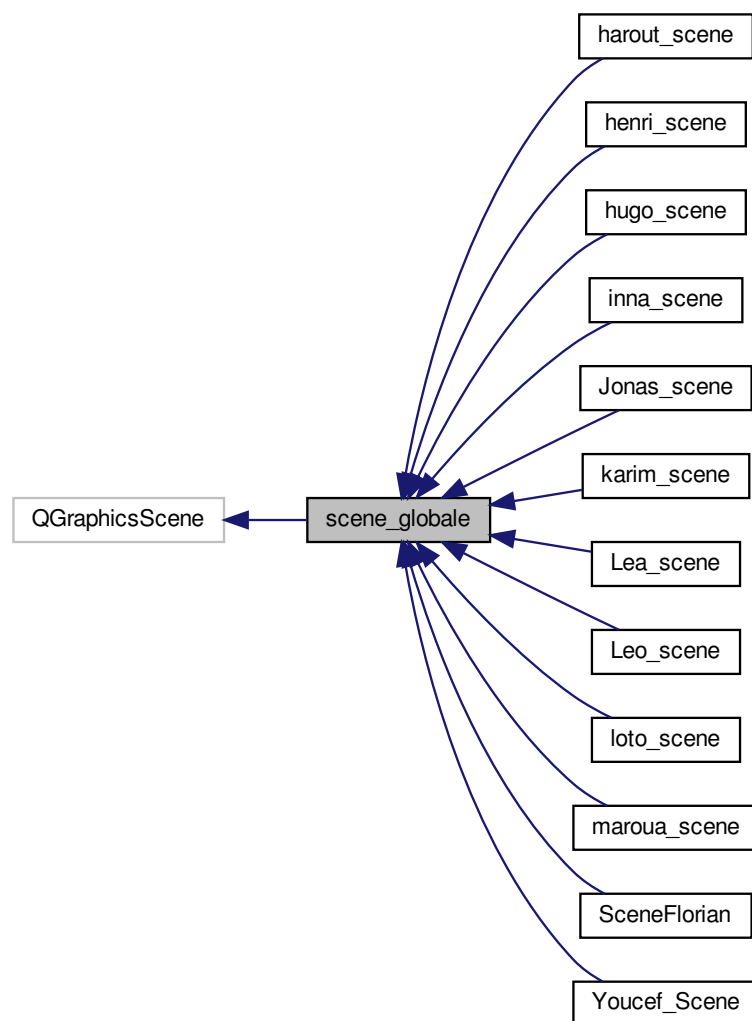
Dans autres cas le voyant "\_probleme avec la boite de vitesse\_" et affiché.

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

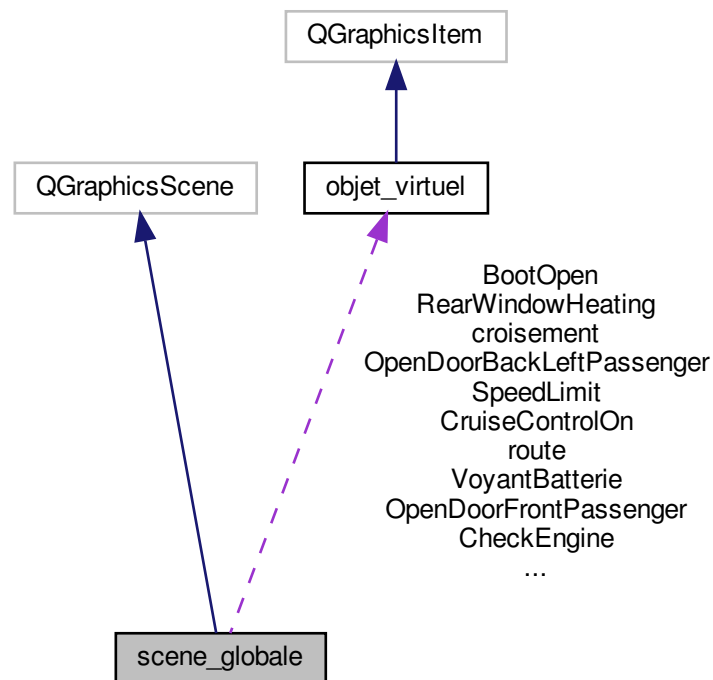
- serveur/Inna/prnd\_inna.h
- serveur/Inna/[prnd\\_inna.cpp](#)

## 6.86 scene\_globale Class Reference

Inheritance diagram for scene\_globale:



Collaboration diagram for scene\_globale:



## Public Member Functions

- `scene_globale` (`QGraphicsScene *parent=nullptr`)

## Public Attributes

- `objet_virtuel` \* **Vitesse**
- `objet_virtuel` \* **Essence**
- `objet_virtuel` \* **CompteTours**
- `objet_virtuel` \* **jaugeTemperature**
- `objet_virtuel` \* **Clignotant**
- `objet_virtuel` \* **VoyantBatterie**
- `objet_virtuel` \* **position**
- `objet_virtuel` \* **croisement**
- `objet_virtuel` \* **route**
- `objet_virtuel` \* **warning**
- `objet_virtuel` \* **CompteurKm** = nullptr
- `objet_virtuel` \* **AdaptiveSuspensionDampers**
- `objet_virtuel` \* **AutomaticTransmissionMode**
- `objet_virtuel` \* **FrontAntifog**
- `objet_virtuel` \* **RearAntifog**
- `objet_virtuel` \* **SeatBelt**
- `objet_virtuel` \* **RearWindowHeating**



- [objet\\_virtuel](#) \* **CheckEngine**
- [objet\\_virtuel](#) \* **OpenDoorDriver**
- [objet\\_virtuel](#) \* **OpenDoorFrontPassenger**
- [objet\\_virtuel](#) \* **OpenDoorBackLeftPassenger**
- [objet\\_virtuel](#) \* **OpenDoorBackRightPassenger**
- [objet\\_virtuel](#) \* **AdaptiveCruiseControl**
- [objet\\_virtuel](#) \* **AirbagOn**
- [objet\\_virtuel](#) \* **BonnetOpen**
- [objet\\_virtuel](#) \* **BootOpen**
- [objet\\_virtuel](#) \* **CruiseControlOn**
- [objet\\_virtuel](#) \* **OilTemp**
- [objet\\_virtuel](#) \* **SpeedLimit**
- [objet\\_virtuel](#) \* **oilLevel**
- [objet\\_virtuel](#) \* **ABS**
- [objet\\_virtuel](#) \* **handbrake**

## 6.86.1 Constructor & Destructor Documentation

### 6.86.1.1 scene\_globale()

```
scene_globale::scene_globale (
    QGraphicsScene * parent = nullptr )
```

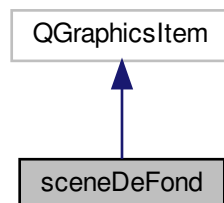
Classe dérivant de QGraphicsScene. Va servir de classe mère pour toutes les scènes, ce qui permettra de passer d'une scène à l'autre de façon dynamique. Tous les objets utilisés dans le dashboard seront définis dans "scène↔\_globale.h"

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

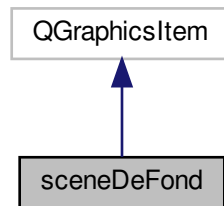
- serveur/scene\_globale.h
- serveur/scene\_globale.cpp

## 6.87 sceneDeFond Class Reference

Inheritance diagram for sceneDeFond:



Collaboration diagram for sceneDeFond:



### Public Member Functions

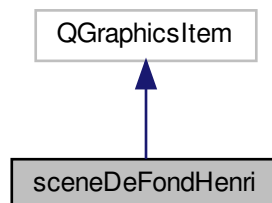
- **sceneDeFond** (`QGraphicsItem *parent=nullptr`)

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

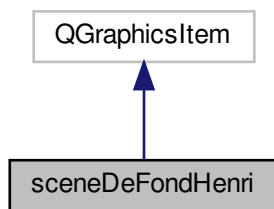
- `serveur/Florian/scenedefond.h`
- `serveur/Florian/scenedefond.cpp`

## 6.88 sceneDeFondHenri Class Reference

Inheritance diagram for sceneDeFondHenri:



Collaboration diagram for sceneDeFondHenri:



### Public Member Functions

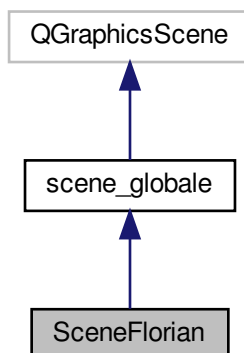
- **sceneDeFondHenri** (`QGraphicsItem *parent=nullptr`)
- void **paint** (`QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget`)
- `QRectF` **boundingRect** () const

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

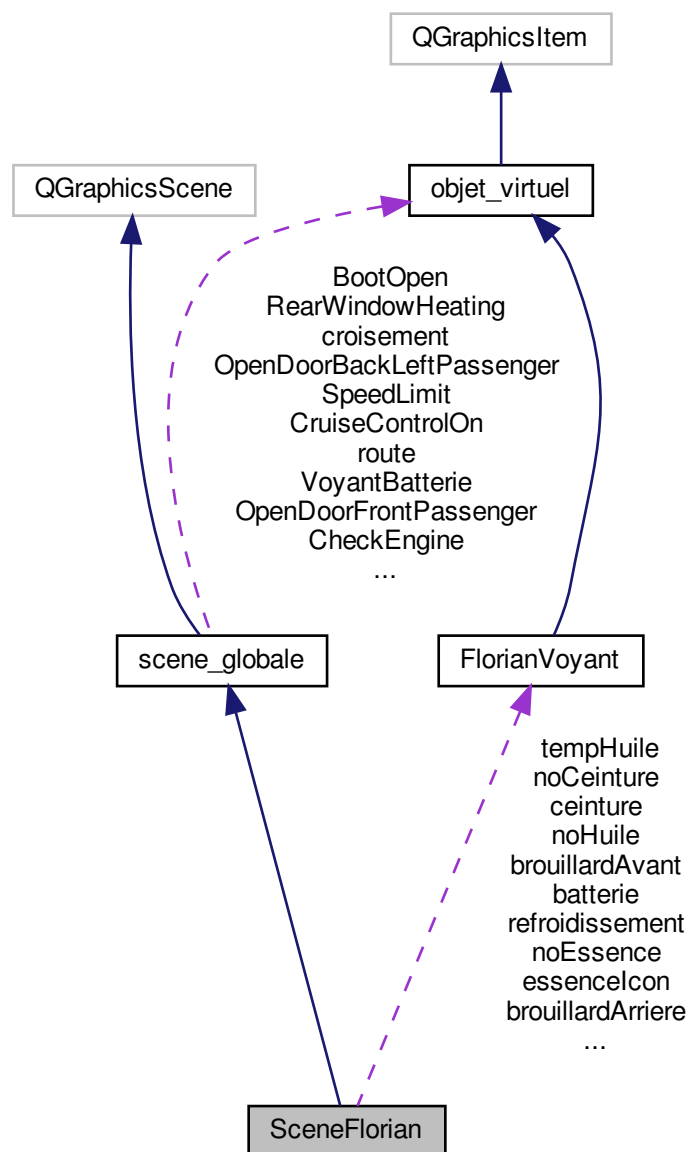
- `serveur/Henri/scenedefondhenri.h`
- `serveur/Henri/scenedefondhenri.cpp`

