

Doc car_dashboard

Generated by Doxygen 1.8.13

Contents

1	Projet dashboard	1
2	Bug List	5
3	Hierarchical Index	7
3.1	Class Hierarchy	7
4	Class Index	9
4.1	Class List	9
5	File Index	11
5.1	File List	11
6	Class Documentation	15
6.1	afficheKmHenri Class Reference	15
6.2	aiguille_Lea Class Reference	16
6.3	ArrowOilInna Class Reference	18
6.4	ArrowOilTInna Class Reference	19
6.5	ArrowSpeedometerInna Class Reference	21
6.6	ArrowTachometerInna Class Reference	22
6.7	CadranEss Class Reference	24
6.8	CadranTFlorian Class Reference	25
6.9	cadranTHenri Class Reference	26
6.10	CadranTourParMin Class Reference	27
6.11	cadranTVirtuel Class Reference	28
6.12	CadranVitesse Class Reference	29

6.13	clignot Class Reference	30
6.14	clignotantInna Class Reference	32
6.15	deux_voyants_lea Class Reference	33
6.16	EngineTInna Class Reference	35
6.17	feuxHenri Class Reference	36
6.18	FlorianClignotant Class Reference	38
6.19	FlorianCompteurKm Class Reference	39
6.20	FlorianVoyant Class Reference	40
6.21	fuel_guage Class Reference	41
6.22	FuelEngineInna Class Reference	42
6.23	FuelGauge_lea Class Reference	43
6.24	FuelLevelInna Class Reference	45
6.25	henri_scene Class Reference	46
6.26	hugo_boite_vitesse Class Reference	48
6.26.1	Member Function Documentation	48
6.26.1.1	paint()	49
6.27	hugo_Compteur Class Reference	49
6.27.1	Constructor & Destructor Documentation	50
6.27.1.1	hugo_Compteur()	51
6.27.2	Member Function Documentation	51
6.27.2.1	paint()	51
6.28	hugo_ecran Class Reference	53
6.28.1	Constructor & Destructor Documentation	54
6.28.1.1	hugo_ecran()	54
6.28.2	Member Function Documentation	54
6.28.2.1	paint()	54
6.29	hugo_MyGraphicsitem Class Reference	55
6.29.1	Constructor & Destructor Documentation	56
6.29.1.1	hugo_MyGraphicsitem()	56
6.29.2	Member Function Documentation	56

6.29.2.1	paint()	56
6.30	hugo_scene Class Reference	57
6.31	hugo_voyant_warning Class Reference	59
6.31.1	Constructor & Destructor Documentation	60
6.31.1.1	hugo_voyant_warning()	60
6.31.2	Member Function Documentation	60
6.31.2.1	paint()	60
6.32	hugo_voyants_clignotant Class Reference	61
6.32.1	Constructor & Destructor Documentation	62
6.32.1.1	hugo_voyants_clignotant()	62
6.32.2	Member Function Documentation	62
6.32.2.1	paint()	62
6.33	hugo_voyants_simples Class Reference	63
6.33.1	Constructor & Destructor Documentation	64
6.33.1.1	hugo_voyants_simples()	64
6.33.2	Member Function Documentation	64
6.33.2.1	paint()	64
6.34	IconInna Class Reference	65
6.35	iconOnOff Class Reference	66
6.36	iconOnOffInna Class Reference	68
6.37	InfoWindowInna Class Reference	69
6.38	inna_scene Class Reference	70
6.39	jaugeClignotantHenri Class Reference	72
6.40	jaugeEssenceHenri Class Reference	73
6.41	jaugeTemperatureHenri Class Reference	74
6.42	jaugeToursMinuteHenri Class Reference	75
6.43	jaugeVirtual Class Reference	76
6.44	Jonas_compteur Class Reference	77
6.45	jonas_compteurKm Class Reference	79
6.46	Jonas_scene Class Reference	80

6.47	jonas_voyant_clignotant Class Reference	82
6.48	jonas_voyant_simple Class Reference	83
6.49	jonas_voyant_warning Class Reference	84
6.50	karim_scene Class Reference	85
6.51	lea_clignottants Class Reference	87
6.52	Lea_scene Class Reference	88
6.53	lea_SceneDeFond Class Reference	90
6.54	LEDS Class Reference	91
6.55	Leo_display Class Reference	92
6.56	Leo_gauge Class Reference	93
6.57	Leo_indicator Class Reference	94
6.58	Leo_label Class Reference	95
6.59	Leo_scene Class Reference	96
6.60	loto_dashboard Class Reference	98
6.61	loto_scene Class Reference	99
6.62	MainWindow Class Reference	101
6.62.1	Constructor & Destructor Documentation	102
6.62.1.1	MainWindow()	102
6.62.2	Member Function Documentation	102
6.62.2.1	acceleration()	102
6.63	mygraphicItem Class Reference	103
6.64	objet_virtuel Class Reference	104
6.64.1	Constructor & Destructor Documentation	105
6.64.1.1	objet_virtuel()	105
6.64.2	Member Function Documentation	105
6.64.2.1	getValue()	105
6.64.2.2	getValueMax()	105
6.64.2.3	setValue()	105
6.65	OilGaugeInna Class Reference	106
6.66	onoff Class Reference	107

6.67	onoffpaintHenri Class Reference	109
6.68	phares_Lea Class Reference	110
6.69	portesHenri Class Reference	111
6.70	PRND_Inna Class Reference	113
6.71	Quatre_Voyants_Lea Class Reference	114
6.72	scene_globale Class Reference	116
6.72.1	Constructor & Destructor Documentation	118
6.72.1.1	scene_globale()	118
6.73	sceneDeFond Class Reference	118
6.74	sceneDeFondHenri Class Reference	119
6.75	SceneFlorian Class Reference	120
6.76	sceneGlobale Class Reference	122
6.77	speedometer_Lea Class Reference	123
6.78	SpeedometerInna Class Reference	125
6.79	TachometerGauge_Lea Class Reference	126
6.80	TachometerInna Class Reference	127
6.81	tachometre Class Reference	129
6.82	tempGauge Class Reference	130
6.83	voyant_Lea Class Reference	131
6.84	Voyants Class Reference	133
6.85	warninghenri Class Reference	134
7	File Documentation	135
7.1	serveur/Hugo/hugo_compteur.cpp File Reference	135
7.1.1	Detailed Description	135
7.2	serveur/Hugo/hugo_ecran.cpp File Reference	136
7.2.1	Detailed Description	136
7.3	serveur/Hugo/hugo_scene.cpp File Reference	136
7.3.1	Detailed Description	137
7.4	serveur/Hugo/hugo_voyant_warning.cpp File Reference	137
7.4.1	Detailed Description	137
7.5	serveur/Hugo/hugo_voyants_simples.cpp File Reference	138
7.5.1	Detailed Description	138