## 6.89 SceneFlorian Class Reference

Inheritance diagram for SceneFlorian:



Collaboration diagram for SceneFlorian:



## Public Member Functions

- **SceneFlorian** (QGraphicsScene \*parent=nullptr)

## Public Attributes

- **FlorianVoyant** \* **tempHuile**
- **FlorianVoyant** \* **essenceIcon**
- **FlorianVoyant** \* **brouillardAvant**

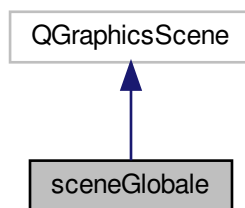
- [FlorianVoyant](#) \* **brouillardArriere**
- [FlorianVoyant](#) \* **noEssence**
- [FlorianVoyant](#) \* **noHuile**
- [FlorianVoyant](#) \* **refroidissement**
- [FlorianVoyant](#) \* **batterie**
- [FlorianVoyant](#) \* **ceinture**
- [FlorianVoyant](#) \* **noCeinture**

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

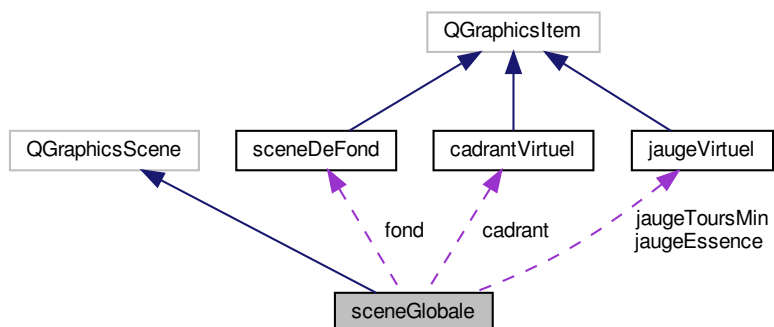
- serveur/Florian/sceneflorian.h
- serveur/Florian/sceneflorian.cpp

## 6.90 sceneGlobale Class Reference

Inheritance diagram for sceneGlobale:



Collaboration diagram for sceneGlobale:



## Public Member Functions

- **sceneGlobale** (QGraphicsScene \*parent=nullptr)

## Public Attributes

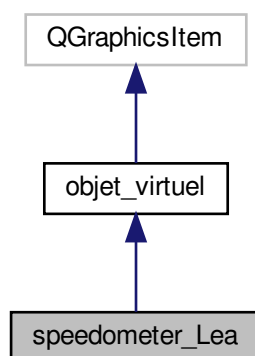
- [sceneDeFond](#) \* **fond**
- [cadrantVirtuel](#) \* **cadrant**
- [jaugeVirtuel](#) \* **jaugeEssence**
- [jaugeVirtuel](#) \* **jaugeToursMin**

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

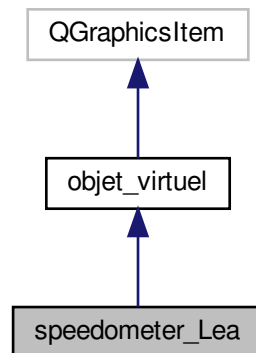
- serveur/Florian/sceneglobale.h
- serveur/Florian/sceneglobale.cpp

## 6.91 speedometer\_Lea Class Reference

Inheritance diagram for speedometer\_Lea:



Collaboration diagram for speedometer\_Lea:



## Public Member Functions

- [speedometer\\_Lea](#) (double, double, double, int, int, int)  
*[speedometer\\_Lea::speedometer\\_Lea](#) Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0*
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`  
*[speedometer\\_Lea::paint](#) Fonction permettant l'affichage du compteur*

## Public Attributes

- `const double pi =3.14159265359`
- `double x`
- `double y`
- `double r`
- `int angle_debut`
- `int span_angle`
- `int vitesse_max =270`
- `int v`
- `int nb_graduation =28`

## Additional Inherited Members

### 6.91.1 Constructor & Destructor Documentation

### 6.91.1.1 speedometer\_Lea()

```
speedometer_Lea::speedometer_Lea (
    double param_x,
    double param_y,
    double param_r,
    int param_start,
    int param_spanAngle,
    int param_vitMax )
```

[speedometer\\_Lea::speedometer\\_Lea](#) Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0

#### Parameters

<i>param_x</i>	position horizontale du centre du compteur
<i>param_y</i>	position verticale du centre du compteur
<i>param_r</i>	rayon du compteur et taille de l'aiguille
<i>param_start</i>	Angle de départ pour le tracé de l'arc de cercle
<i>param_end</i>	Angle de fin pour le tracé de l'arc de cercle
<i>param_spanAngle</i>	angle total du cadran
<i>param_param_vitMax</i>	determine la vitesse maximum jusqu'à laquelle va le cadran

## 6.91.2 Member Function Documentation

### 6.91.2.1 paint()

```
void speedometer_Lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

[speedometer\\_Lea::paint](#) Fonction permettant l'affichage du compteur

Cette fonction construit un compteur en plusieurs étapes : création du cadran, des graduations, ajout du texte sur les graduations, ajout de l'aiguille.

#### Parameters

<i>painter</i>	
----------------	--

calibration de "value"

La variable "value" donnée par la fonction [getValue\(\)](#) de la classe [objet\\_virtuel](#) est mise à 0 si la valeur entrée est inférieure à 0 et mise à "valueMax" si la valeur entrée est plus grande.

Création du cadran.



Le cadran est basé sur un arc de cercle dessiné à l'aide de la fonction `drawArc`. le cercle est dessiné avec un `QPen` a qui a été donné une `QBrush` contenant un gradient radial de couleur bleu.

Création des graduations du cadran.

Les graduations sont créées en utilisant la fonction `drawLine` et une boucle `if` pour changer la couleur des graduations à 50, 90 et 130km/h. Les graduations sont dessinées tous les 10km/h.

Ajout du texte sur les graduations

Le texte est positionné de la même façon que les graduations, tous les 20km/h, avec une translation de manière à correspondre le plus possible aux positions des graduations.

Création de l'aiguille.

Dans un premier temps, affichage de l'aiguille à l'aide de la fonction `drawConvexPolygon`. Puis affichage d'un petit cercle à la base de l'aiguille.

Création de l'affichage de la vitesse.

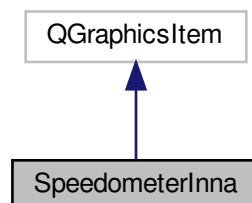
L'affichage de la vitesse se fait avec la fonction `drawText` et en utilisant la variable "value" donnée par la fonction `getValue()` de la classe `objet_virtuel`.

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

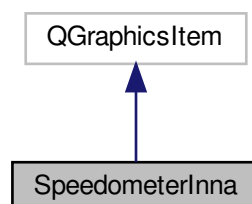
- `serveur/Lea/speedometer_lea.h`
- `serveur/Lea/speedometer_lea.cpp`

## 6.92 SpeedometerInna Class Reference

Inheritance diagram for `SpeedometerInna`:



Collaboration diagram for `SpeedometerInna`:



## Public Member Functions

- **SpeedometerInna** (QGraphicsItem \*parent=nullptr)
- **SpeedometerInna** (int VMAX, int ALPHA0, int ALPHAMAX)
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

## Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **vmax**
- int **v**
- QFont **font** = QFont("Chandas",12,QFont::Bold)
- double **k**
- int **dx**
- int **dy**

## 6.92.1 Member Function Documentation

### 6.92.1.1 [paint\(\)](#)

```
void SpeedometerInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

#### Dessin du circle encadrant

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner un circle encadrant avec un gradient de couleur.

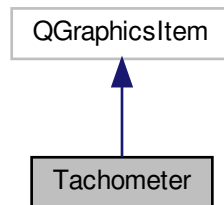
#### Dessin du texte et des traits

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

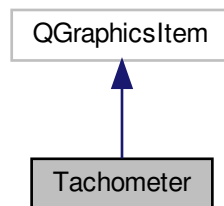
- serveur/Inna/speedometerinna.h
- serveur/Inna/[speedometerinna.cpp](#)

## 6.93 Tachometer Class Reference

Inheritance diagram for Tachometer:



Collaboration diagram for Tachometer:



### Public Member Functions

- **Tachometer** (**QGraphicsItem** \*parent=nullptr)
- **QRectF boundingRect** () const
- void **paint** (**QPainter** \*painter, const **QStyleOptionGraphicsItem** \*option, **QWidget** \*widget)

### Public Attributes

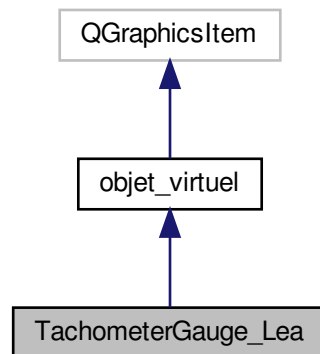
- float **pi** = 3.14159265359
- int **speed**

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

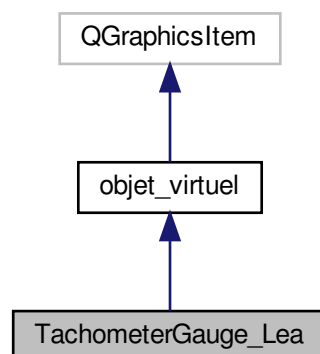
- serveur/Youcef/tachometer.h
- serveur/Youcef/tachometer.cpp

## 6.94 TachometerGauge\_Lea Class Reference

Inheritance diagram for TachometerGauge\_Lea:



Collaboration diagram for TachometerGauge\_Lea:



### Public Member Functions

- [TachometerGauge\\_Lea](#) (double, double, double, int, int, int, int)  
*[TachometerGauge\\_Lea::TachometerGauge\\_Lea](#) Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0.*
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)  
*Fonction permettant l'affichage du compteur.*

## Protected Attributes

- double **r**
- double **x**
- double **y**
- int **angle\_debut**
- int **nb\_graduation** =41
- int **span\_angle**
- int **vitesse\_max** =270
- int **v**
- const double **pi** =3.14159265359

## Additional Inherited Members

### 6.94.1 Constructor & Destructor Documentation

#### 6.94.1.1 TachometerGauge\_Lea()

```
TachometerGauge_Lea::TachometerGauge_Lea (
    double param_x,
    double param_y,
    double param_r,
    int param_start,
    int param_spanAngle,
    int param_graduation,
    int param_rpmMax )
```

[TachometerGauge\\_Lea::TachometerGauge\\_Lea](#) Constructeur de la classe, permet d'initialiser tous les paramètres ainsi que la valeur value de la classe mère à 0.

#### Parameters

<i>param_x</i>	position horizontale du centre du compteur
<i>param_y</i>	position verticale du centre du compteur
<i>param_r</i>	rayon et taille de l'aiguille
<i>param_start</i>	Angle de départ pour le tracé de l'arc de cercle
<i>param_end</i>	Angle de fin pour le tracé de l'arc de cercle
<i>param_spanAngle</i>	angle total de cadran
<i>param_graduation</i>	nombre de graduation voulue pour le cadran
<i>param_param_rpmMax</i>	determine le rpm maximum jusqu'à laquelle va le cadran

### 6.94.2 Member Function Documentation

### 6.94.2.1 paint()

```
void TachometerGauge_Lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

Fonction permettant l'affichage du compteur.

Cette fonction construit un compteur en plusieurs étapes : création du cadran, des graduations, ajout du texte sur les graduations, ajout de l'aiguille.

#### Parameters

<i>painter</i>	
----------------	--

Création du cadran.

Le cadran est basé sur un arc de cercle dessiné à l'aide de la fonction drawArc. le cercle est dessiné avec un QPen a qui a été donné une QBrush contenant un gradient radial de couleur bleu.

Création des graduations et du texte du cadran.

Les graduations sont créées en utilisant la fonction drawLine et une boucle for pour changer la couleur des graduations, comme le compteur n'est pas divisé régulièrement, le numéro des graduations sur lesquels les chiffres sont ajoutés sont rentrés à la main.

Création de l'aiguille.

Dans un premier temps, affichage de l'aiguille à l'aide de la fonction drawConvexPolygon. Puis affichage d'un petit cercle à la base de l'aiguille.

Création de l'affichage RPM.

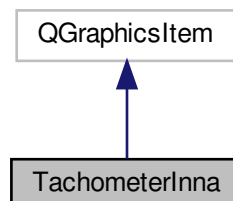
L'affichage RPM se fait avec la fonction drawText.

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

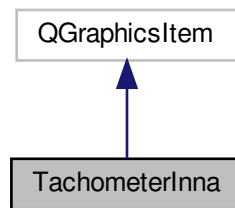
- serveur/Lea/tachometergauge\_lea.h
- serveur/Lea/[tachometergauge\\_lea.cpp](#)

## 6.95 TachometerInna Class Reference

Inheritance diagram for TachometerInna:



Collaboration diagram for TachometerInna:



### Public Member Functions

- **TachometerInna** (*QGraphicsItem* \*parent=nullptr)
- **TachometerInna** (int VMAX, int ALPHA0, int ALPHAMAX)
- *QRectF* **boundingRect** () const
- void **paint** (*QPainter* \*painter, const *QStyleOptionGraphicsItem* \*option, *QWidget* \*widget)

### Public Attributes

- int **r**
- int **A0**
- int **Amax**
- int **vmax**
- int **v**
- *QFont* **font** = *QFont*("MechEffects2 BB",22, -1, true )
- double **k**
- int **dx**
- int **dy**

## 6.95.1 Member Function Documentation

### 6.95.1.1 paint()

```

void TachometerInna::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
  
```

Dessin du circle encadrant.

On crée le gradient radial pour en suite instancier la brosse du \*painter et dessiner un circle encadrant avec un gradient de couleur.

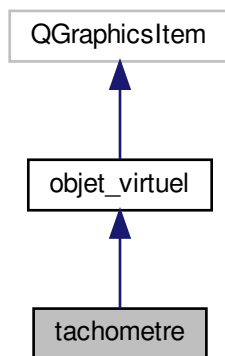
Dessin du texte et des traits.

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

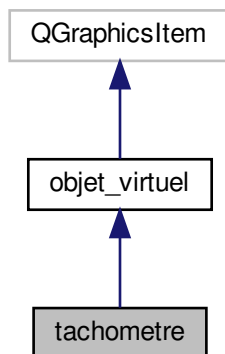
- serveur/Inna/tachometerinna.h
- serveur/Inna/[tachometerinna.cpp](#)

## 6.96 tachometre Class Reference

Inheritance diagram for tachometre:



Collaboration diagram for tachometre:



### Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- float **getRpmValue** (float rpmValue)

### Public Attributes

- float **rpmValue** =[getValue](#)()
- const double **pi** =3.142



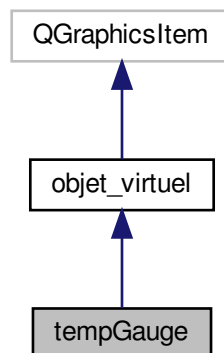
### Additional Inherited Members

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

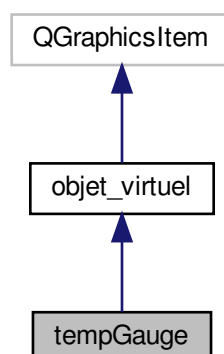
- serveur/Loto/tachometre.h
- serveur/Loto/tachometre.cpp

## 6.97 tempGauge Class Reference

Inheritance diagram for tempGauge:



Collaboration diagram for tempGauge:



## Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- float **getEngineTemp** (float engineTemp)

## Public Attributes

- float **engineTemp** =[getValue\(\)](#)
- const double **pi** =3.142

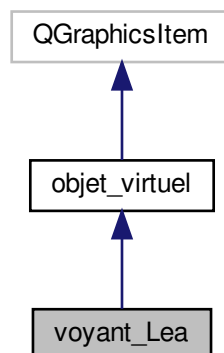
## Additional Inherited Members

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

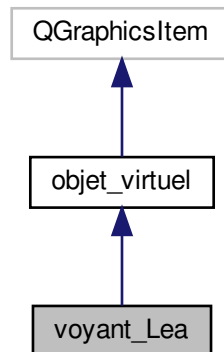
- serveur/Loto/tempgauge.h
- serveur/Loto/tempgauge.cpp

## 6.98 voyant\_Lea Class Reference

Inheritance diagram for voyant\_Lea:



Collaboration diagram for voyant\_Lea:



### Public Member Functions

- [voyant\\_Lea](#) (int, int, QString, int, int, QColor)  
*[voyant\\_Lea::voyant\\_Lea](#) Constructeur permettant de paramétrer la position et l'image des voyants*
- QRectF **boundingRect** () const
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

### Public Attributes

- QString **chemin**
- int **x**
- int **y**
- int **width** =30
- int **height** =30
- QColor **color**

### Additional Inherited Members

#### 6.98.1 Constructor & Destructor Documentation

##### 6.98.1.1 voyant\_Lea()

```

voyant_Lea::voyant_Lea (
    int param_x,
    int param_y,
    QString param_chemin,
    int param_width,
    int param_height,
    QColor param_color )

```

[voyant\\_Lea::voyant\\_Lea](#) Constructeur permettant de paramétrer la position et l'image des voyants

## Parameters

<i>param_x</i>	position horizontale
<i>param_y</i>	position verticale
<i>param_chemin</i>	chemin de la ressource
<i>param_width</i>	contrôle de l'épaisseur du pixmap
<i>param_height</i>	contrôle de la hauteur du pixmap

## 6.98.2 Member Function Documentation

## 6.98.2.1 paint()