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	Temperature 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
jaugeTemperature	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

|

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	16
cadrantVirtuel	28
CarDialerPrototype	??
EngineHeatGauge	??
FuelEngineInna	42
FuelGauge	??
InfoWindowInna	69
jaugeVirtuel	76
lea_SceneDeFond	90
mygraphicItem	103
objet_virtuel	104
afficheKmHenri	15
ArrowOilInna	18
ArrowOilTInna	19
ArrowSpeedometerInna	21
ArrowTachometerInna	22
cadrant	??
CadrantEss	24
CadrantFlorian	25
cadrantHenri	26
CadrantTourParMin	27
CadrantVitesse	29
CarSpeedDial	??
clignot	30
Compteur_kilometre_lea	??
compteurKmInna	??
EngineTInna	35
FlorianClignotant	38
FlorianCompteurKm	39
FlorianVoyant	40
fuel_guage	41
FuelGauge_lea	43
FuelLevelInna	45
Harc	??

hugo_boite_vitesse	48
hugo_Compteur	49
hugo_ecran	53
hugo_MyGraphicsitem	55
hugo_voyant_warning	59
hugo_voyants_clignotant	61
hugo_voyants_simples	63
IconInna	65
clignotantInna	32
iconOnOff	66
iconOnOffInna	68
PRND_Inna	113
jauge_temperature_lea	??
jaugeClignotantHenri	72
jaugeEssenceHenri	73
jaugeTemperatureHenri	74
jaugeToursMinuteHenri	75
Jonas_compteur	77
jonas_compteurKm	79
jonas_voyant_clignotant	82
jonas_voyant_simple	83
jonas_voyant_warning	84
lea_clignottants	87
LEDS	91
Leo_display	92
Leo_gauge	93
Leo_indicator	94
loto_dashboard	98
onoff	107
onoffpaintHenri	109
portesHenri	111
speedometer_Lea	123
TachometerGauge_Lea	126
tachometre	129
tempGauge	130
voyant_Lea	131
Voyants	133
voyantsYucef	??
warning_lea	??
warninghenri	134
OilGaugeInna	106
sceneDeFond	118
sceneDeFondHenri	119
SpeedometerInna	125
Tachometer	??
TachometerInna	127
QGraphicsScene	
scene_globale	116
harout_scene	??
henri_scene	46
hugo_scene	57
inna_scene	70
Jonas_scene	80
karim_scene	85
Lea_scene	88
Leo_scene	96
loto_scene	99
SceneFlorian	120

Youcef_Scene	??
sceneGlobale	122
QGraphicsSimpleTextItem	
Leo_label	95
QMainWindow	
MainWindow	101
MainWindow	101
MainWindow	101
MainWindow	101

Chapter 4

Class Index

4.1 Class List

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

afficheKmHenri	15
aiguille_Lea	16
ArrowOilInna	18
ArrowOilTInna	19
ArrowSpeedometerInna	21
ArrowTachometerInna	22
cadran	??
CadranEss	24
CadrantFlorian	25
cadrantHenri	26
CadranTourParMin	27
cadrantVirtuel	28
CadranVitesse	29
CarDialerPrototype	??
CarSpeedDial	??
clignot	30
clignotantInna	32
Compteur_kilometre_lea	??
compteurKmlInna	??
EngineHeatGauge	??
EngineTInna	35
FlorianClignotant	38
FlorianCompteurKm	39
FlorianVoyant	40
fuel_guage	41
FuelEngineInna	42
FuelGauge	??
FuelGauge_lea	43
FuelLevellInna	45
Harc	??
harout_scene	??
henri_scene	46
hugo_boite_vitesse	48
hugo_Compteur	49
hugo_ecran	53

hugo_MyGraphicsitem	55
hugo_scene	57
hugo_voyant_warning	59
hugo_voyants_clignotant	61
hugo_voyants_simples	63
IconInna	65
iconOnOff	66
iconOnOffInna	68
InfoWindowInna	69
inna_scene	70
jauge_temperature_lea	??
jaugeClignotantHenri	72
jaugeEssenceHenri	73
jaugeTemperatureHenri	74
jaugeToursMinuteHenri	75
jaugeVirtuel	76
Jonas_compteur	77
jonas_compteurKm	79
Jonas_scene	80
jonas_voyant_clignotant	82
jonas_voyant_simple	83
jonas_voyant_warning	84
karim_scene	85
lea_clignottants	87
Lea_scene	88
lea_SceneDeFond	90
LEDS	91
Leo_display	92
Leo_gauge	93
Leo_indicator	94
Leo_label	95
Leo_scene	96
loto_dashboard	98
loto_scene	99
MainWindow	101
mygraphicItem	103
objet_virtuel	104
OilGaugeInna	106
onoff	107
onoffpaintHenri	109
portesHenri	111
PRND_Inna	113
scene_globale	116
sceneDeFond	118
sceneDeFondHenri	119
SceneFlorian	120
sceneGlobale	122
speedometer_Lea	123
SpeedometerInna	125
Tachometer	??
TachometerGauge_Lea	126
TachometerInna	127
tachometre	129
tempGauge	130
voyant_Lea	131
Voyants	133
voyantsYoucef	??
warning_lea	??

warninghenri	134
Youcef_Scene	??

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.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 objet_virtuel permettant l'affichage du mode de transmission automatique du véhicule	??
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	135

serveur/Hugo/ hugo_compteur.h	??
serveur/Hugo/ hugo_ecran.cpp	
Classe dérivant de objet_virtuel permettant un affichage en temps réel du temps ainsi que de la distance parcourue depuis le redémarrage du programme	136
serveur/Hugo/ hugo_ecran.h	??
serveur/Hugo/ hugo_mygraphicsitem.h	??
serveur/Hugo/ hugo_scene.cpp	
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	136
serveur/Hugo/ hugo_scene.h	??
serveur/Hugo/ hugo_voyant_warning.cpp	
Classe dérivée de objet_virtuel permettant l'affichage d'un voyant warning clignotant	137
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	138
serveur/Hugo/ hugo_voyants_simples.h	??
serveur/Inna/ arrowoillinna.cpp	
Classe ArrowOilInna dérivée de la classe objet_virtuel	??
serveur/Inna/ arrowoillinna.h	??
serveur/Inna/ arrowoiltinna.cpp	
Classe ArrowOilTInna dérivée de la classe objet_virtuel	??
serveur/Inna/ arrowoiltinna.h	??
serveur/Inna/ arrowspeedometerinna.cpp	
Classe ArrowSpeedometerInna dérivée de la classe objet_virtuel	??
serveur/Inna/ arrowspeedometerinna.h	??
serveur/Inna/ arrowtachometerinna.cpp	
Classe ArrowTachometerInna dérivée de la classe objet_virtuel	??
serveur/Inna/ arrowtachometerinna.h	??
serveur/Inna/ clignotantinna.cpp	
Classe permettant d'afficher les voyants clignotants	??
serveur/Inna/ clignotantinna.h	??
serveur/Inna/ compteurkminna.cpp	
Classe dérivant de objet_virtuel permettant un affichage en temps réel de la distance parcourue depuis le redémarrage du programme	??
serveur/Inna/ compteurkminna.h	??
serveur/Inna/ enginetinna.cpp	
Classe EngineTInna dérivée de la classe objet_virtuel	??
serveur/Inna/ enginetinna.h	??
serveur/Inna/ fuelengineinna.cpp	
Classe dérivée de QGraphicsItem permettant l'affichage des jauges de l'essence et de la température du moteur	??
serveur/Inna/ fuelengineinna.h	??
serveur/Inna/ fuellevelinna.cpp	
Classe FuelLevelInna dérivée de la classe objet_virtuel	??
serveur/Inna/ fuellevelinna.h	??
serveur/Inna/ iconinna.cpp	
Classe IconInna dérivée de la classe objet_virtuel	??
serveur/Inna/ iconinna.h	??
serveur/Inna/ icononoff.h	??
serveur/Inna/ icononoffinna.cpp	
Classe iconOnOffInna dérivée de la classe IconInna	??
serveur/Inna/ icononoffinna.h	??
serveur/Inna/ infowindowinna.cpp	
Classe InfoWindowInna dérivée de la classe QGraphicsItem	??
serveur/Inna/ infowindowinna.h	??

serveur/Inna/inna_scene.cpp	
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	??
serveur/Inna/inna_scene.h	??
serveur/Inna/oilgaugeinna.cpp	
Classe dérivée de QGraphicsItem permettant l'affichage des jauges de niveau et de température d'huile	??
serveur/Inna/oilgaugeinna.h	??
serveur/Inna/prnd_inna.cpp	
Classe PRND_Inna dérivée de la classe IconInna pour afficher les modes de la transmission	??
serveur/Inna/prnd_inna.h	??
serveur/Inna/speedometerinna.cpp	
Classe SpeedometerInna dérivée de la classe QGraphicsItem	??
serveur/Inna/speedometerinna.h	??
serveur/Inna/tachometerinna.cpp	
Classe TachometerInna dérivée de la classe QGraphicsItem	??
serveur/Inna/tachometerinna.h	??
serveur/Jonas/jonas_compteur.h	??
serveur/Jonas/jonas_compteurkm.h	??
serveur/Jonas/jonas_scene.h	??
serveur/Jonas/jonas_voyant_clignotant.h	??
serveur/Jonas/jonas_voyant_simple.h	??
serveur/Jonas/jonas_voyant_warning.h	??
serveur/Karim/cadranEss.h	??
serveur/Karim/cadrantourparmin.h	??
serveur/Karim/CadranVitesse.h	??
serveur/Karim/clignot.h	??
serveur/Karim/karim_scene.h	??
serveur/Karim/voyants.h	??
serveur/Lea/aiguille_lea.h	??
serveur/Lea/compteur_kilometre_lea.h	??
serveur/Lea/fuelgauge_lea.cpp	
Classe jauge d'essence, permettant la création d'une jauge paramétrables	??
serveur/Lea/fuelgauge_lea.h	??
serveur/Lea/jauge_temperature_lea.h	??
serveur/Lea/lea_clignotants.cpp	
Classe dérivée de objet_virtuel permettant l'affichage d'un voyant clignotant droite ou gauche	??
serveur/Lea/lea_clignotants.h	??
serveur/Lea/lea_scene.cpp	
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	??
serveur/Lea/lea_scene.h	??
serveur/Lea/lea_scenedefond.h	??
serveur/Lea/mainwindow.h	??
serveur/Lea/mygraphicitem.h	??
serveur/Lea/speedometer_lea.cpp	
Classe speedometer, permettant la création du compteur vitesse paramétrable avec aiguille	??
serveur/Lea/speedometer_lea.h	??
serveur/Lea/tachometergauge_lea.cpp	
Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille	??
serveur/Lea/tachometergauge_lea.h	??
serveur/Lea/voyant_lea.cpp	
Classe permettant l'affichage de voyants dont l'état est allumé ou éteint	??
serveur/Lea/voyant_lea.h	??
serveur/Lea/warning_lea.cpp	
Classe dérivée de objet_virtuel permettant l'affichage d'un voyant warning clignotant	??
serveur/Lea/warning_lea.h	??
serveur/Leo/leo_display.h	??

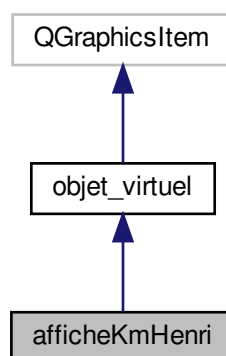
serveur/Leo/ leo_gauge.h	??
serveur/Leo/ leo_indicator.h	??
serveur/Leo/ leo_label.h	??
serveur/Leo/ leo_scene.h	??
serveur/Loto/ fuel_guage.h	??
serveur/Loto/ leds.h	??
serveur/Loto/ loto_dashboard.h	??
serveur/Loto/ loto_scene.h	??
serveur/Loto/ tachometre.h	??
serveur/Loto/ tempgauge.h	??
serveur/Youcef/ 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

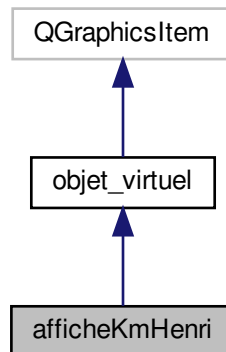
Class Documentation

6.1 afficheKmHenri Class Reference

Inheritance diagram for afficheKmHenri:



Collaboration diagram for afficheKmHenri:



Public Member Functions

- **afficheKmHenri** ([objet_virtuel](#) *parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)

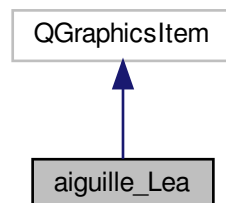
Additional Inherited Members

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

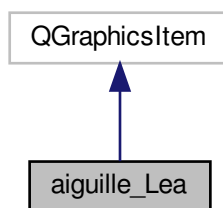
- serveur/Henri/affichekmhenri.h
- serveur/Henri/affichekmhenri.cpp

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

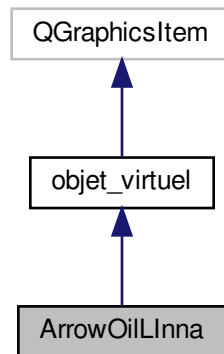
- double **x**
- double **y**
- double **r**
- double **taille_aiguille**
- int **vitesse**
- int **vitesse_max**
- int **angle_depart** =0
- int **angle_fin** =0
- int **angle** =0
- const double **pi** =3.14159265359