```
void voyant_Lea::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget )
```

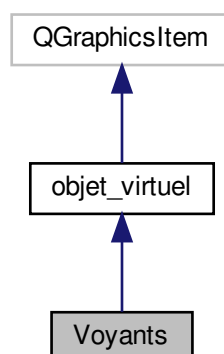
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché. Si ==0, le voyant doit être éteint

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

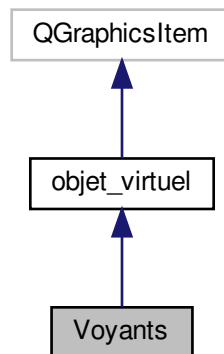
- serveur/Lea/voyant\_lea.h
- serveur/Lea/[voyant\\_lea.cpp](#)

## 6.99 Voyants Class Reference

Inheritance diagram for Voyants:



Collaboration diagram for Voyants:



### Public Member Functions

- **Voyants** (QString, int, int, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

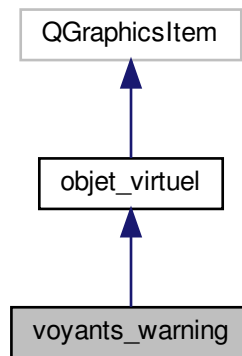
### Additional Inherited Members

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

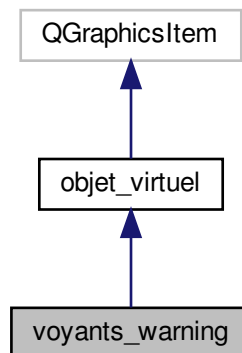
- serveur/Karim/voyants.h
- serveur/Karim/voyants.cpp

## 6.100 voyants\_warning Class Reference

Inheritance diagram for voyants\_warning:



Collaboration diagram for voyants\_warning:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)
- QRectF **boundingRect** () const

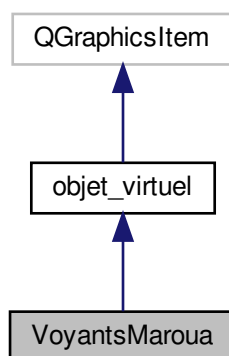
### Additional Inherited Members

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

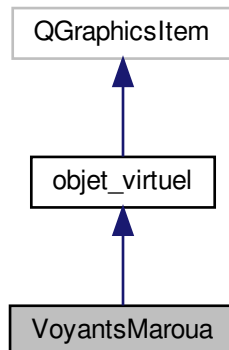
- serveur/Karim/voyants\_warning.h
- serveur/Karim/voyants\_warning.cpp

## 6.101 VoyantsMaroua Class Reference

Inheritance diagram for VoyantsMaroua:



Collaboration diagram for VoyantsMaroua:



### Public Member Functions

- **VoyantsMaroua** (QString, int, int, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

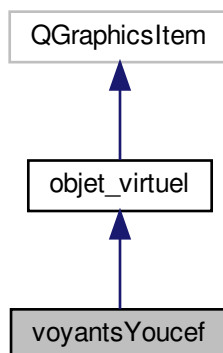
### Additional Inherited Members

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

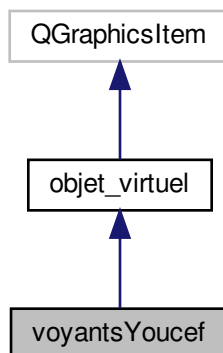
- serveur/Maroua/voyantsmaroua.h
- serveur/Maroua/voyantsmaroua.cpp

## 6.102 voyantsYoucef Class Reference

Inheritance diagram for voyantsYoucef:



Collaboration diagram for voyantsYoucef:



### Public Member Functions

- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
- `void flashing ()`

### Public Attributes

- `int clignotant`



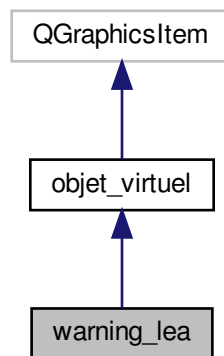
### Additional Inherited Members

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

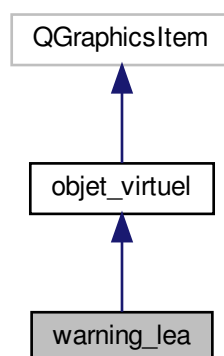
- serveur/Youcef/voyantsyoucef.h
- serveur/Youcef/voyantsyoucef.cpp

## 6.103 warning\_lea Class Reference

Inheritance diagram for warning\_lea:



Collaboration diagram for warning\_lea:



## Public Member Functions

- **warning\_lea** (int, int, QString, int, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

## Public Attributes

- QString **chemin**
- int **x**
- int **y**
- int **width**
- int **height**

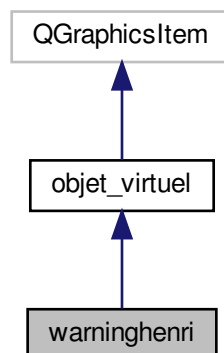
## Additional Inherited Members

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

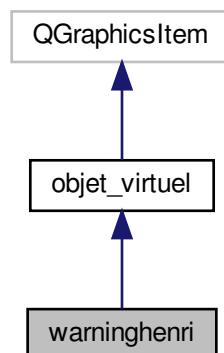
- serveur/Lea/warning\_lea.h
- serveur/Lea/[warning\\_lea.cpp](#)

## 6.104 warninghenri Class Reference

Inheritance diagram for warninghenri:



Collaboration diagram for warninghenri:



### Public Member Functions

- QRectF **boundingRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

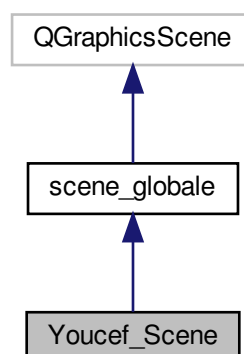
### Additional Inherited Members

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

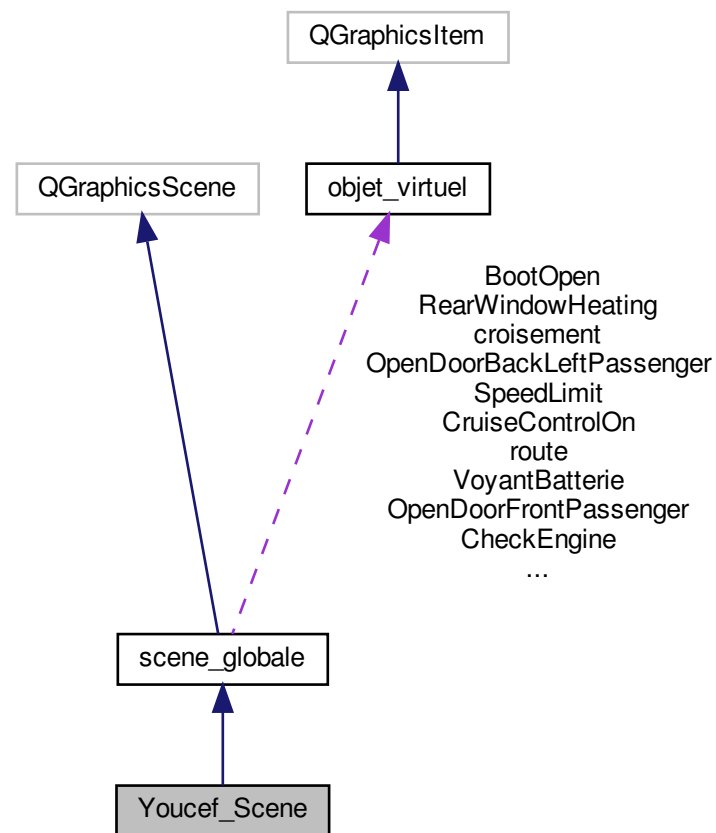
- serveur/Henri/warninghenri.h
- serveur/Henri/warninghenri.cpp

## 6.105 Youcef\_Scene Class Reference

Inheritance diagram for Youcef\_Scene:



Collaboration diagram for Youcef\_Scene:



## Public Member Functions

- **Youcef\_Scene** ([scene\\_globale](#) \*parent=nullptr)

## Additional Inherited Members

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

- serveur/Youcef/youcef\_scene.h
- serveur/Youcef/youcef\_scene.cpp

## Chapter 7

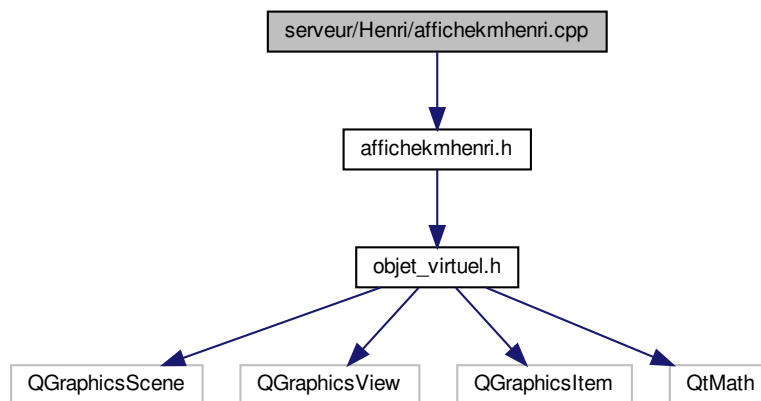
# File Documentation

### 7.1 serveur/Henri/affichekmhenri.cpp File Reference

Classe dérivant de [objet\\_virtuel](#) permettant l'affichage du nombre de km parcourut.

```
#include "affichekmhenri.h"
```

Include dependency graph for affichekmhenri.cpp:



#### 7.1.1 Detailed Description

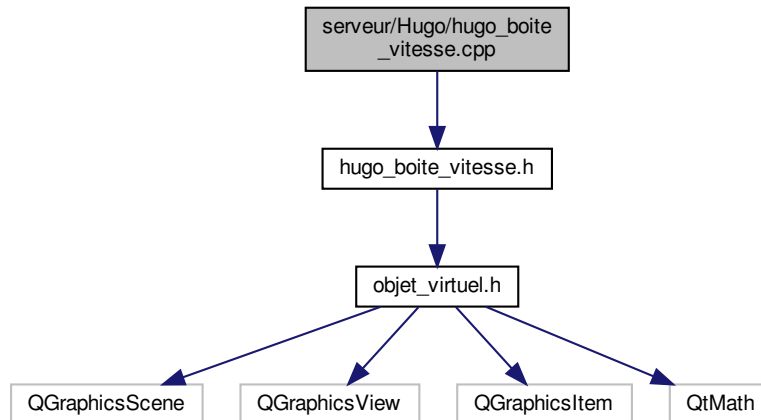
Classe dérivant de [objet\\_virtuel](#) permettant l'affichage du nombre de km parcourut.

## 7.2 serveur/Hugo/hugo\_boite\_vitesse.cpp File Reference

Classe dérivant de [objet\\_virtuel](#) permettant l'affichage du mode de transmission automatique du véhicule.