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

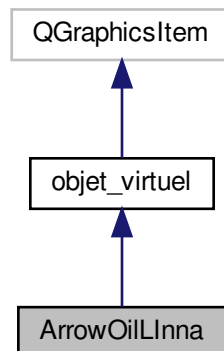
- serveur/Lea/aiguille_lea.h
- serveur/Lea/aiguille_lea.cpp

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

dessin du circle central

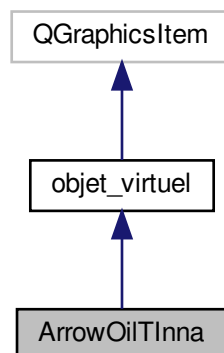
dessin de la fleche

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

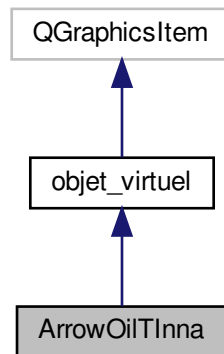
- serveur/lnna/arrowsoilinna.h
- serveur/lnna/arrowsoilinna.cpp

6.4 ArrowOilTInna Class Reference

Inheritance diagram for ArrowOilTInna:



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

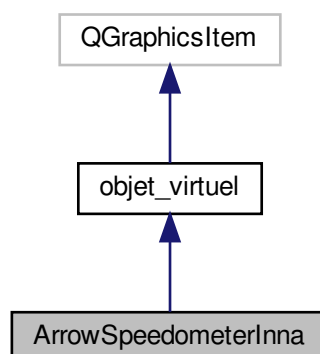
Dessin de la fleche

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

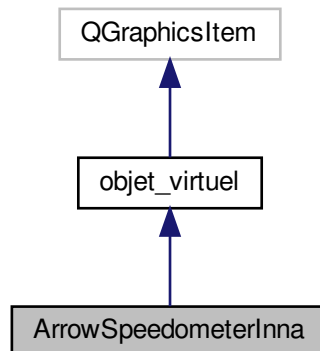
- serveur/Inna/arrowoiltinna.h
- serveur/Inna/[arrowoiltinna.cpp](#)

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 circle central

dessin de la fleche

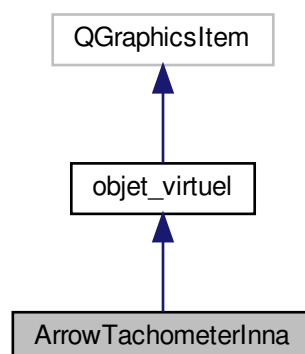
Affichage de la vitesse actuelle

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

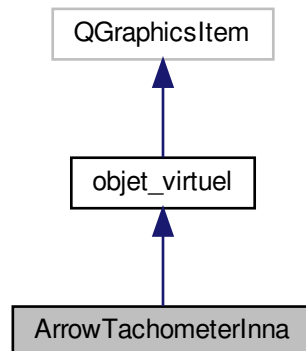
- serveur/Inna/arrowspeedometerinna.h
- serveur/Inna/[arrowspeedometerinna.cpp](#)

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

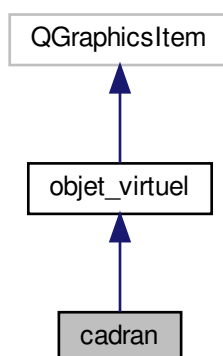
dessin de la fleche

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

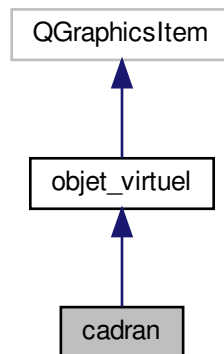
- serveur/lnna/arrowtachometerinna.h
- serveur/lnna/[arrowtachometerinna.cpp](#)

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**

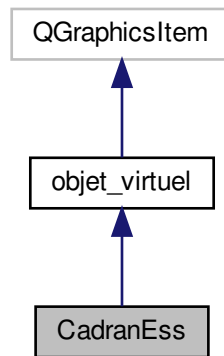
Additional Inherited Members

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

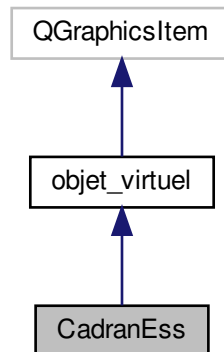
- serveur/Harout/cadran.h
- serveur/Harout/cadran.cpp

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)

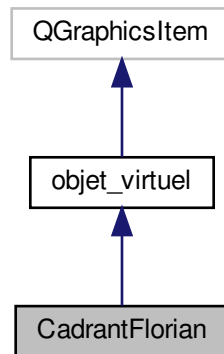
Additional Inherited Members

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

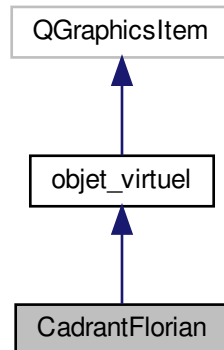
- serveur/Karim/cadranEss.h
- serveur/Karim/cadranEss.cpp

6.9 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, 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

6.9.1 Constructor & Destructor Documentation

6.9.1.1 CadrantFlorian()

```
CadrantFlorian::CadrantFlorian (
    bool hasText = true,
    bool hasSubTrait = true,
    int invertAiguille = 1,
    int pas = 10,
    int angleB = 220,
    int angleE = 260,
    int valeurMax = 260,
    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>	

6.9.2 Member Function Documentation

6.9.2.1 boundingRect()

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

[CadrantFlorian::boundingRect.](#)

Returns

retourne un rectangle qui encadre l'objet

6.9.2.2 generateAngle()

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

[CadrantFlorian::generateAngle.](#)

Returns

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

6.9.2.3 paint()

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

[CadrantFlorian::paint.](#)

Parameters

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

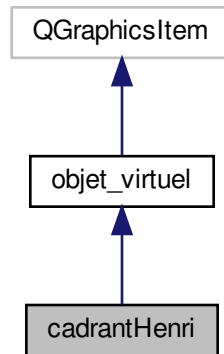
dessine les différents éléments du compteur : aiguille, compteur, chiffre...

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

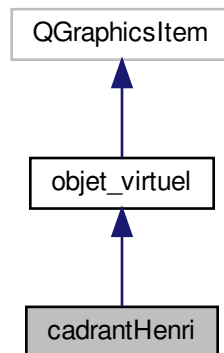
- serveur/Florian/cadrantflorian.h
- serveur/Florian/cadrantflorian.cpp

6.10 cadrantHenri Class Reference

Inheritance diagram for cadrantHenri:



Collaboration diagram for cadrantHenri:



Public Member Functions

- **cadrantHenri** ([objet_virtuel](#) *parent=nullptr)
- `QRectF` **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)

Public Attributes

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

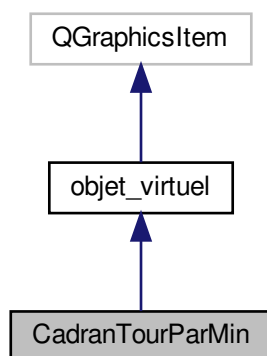
Additional Inherited Members

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

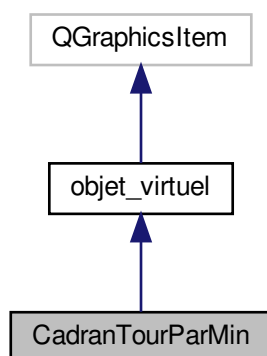
- serveur/Henri/cadranthenri.h
- serveur/Henri/cadranthenri.cpp

6.11 CadranTourParMin Class Reference

Inheritance diagram for CadranTourParMin:



Collaboration diagram for CadranTourParMin:



Public Member Functions

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

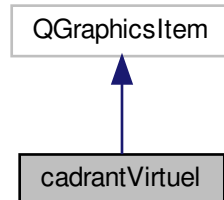
Additional Inherited Members

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

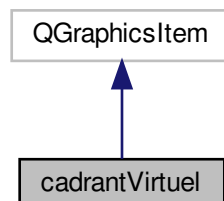
- serveur/Karim/cadrantourparmin.h
- serveur/Karim/cadrantourparmin.cpp

6.12 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

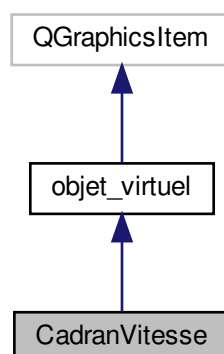
- int **valeur**
- int **valeurMax**

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

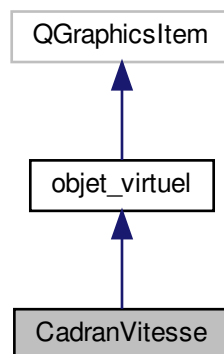
- serveur/Florian/cadrantvirtuel.h
- serveur/Florian/cadrantvirtuel.cpp

6.13 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)`

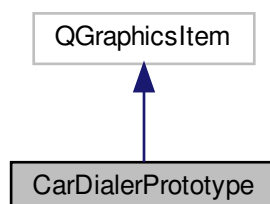
Additional Inherited Members

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

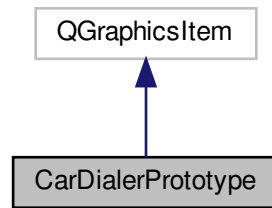
- `serveur/Karim/CadranVitesse.h`
- `serveur/Karim/CadranVitesse.cpp`

6.14 CarDialerPrototype Class Reference

Inheritance diagram for CarDialerPrototype:



Collaboration diagram for CarDialerPrototype:



Public Member Functions

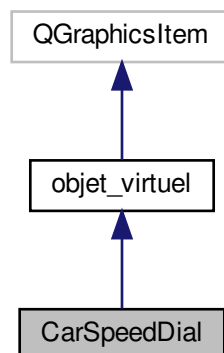
- **CarDialerPrototype** (`QGraphicsItem *parent=nullptr`)
- `QRectF boundingRect () const`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`

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

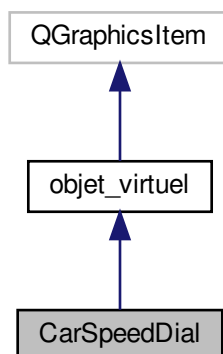
- `serveur/Youcef/cardialerprototype.h`
- `serveur/Youcef/cardialerprototype.cpp`

6.15 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

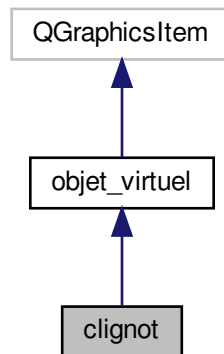
Additional Inherited Members

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

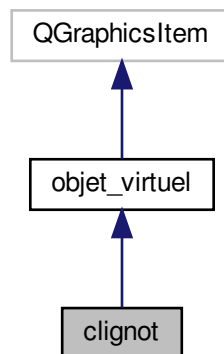
- serveur/Youcef/carspeeddial.h
- serveur/Youcef/carspeeddial.cpp

6.16 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**

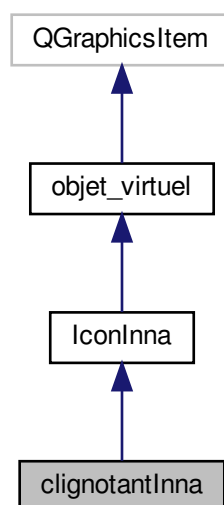
Additional Inherited Members

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

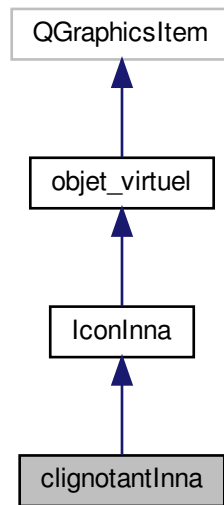
- serveur/Karim/clignot.h
- serveur/Karim/clignot.cpp

6.17 clignotantInna Class Reference

Inheritance diagram for clignotantInna:



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

6.17.1 Constructor & Destructor Documentation

6.17.1.1 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.

6.17.2 Member Function Documentation

6.17.2.1 paint()

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

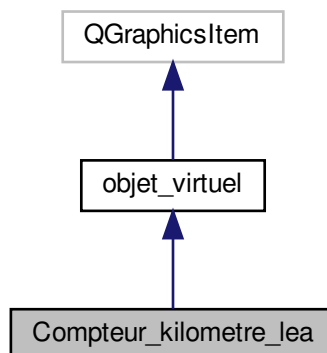
Va chercher la valeur de la variable *value* de la classe [clignotantInna](#). *value* = 1 correspond à l'affichage du clignotant droit *value* = -1 correspond à l'affichage du clignotant gauche *value* = 2 correspond à l'affichage des clignotants droit et gauche au même temps

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

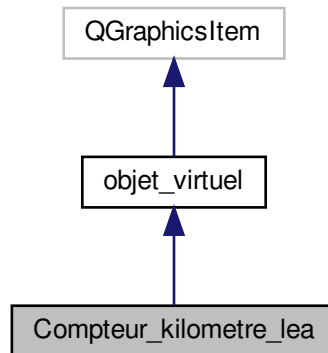
- serveur/Inna/clignotantinna.h
- serveur/Inna/[clignotantinna.cpp](#)

6.18 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)
- `QRectF` **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)

Public Attributes

- int **x**
- int **y**

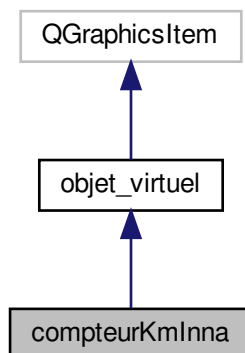
Additional Inherited Members

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

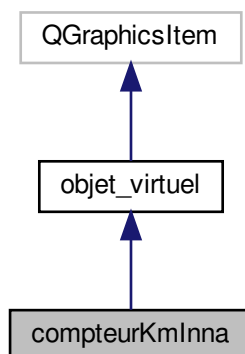
- serveur/Lea/compteur_kilometre_lea.h
- serveur/Lea/compteur_kilometre_lea.cpp

6.19 compteurKmlInna Class Reference

Inheritance diagram for compteurKmlInna:



Collaboration diagram for compteurKmlInna:



Public Member Functions

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

Additional Inherited Members

6.19.1 Constructor & Destructor Documentation

6.19.1.1 `compteurKmlInna()`

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

[compteurKmlInna::compteurKmlInna](#)

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

6.19.2 Member Function Documentation

6.19.2.1 `paint()`

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

[compteurKmlInna::paint](#)

Parameters

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

La fonction paint permet de réaliser un affichage pour le compteur kilometrique.

La valeur *value* est envoyer par la fonction **MainWindow::update_km()** dans le fichier *mainwindow.cpp*. Affichage de la valeur du kilométrage.

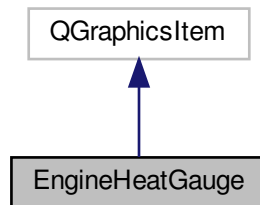
Affichage d'un rectangle avec gradient.

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

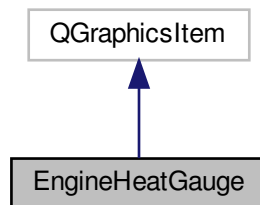
- serveur/lnna/compteurkminna.h
- serveur/lnna/[compteurkminna.cpp](#)

6.20 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

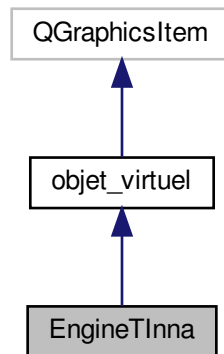
- float **pi** = 3.14159265359

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

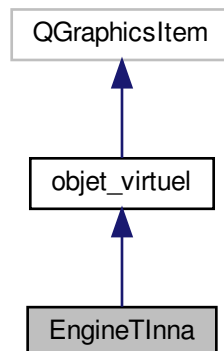
- `serveur/Youcef/engineheatgauge.h`
- `serveur/Youcef/engineheatgauge.cpp`

6.21 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

6.21.1 Member Function Documentation

6.21.1.1 paint()

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

Dessin du circle central

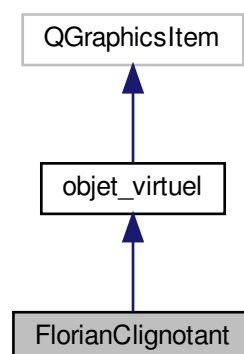
Dessin de la fleche

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

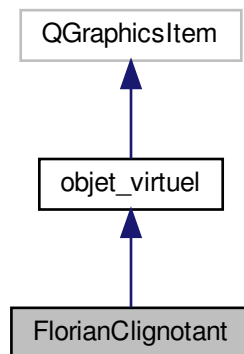
- serveur/lnna/engineinna.h
- serveur/lnna/[engineinna.cpp](#)

6.22 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

6.22.1 Constructor & Destructor Documentation

6.22.1.1 FlorianClignotant()