```
#include "hugo_boite_vitesse.h"
```

Include dependency graph for hugo\_boite\_vitesse.cpp:



### 7.2.1 Detailed Description

Classe dérivant de [objet\\_virtuel](#) permettant l'affichage du mode de transmission automatique du véhicule.

## 7.3 serveur/Hugo/hugo\_compteur.cpp File Reference

Classe compteur, permettant la création de compteurs paramétrables avec aiguille et effet de verre.

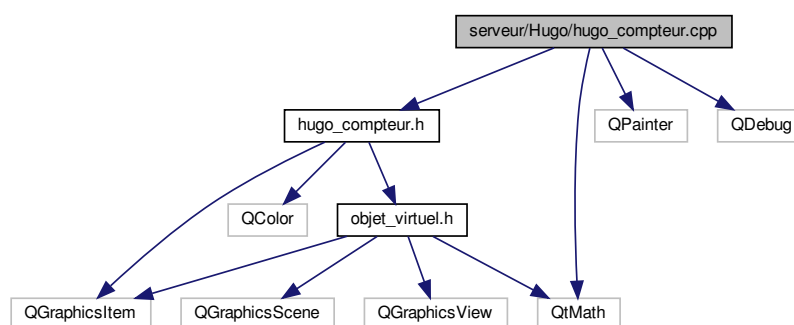
```
#include "hugo_compteur.h"
```

```
#include <QPainter>
```

```
#include <QtMath>
```

```
#include <QDebug>
```

Include dependency graph for hugo\_compteur.cpp:



## Macros

- `#define pi 3.14159265`

### 7.3.1 Detailed Description

Classe compteur, permettant la création de compteurs paramétrables avec aiguille et effet de verre.

Classe héritée de [objet\\_virtuel](#). Utilisée pour l'affichage du compteur de vitesse, du compteur rpm, de la jauge d'essence ainsi que de la jauge de température. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction paint permet l'affichage.

**Bug** L'utilisation du paramètre critique semble être soumis à certains bugs.

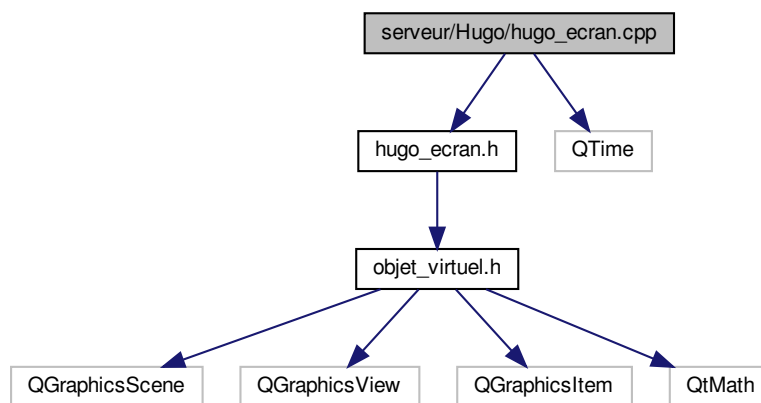
## 7.4 serveur/Hugo/hugo\_ecran.cpp File Reference

Classe dérivant de [objet\\_virtuel](#) permettant un affichage en temps réel du temps ainsi que de la distance parcourue depuis le redémarrage du programme.

```
#include "hugo_ecran.h"
```

```
#include <QTime>
```

Include dependency graph for hugo\_ecran.cpp:



### 7.4.1 Detailed Description

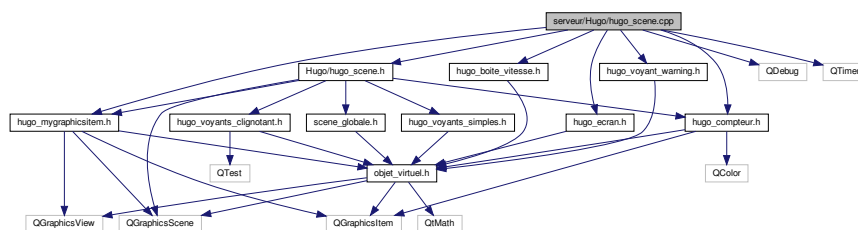
Classe dérivant de [objet\\_virtuel](#) permettant un affichage en temps réel du temps ainsi que de la distance parcourue depuis le redémarrage du programme.

## 7.5 serveur/Hugo/hugo\_scene.cpp File Reference

Classe dérivée de [scene\\_globale](#) où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans [scene\\_globale.h](#).

```
#include "Hugo/hugo_scene.h"
#include "Hugo/hugo_mygraphicsitem.h"
#include "Hugo/hugo_compteur.h"
#include "hugo_ecran.h"
#include "hugo_voyant_warning.h"
#include "hugo_boite_vitesse.h"
#include <QDebug>
#include <QTimer>
```

Include dependency graph for hugo\_scene.cpp:



### 7.5.1 Detailed Description

Classe dérivée de [scene\\_globale](#) où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans [scene\\_globale.h](#).

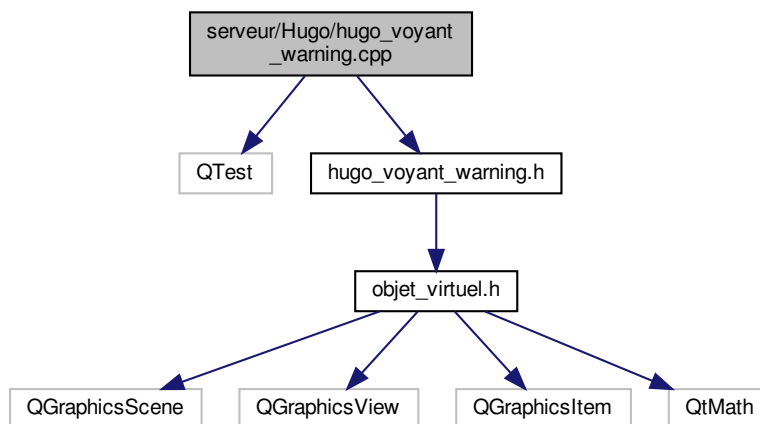
## 7.6 serveur/Hugo/hugo\_voyant\_warning.cpp File Reference

Classe dérivée de [objet\\_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

```
#include <QTest>
#include "hugo_voyant_warning.h"
```



Include dependency graph for hugo\_voyant\_warning.cpp:



### 7.6.1 Detailed Description

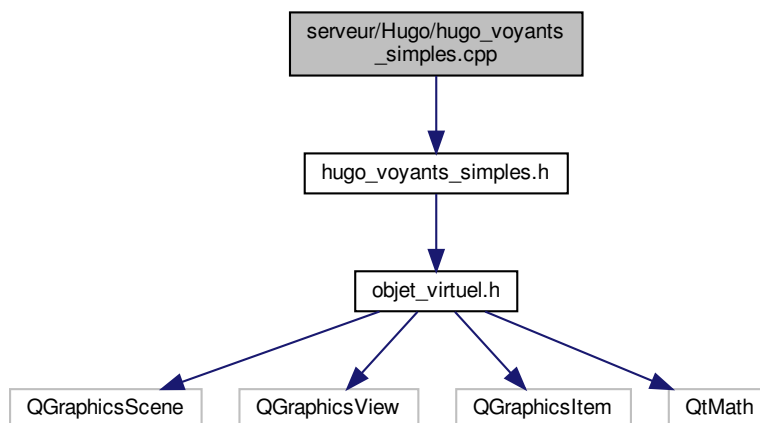
Classe dérivée de [objet\\_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

## 7.7 serveur/Hugo/hugo\_voyants\_simples.cpp File Reference

Classe permettant l'affichage de voyants simples i.e. dont l'état est allumé ou éteint.

```
#include "hugo_voyants_simples.h"
```

Include dependency graph for hugo\_voyants\_simples.cpp:



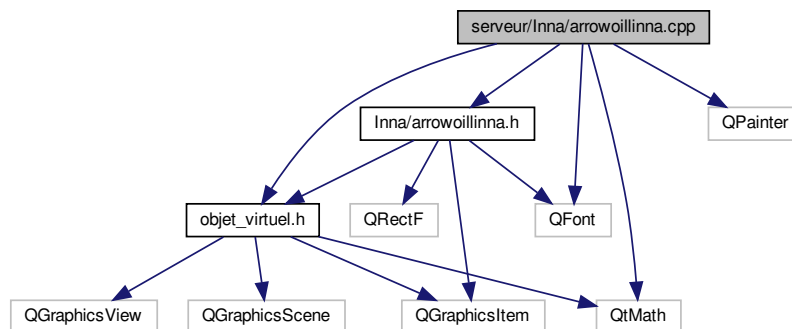
### 7.7.1 Detailed Description

Classe permettant l'affichage de voyants simples i.e. dont l'état est allumé ou éteint.

## 7.8 serveur/Inna/arrowoillinna.cpp File Reference

Classe [ArrowOilInna](#) dérivée de la classe [objet\\_virtuel](#).

```
#include "Inna/arrowoillinna.h"
#include "objet_virtuel.h"
#include <QFont>
#include <QtMath>
#include <QPainter>
Include dependency graph for arrowoillinna.cpp:
```



### 7.8.1 Detailed Description

Classe [ArrowOilInna](#) dérivée de la classe [objet\\_virtuel](#).

Permet la création de l' aiguille de la jauge du niveau d'huile qui tourne en fonction de la valeur **value**.

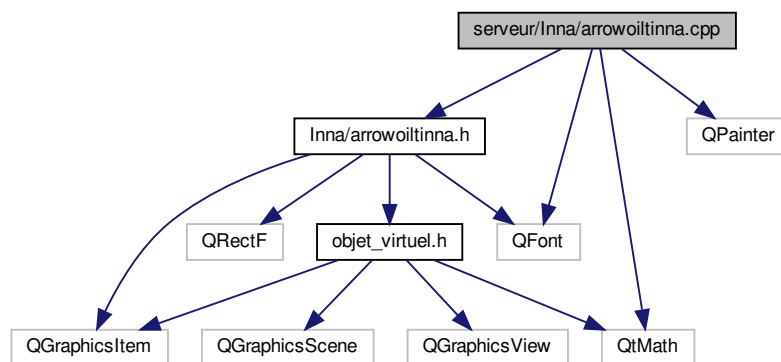
#### Parameters

<i>value</i>	- la valeur du niveau d'huile
<i>valueMax</i>	- le niveau d'huile maximum
<i>A0</i>	- l'aunge de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.9 serveur/Inna/arrowoiltinna.cpp File Reference

Classe [ArrowOilTinna](#) dérivée de la classe [objet\\_virtuel](#).