```

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

[FlorianClignotant::FlorianClignotant.](#)

Parameters

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

6.22.2 Member Function Documentation

6.22.2.1 boundingRect()

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

[FlorianClignotant::boundingRect](#).

Returns

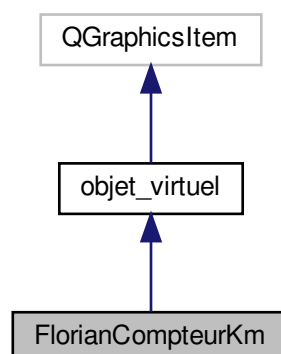
retourne un rectangle qui encadre l'objet

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

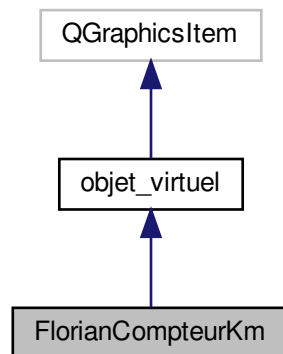
- serveur/Florian/florianclignotant.h
- serveur/Florian/florianclignotant.cpp

6.23 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

6.23.1 Constructor & Destructor Documentation

6.23.1.1 FlorianCompteurKm()

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

[FlorianCompteurKm::FlorianCompteurKm.](#)

Parameters

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

6.23.2 Member Function Documentation

6.23.2.1 boundingRect()

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

[FlorianCompteurKm::boundingRect.](#)

Returns

retourne un rectangle qui encadre l'objet

6.23.2.2 paint()

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

[FlorianCompteurKm::paint.](#)

Parameters

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

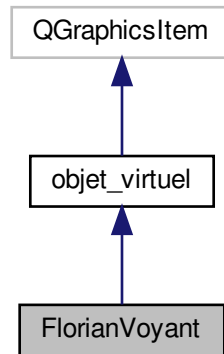
dessine le compteur km avec une police d'écriture digital

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

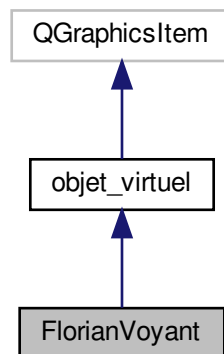
- serveur/Florian/floriancompteurkm.h
- serveur/Florian/floriancompteurkm.cpp

6.24 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

6.24.1 Member Function Documentation

6.24.1.1 boundingRect()

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

[FlorianVoyant::boundingRect.](#)

Returns

retourne le rectangle qui encadre l'objet

6.24.1.2 paint()

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

[FlorianVoyant::paint.](#)

Parameters

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

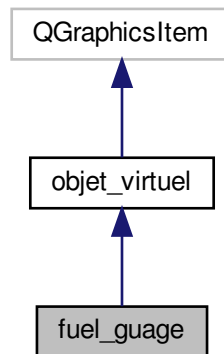
si value == 1, dessine l'image qui est en attribut