```
#include "Inna/arrowoiltinna.h"
#include <QFont>
#include <QtMath>
#include <QPainter>
Include dependency graph for arrowoiltinna.cpp:
```



### 7.9.1 Detailed Description

Classe [ArrowOilTinna](#) dérivée de la classe [objet\\_virtuel](#).

Permet la création de l'aiguille de la jauge de la température d'huile qui tourne en fonction de la valeur "value".

#### Parameters

<i>value</i>	- la valeur de la température d'huile
<i>valueMax</i>	- la température d'huile maximum
<i>A0</i>	- l'aunge de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

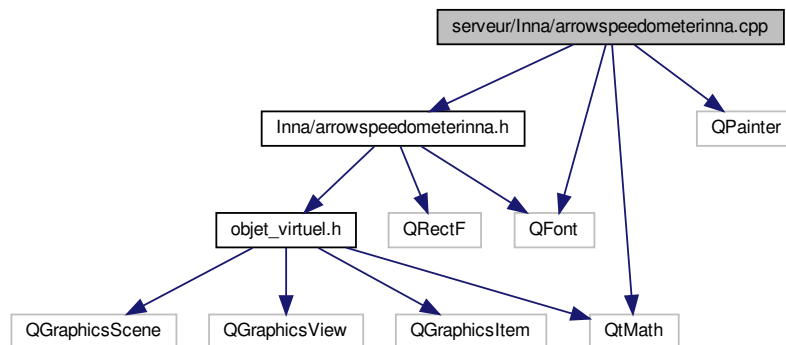
## 7.10 serveur/Inna/arrowspeedometerinna.cpp File Reference

Classe [ArrowSpeedometerInna](#) dérivée de la classe [objet\\_virtuel](#).

```
#include "Inna/arrowspeedometerinna.h"
#include <QFont>
#include <QtMath>
```

```
#include <QPainter>
```

Include dependency graph for `arrowspeedometerinna.cpp`:



### 7.10.1 Detailed Description

Classe [ArrowSpeedometerInna](#) dérivée de la classe [objet\\_virtuel](#).

Permettant la création de l'aiguille de la jauge de vitesse qui tourne en fonction de la valeur "value".

#### Parameters

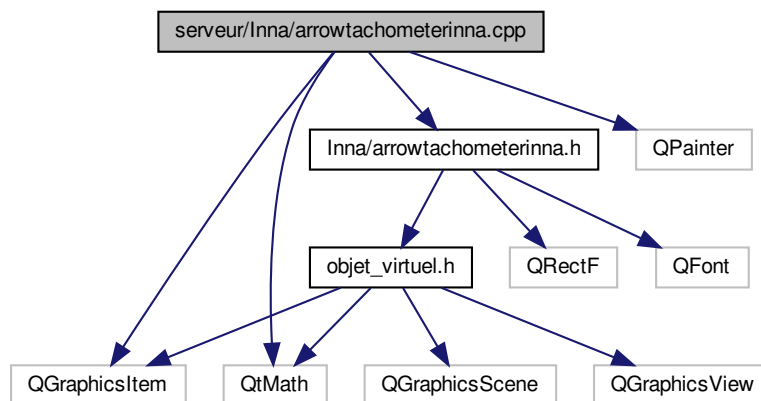
<i>value</i>	- la valeur de vitesse
<i>valueMax</i>	- la vitesse maximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.11 serveur/Inna/arrowtachometerinna.cpp File Reference

Classe [ArrowTachometerInna](#) dérivée de la classe [objet\\_virtuel](#).

```
#include "Inna/arrowtachometerinna.h"
#include <QGraphicsItem>
#include <QtMath>
#include <QPainter>
```

Include dependency graph for arrowtachometerinna.cpp:



### 7.11.1 Detailed Description

Classe [ArrowTachometerInna](#) dérivée de la classe [objet\\_virtuel](#).

Permettant la création de l'aiguille de la jauge compte-tours qui tourne en fonction de la valeur "value".

#### Parameters

<i>value</i>	- la valeur des tours par minute
<i>valueMax</i>	- la valeur des tours par minute maximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.12 serveur/Inna/clignotantinna.cpp File Reference

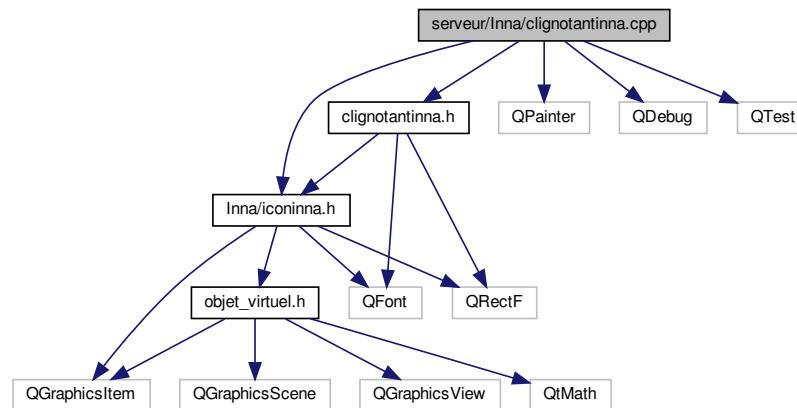
Classe permettant d'afficher les voyants clignotants.

```

#include "clignotantinna.h"
#include "Inna/iconinna.h"
#include <QPainter>
#include <QDebug>
#include <QTest>

```

Include dependency graph for `clignotantinna.cpp`:



### 7.12.1 Detailed Description

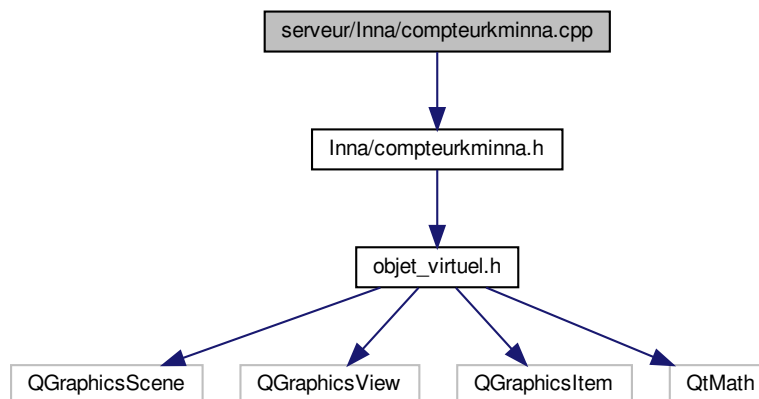
Classe permettant d'afficher les voyants clignotants.

## 7.13 serveur/Inna/compteurkminna.cpp File Reference

Classe dérivant de `objet_virtuel` permettant un affichage en temps réel de la distance parcourue depuis le redémarrage du programme.

```
#include "Inna/compteurkminna.h"
```

Include dependency graph for `compteurkminna.cpp`:



### 7.13.1 Detailed Description

Classe dérivant de [objet\\_virtuel](#) permettant un affichage en temps réel de la distance parcourue depuis le redémarrage du programme.

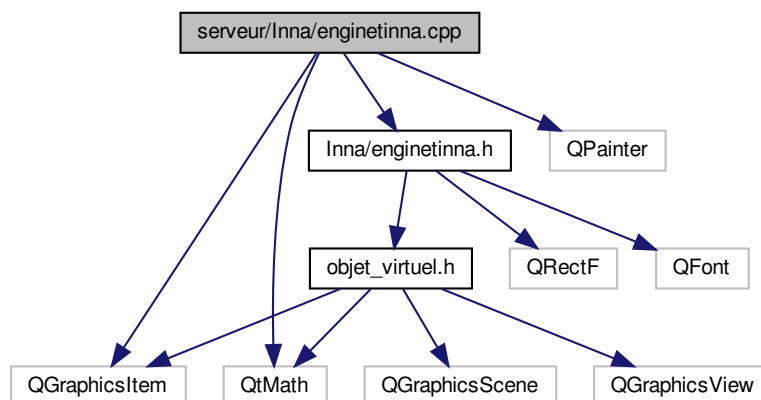
compteurKmlInna::compteurKmlInna

Constructeur sans paramètres spécifiques. Initialise la valeur **value** de la classe mère à 0.

## 7.14 serveur/Inna/enginetinna.cpp File Reference

Classe [EngineTInna](#) dérivée de la classe [objet\\_virtuel](#).

```
#include "Inna/enginetinna.h"
#include <QGraphicsItem>
#include <QtMath>
#include <QPainter>
Include dependency graph for enginetinna.cpp:
```



### 7.14.1 Detailed Description

Classe [EngineTInna](#) dérivée de la classe [objet\\_virtuel](#).

Permettant la création de l' aiguille de la jauge de la température du moteur qui tourne en fonction de la valeur "value".

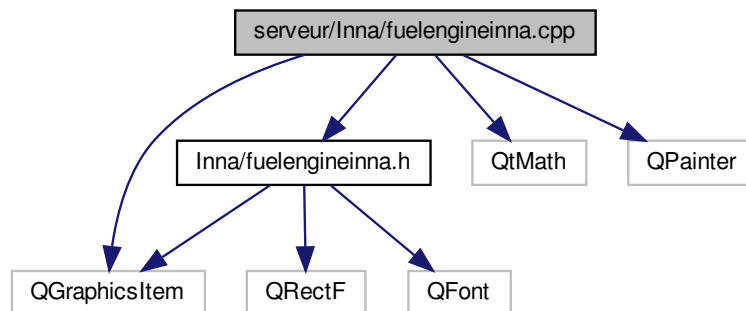
#### Parameters

<i>value</i>	- la valeur de la température du moteur
<i>valueMax</i>	- la température maximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.15 serveur/Inna/fuelengineinna.cpp File Reference

Classe dérivée de QGraphicsItem permettant l'affichage des jauges de l'essence et de la température du moteur.

```
#include "Inna/fuelengineinna.h"
#include <QGraphicsItem>
#include <QtMath>
#include <QPainter>
Include dependency graph for fuelengineinna.cpp:
```



### 7.15.1 Detailed Description

Classe dérivée de QGraphicsItem permettant l'affichage des jauges de l'essence et de la température du moteur.

#### Parameters

<i>tmax</i>	- température du moteur;
<i>r</i>	- rayon de l'arche de la jauge;
<i>A0</i>	- l'angle du début de la jauge;
<i>Amax</i>	- l'angle total
<i>k</i>	- coefficient pour convertir la valeur en degrés

## 7.16 serveur/Inna/fuellevelinna.cpp File Reference

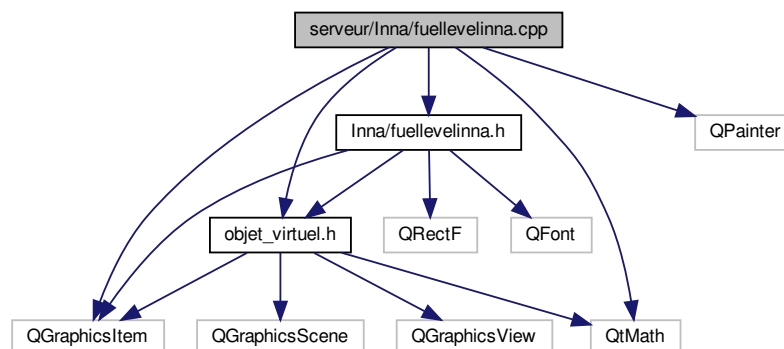
Classe `FuelLevelInna` dérivée de la classe `objet_virtuel`.