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

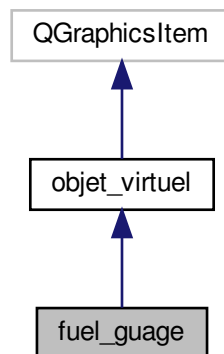
- serveur/Florian/florianvoyant.h
- serveur/Florian/florianvoyant.cpp

6.25 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

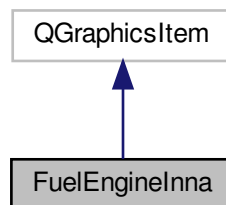
Additional Inherited Members

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

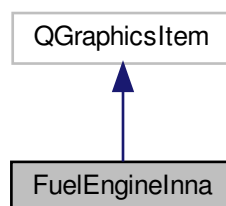
- serveur/Loto/fuel_guage.h
- serveur/Loto/fuel_guage.cpp

6.26 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**

6.26.1 Member Function Documentation

6.26.1.1 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

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

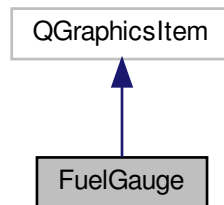
Affichage des traits et du texte pour le niveau d'essence.

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

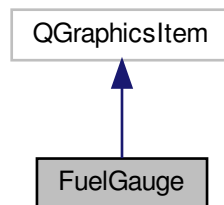
- serveur/Inna/fuelengineinna.h
- serveur/Inna/[fuelengineinna.cpp](#)

6.27 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

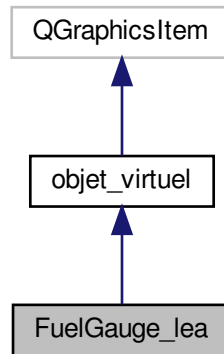
- float **pi** = 3.14159265359

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

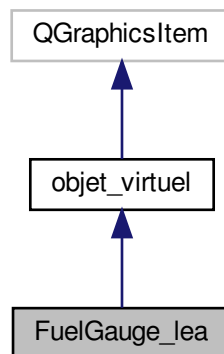
- serveur/Youcef/fuelgauge.h
- serveur/Youcef/fuelgauge.cpp

6.28 FuelGauge_lea Class Reference

Inheritance diagram for FuelGauge_lea:



Collaboration diagram for FuelGauge_lea:



Public Member Functions

- [FuelGauge_lea](#) (double, double, double, double, double, double, double, double)
FuelGauge_lea::FuelGauge_lea, Constructeur de la classe, permet d'initialiser tous les paramètres.
- `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

6.28.1 Constructor & Destructor Documentation

6.28.1.1 FuelGauge_lea()

```
FuelGauge_lea::FuelGauge_lea (
    double param_x,
    double param_y,
    double param_width,
    double param_height,
    double param_x2,
    double param_y2,
    double param_x3,
    double param_y3 )
```

[FuelGauge_lea::FuelGauge_lea](#), Constructeur de la classe, permet d'initialiser tous les paramètres.

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
<i>param_x2</i>	position horizontale du coin en haut à gauche de la position de la lettre F (Full)
<i>param_y2</i>	position verticale du coin en haut à gauche de la position de la lettre F (Full)
<i>param_x3</i>	position horizontale du coin en haut à gauche de la position de la lettre E (Empty)
<i>param_y3</i>	position verticale coin du coin en haut à gauche de la position de la lettre F (Empty)

6.28.2 Member Function Documentation

6.28.2.1 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.

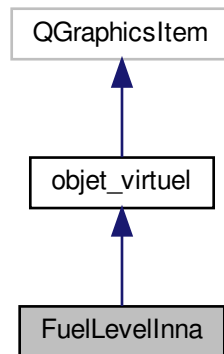
Les voyants sont obtenus via un drawPixmap dans une boucle if pour changer le pixmap en fonction de la "value" d'essence(devient rouge quand le niveau passe en dessous de 25%).

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

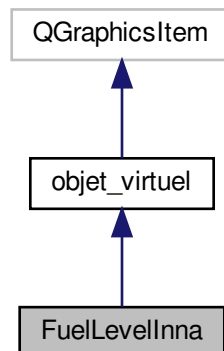
- serveur/Lea/fuelgauge_lea.h
- serveur/Lea/[fuelgauge_lea.cpp](#)

6.29 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

6.29.1 Member Function Documentation

6.29.1.1 paint()

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

dessin du cercle central

dessin de la fleche

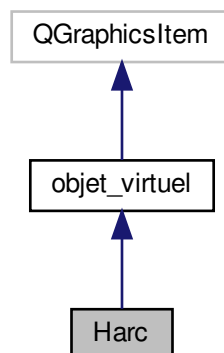
dessin de l'icone du niveau d'essence

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

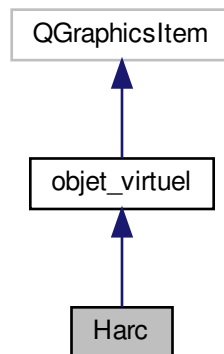
- serveur/lnna/fuellevelinna.h
- serveur/lnna/[fuellevelinna.cpp](#)

6.30 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)

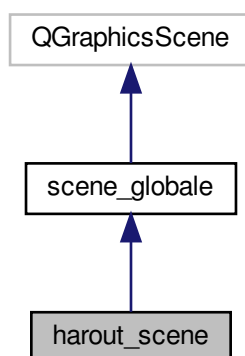
Additional Inherited Members

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

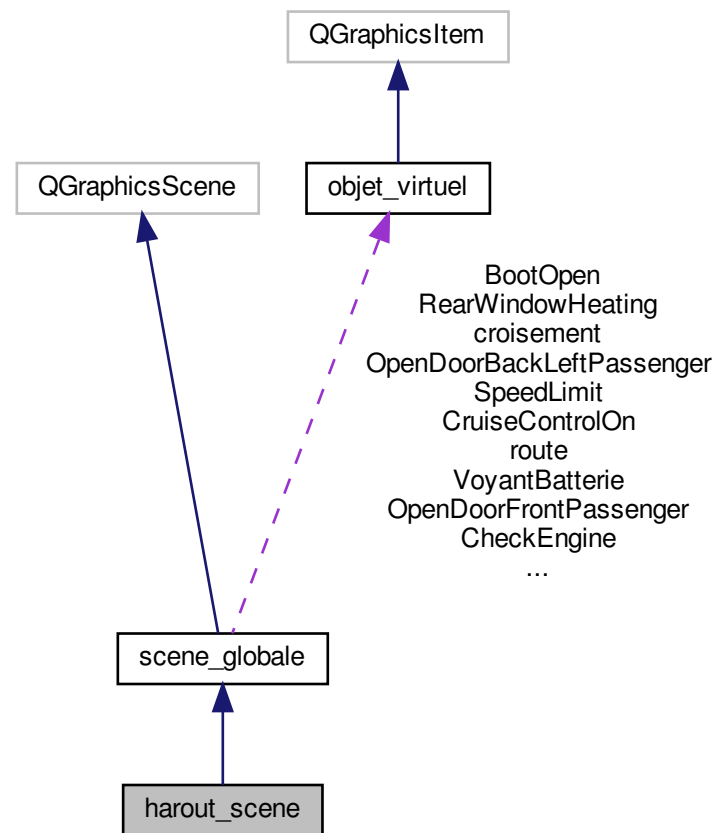
- serveur/Harout/harc.h
- serveur/Harout/harc.cpp