```
#include "Inna/fuellevelinna.h"
#include "objet_virtuel.h"
#include <QGraphicsItem>
#include <QtMath>
```



```
#include <QPainter>
```

Include dependency graph for fuellevelinna.cpp:



### 7.16.1 Detailed Description

Classe `FuelLevelInna` dérivée de la classe `objet_virtuel`.

Permettant la création de l'aiguille de la jauge du niveau d'essence qui tourne en fonction de la valeur "value".

#### Parameters

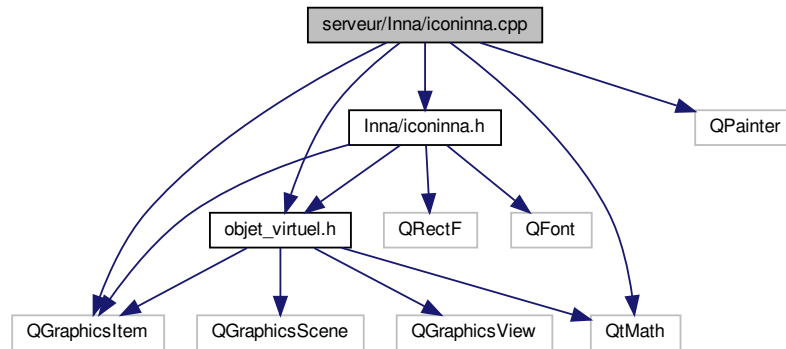
<i>value</i>	- la valeur du niveau d'essence
<i>valueMax</i>	- le niveau d'essence maximum
<i>A0</i>	- l'aunge de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.17 serveur/Inna/iconinna.cpp File Reference

Classe `IconInna` dérivée de la classe `objet_virtuel`.

```
#include "Inna/iconinna.h"
#include <QGraphicsItem>
#include <QtMath>
#include <QPainter>
#include "objet_virtuel.h"
```

Include dependency graph for `iconinna.cpp`:



### 7.17.1 Detailed Description

Classe `IconInna` dérivée de la classe `objet_virtuel`.

La classe est utilisée pour ajouter des attributs communs pour les voyants.

#### Parameters

<i>position</i>	- parameter de type <code>QPoint(int,int)</code> pour initialiser la position d'un voyant.
<i>size</i>	- parameter de type <code>QSize(int,int)</code> pour initialiser la taille d'un voyant.
<i>imagePath</i>	- parameter de type <code>QString</code> qui correspond au chemin de l'image de voyant.

## 7.18 serveur/Inna/icononoffinna.cpp File Reference

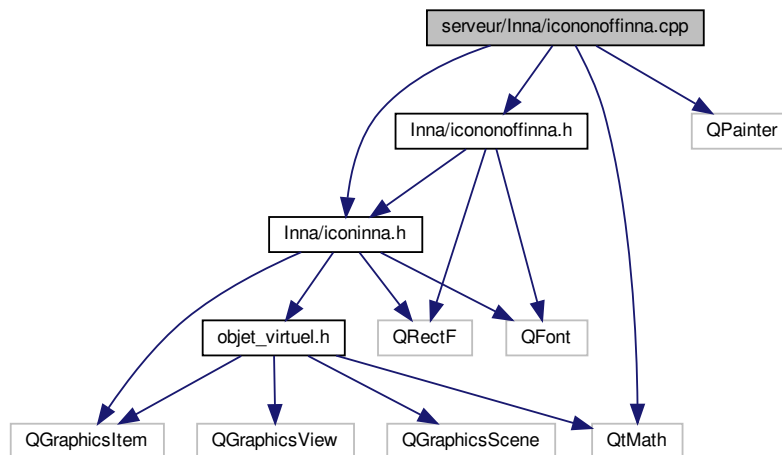
Classe `iconOnOffInna` dérivée de la classe `IconInna`.

```

#include "Inna/icononoffinna.h"
#include "Inna/iconinna.h"
#include <QPainter>
#include <QtMath>

```

Include dependency graph for icononoffinna.cpp:



### 7.18.1 Detailed Description

Classe `iconOnOffInna` dérivée de la classe `IconInna`.

La classe est utilisée pour instancier les voyants qui n'ont que deux états: allumé ou éteint.

## 7.19 serveur/Inna/infowindowinna.cpp File Reference

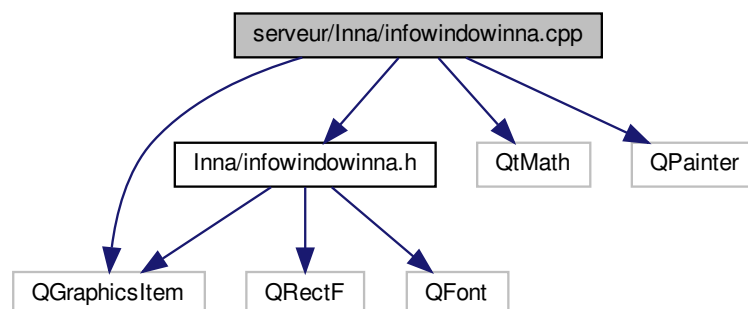
Classe `InfoWindowInna` dérivée de la classe `QGraphicsItem`.

```

#include "Inna/infowindowinna.h"
#include <QGraphicsItem>
#include <QtMath>
#include <QPainter>

```

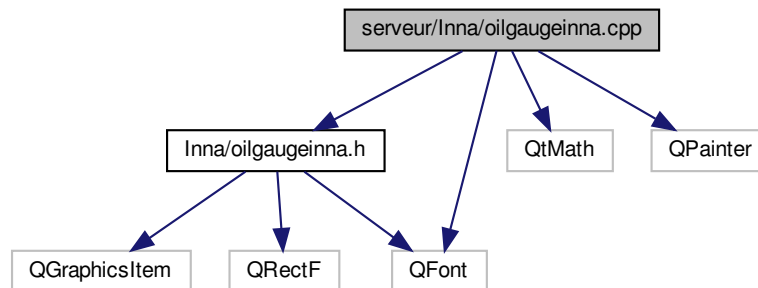
Include dependency graph for infowindowinna.cpp:





```
#include <QPainter>
```

Include dependency graph for oilgaugeinna.cpp:



### 7.21.1 Detailed Description

Classe dérivée de `QGraphicsItem` permettant l'affichage des jauges de niveau et de température d'huile.

#### Parameters

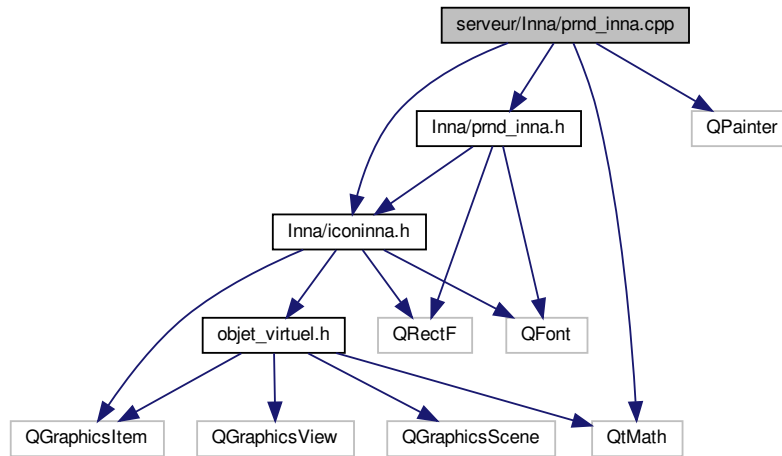
<i>tmax</i>	- la température maximum;
<i>lmax</i>	- le niveau maximum
<i>r</i>	- rayon de l'arche de la jauge;
<i>A0</i>	- l'angle du début de la jauge;
<i>Amax</i>	- l'angle total
<i>k</i>	- coefficient pour convertir la valeur en degrés

## 7.22 serveur/Inna/prnd\_inna.cpp File Reference

Classe `PRND_Inna` dérivée de la classe `IconInna` pour afficher les modes de la transmission.

```
#include "Inna/prnd_inna.h"
#include "Inna/iconinna.h"
#include <QPainter>
#include <QtMath>
```

Include dependency graph for prnd\_inna.cpp:



### 7.22.1 Detailed Description

Classe `PRND_Inna` dérivée de la classe `IconInna` pour afficher les modes de la transmission.

Affiche la mode de la transmission de la boîte de vitesse automatique.

## 7.23 serveur/Inna/speedometerinna.cpp File Reference

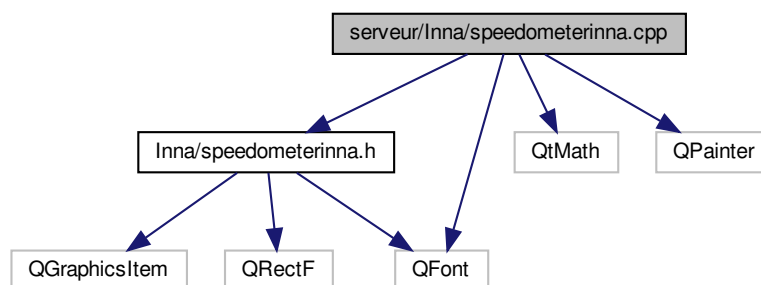
Classe `SpeedometerInna` dérivée de la classe `QGraphicsItem`.

```

#include "Inna/speedometerinna.h"
#include <QFont>
#include <QtMath>
#include <QPainter>

```

Include dependency graph for `speedometerinna.cpp`:



### 7.23.1 Detailed Description

Classe [SpeedometerInna](#) dérivée de la classe `QGraphicsItem`.

Permettant la création d'un objet static de la jauge vitesse

#### Parameters

<i>value</i>	- la valeur de vitesse
<i>valueMax</i>	- la vitesse maximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.24 serveur/Inna/tachometerinna.cpp File Reference

Classe [TachometerInna](#) dérivée de la classe `QGraphicsItem`.

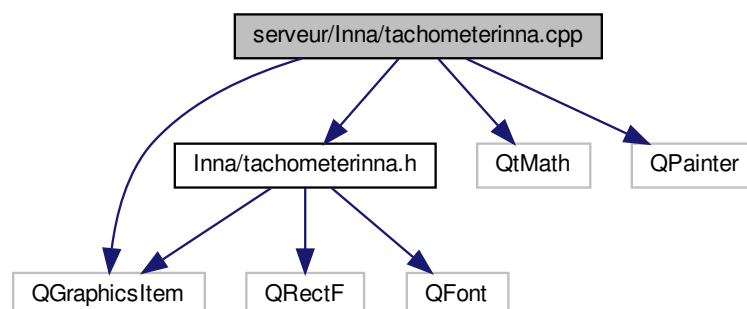
```
#include "Inna/tachometerinna.h"
```

```
#include <QGraphicsItem>
```

```
#include <QtMath>
```

```
#include <QPainter>
```

Include dependency graph for tachometerinna.cpp:



### 7.24.1 Detailed Description

Classe [TachometerInna](#) dérivée de la classe `QGraphicsItem`.

Permettant la création d'un objet static de la jauge compte-tours

#### Parameters

<i>value</i>	- la valeur de tours par minute
--------------	---------------------------------

## Parameters

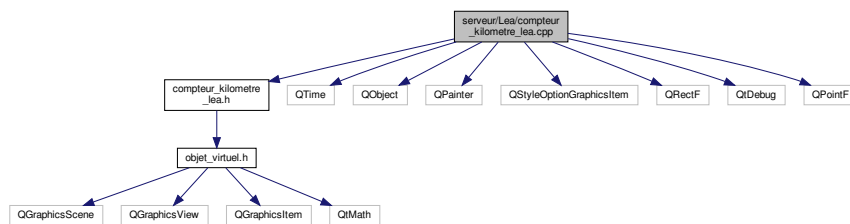
<i>valueMax</i>	- la valeur de tours par minutemaximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

## 7.25 serveur/Lea/compteur\_kilometre\_lea.cpp File Reference

Classe dérivant de [objet\\_virtuel](#) permettant un affichage en temps réel de la distance parcourue depuis le démarrage du programme.

```
#include "compteur_kilometre_lea.h"
#include <QTime>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QRectF>
#include <QtDebug>
#include <QPointF>
```

Include dependency graph for compteur\_kilometre\_lea.cpp:



### 7.25.1 Detailed Description

Classe dérivant de [objet\\_virtuel](#) permettant un affichage en temps réel de la distance parcourue depuis le démarrage du programme.

## 7.26 serveur/Lea/fuelgauge\_lea.cpp File Reference

Classe jauge d'essence, permettant la création d'une jauge paramétrables .

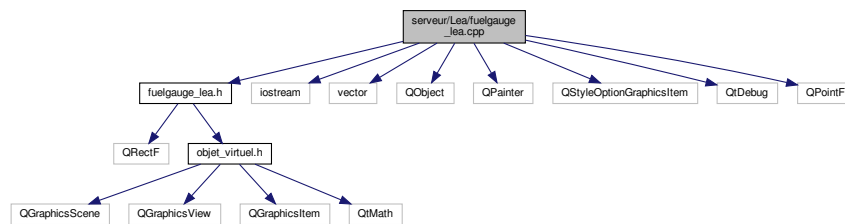
```
#include "fuelgauge_lea.h"
#include <iostream>
#include <vector>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
```



```
#include <QtDebug>
```

```
#include <QPointF>
```

Include dependency graph for fuelgauge\_lea.cpp:



### 7.26.1 Detailed Description

Classe jauge d'essence, permettant la création d'une jauge paramétrables .

Classe héritée de [objet\\_virtuel](#). Utilisée pour l'affichage de la jauge d'essence ou de toute autre jauge du même type. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction paint permet l'affichage.

## 7.27 serveur/Lea/jauge\_temperature\_lea.cpp File Reference

Classe jauge de température, permettant la création d'une jauge paramétrable.

```
#include "jauge_temperature_lea.h"
```

```
#include <QObject>
```

```
#include <QPainter>
```

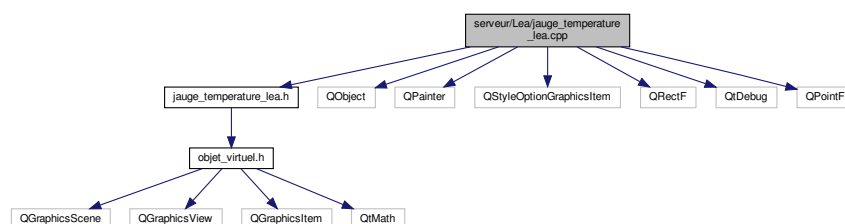
```
#include <QStyleOptionGraphicsItem>
```

```
#include <QRectF>
```

```
#include <QtDebug>
```

```
#include <QPointF>
```

Include dependency graph for jauge\_temperature\_lea.cpp:



### 7.27.1 Detailed Description

Classe jauge de température, permettant la création d'une jauge paramétrable.

Classe héritée de [objet\\_virtuel](#). Utilisée pour l'affichage de la jauge d'essence ou de toute autre jauge du même type. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction paint permet l'affichage.



### 7.29.1 Detailed Description

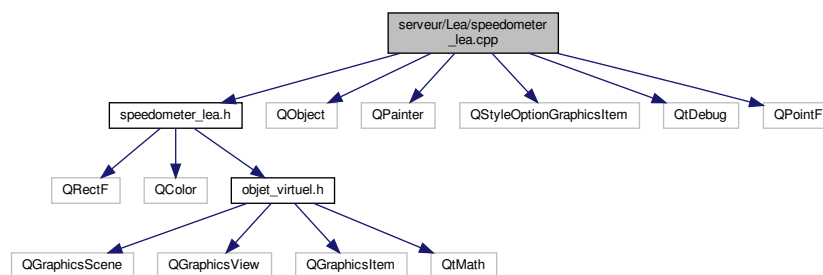
Classe dérivée de [scene\\_globale](#) où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans [scene\\_globale.h](#).

## 7.30 serveur/Lea/speedometer\_lea.cpp File Reference

Classe speedometer, permettant la création du compteur vitesse paramétrable avec aiguille.

```
#include "speedometer_lea.h"
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QtDebug>
#include <QPointF>
```

Include dependency graph for `speedometer_lea.cpp`:



### 7.30.1 Detailed Description

Classe speedometer, permettant la création du compteur vitesse paramétrable avec aiguille.

Classe héritée de [objet\\_virtuel](#). L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction `paint` permet l'affichage.

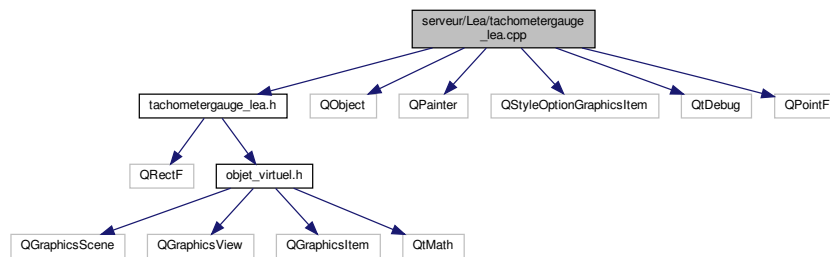
## 7.31 serveur/Lea/tachometergauge\_lea.cpp File Reference

Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille.

```
#include "tachometergauge_lea.h"
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QtDebug>
```

```
#include <QPointF>
```

Include dependency graph for tachometergauge\_lea.cpp:



### 7.31.1 Detailed Description

Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille.

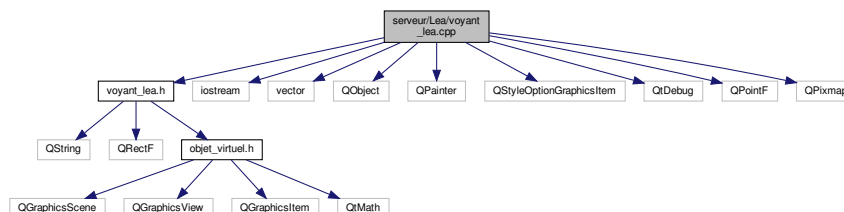
Classe héritée de [objet\\_virtuel](#). Utilisée pour l'affichage du compteur rpm. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction paint permet l'affichage.

## 7.32 serveur/Lea/voyant\_lea.cpp File Reference

Classe permettant l'affichage de voyants dont l'état est allumé ou éteint.

```
#include "voyant_lea.h"
#include <iostream>
#include <vector>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QtDebug>
#include <QPointF>
#include <QPixmap>
```

Include dependency graph for voyant\_lea.cpp:



### 7.32.1 Detailed Description

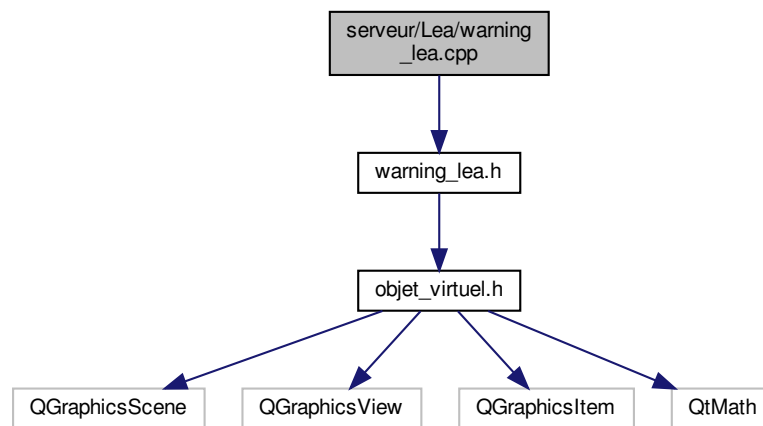
Classe permettant l'affichage de voyants dont l'état est allumé ou éteint.

## 7.33 serveur/Lea/warning\_lea.cpp File Reference

Classe dérivée de [objet\\_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

```
#include "warning_lea.h"
```

Include dependency graph for warning\_lea.cpp:



### 7.33.1 Detailed Description

Classe dérivée de [objet\\_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