6.31 harout_scene Class Reference

Inheritance diagram for harout_scene:



Collaboration diagram for harout_scene:



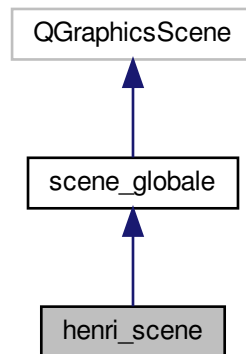
Additional Inherited Members

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

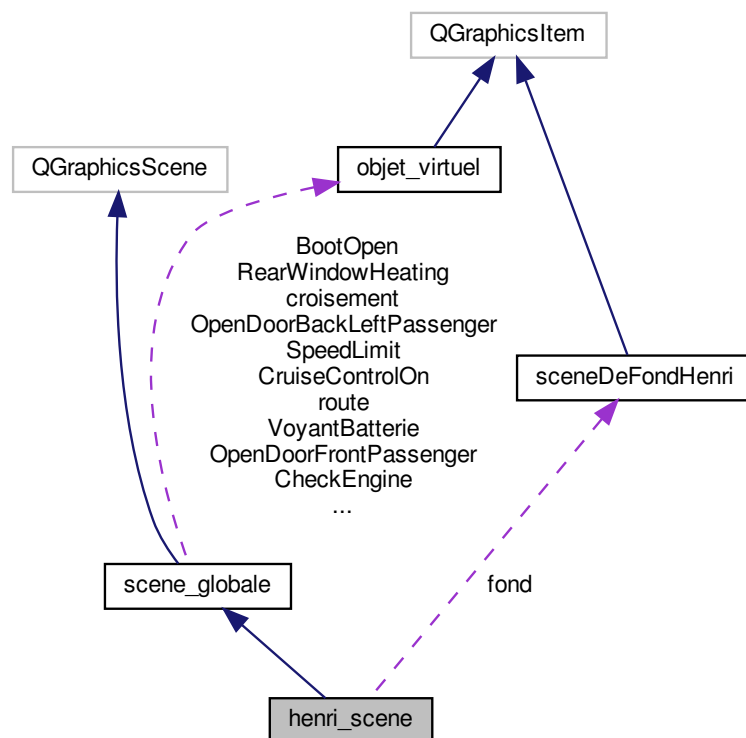
- serveur/Harout/harout_scene.h
- serveur/Harout/harout_scene.cpp

6.32 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

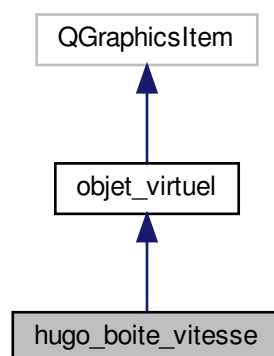
- [sceneDeFondHenri](#) * **fond**

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

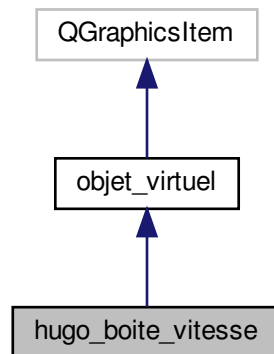
- serveur/Henri/henri_scene.h
- serveur/Henri/henri_scene.cpp

6.33 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

6.33.1 Constructor & Destructor Documentation

6.33.1.1 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)

6.33.2 Member Function Documentation

6.33.2.1 paint()

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

[hugo_ecran::paint](#)

Parameters

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

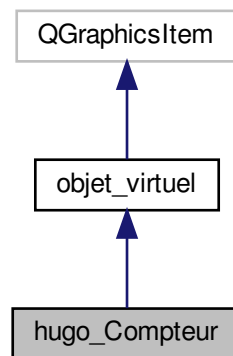
La fonction paint va permettre l'affichage des différents modes P,R,N,D,3,2,L suivant un arc de cercle autour du compteur vitesse. L'affichage se fait par l'intermédiaire d'une boucle. Si l'itérateur correspond à value, le mode de transmission est affiché en vert avec un effet de halo autour.

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

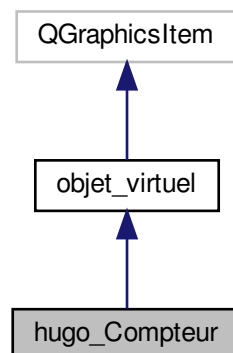
- serveur/Hugo/hugo_boite_vitesse.h
- serveur/Hugo/[hugo_boite_vitesse.cpp](#)

6.34 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 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

6.34.1 Constructor & Destructor Documentation

6.34.1.1 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
<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>	

6.34.2 Member Function Documentation

6.34.2.1 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

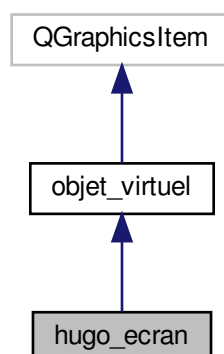
L'effet de verre est obtenu en superposant un cercle semi-transparent au compteur. L'option `direction_grad` permet de définir le sens du gradient.

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

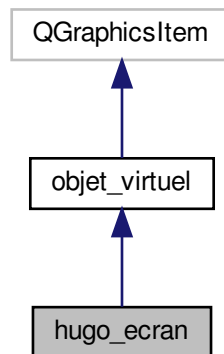
- `serveur/Hugo/hugo_compteur.h`
- `serveur/Hugo/hugo_compteur.cpp`

6.35 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

6.35.1 Constructor & Destructor Documentation

6.35.1.1 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.

6.35.2 Member Function Documentation

6.35.2.1 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

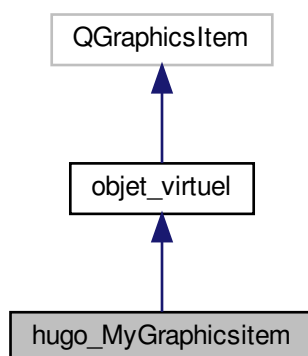
Affichage des différents textes. Le kilométrage est récupéré grâce à la fonction getValue.

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

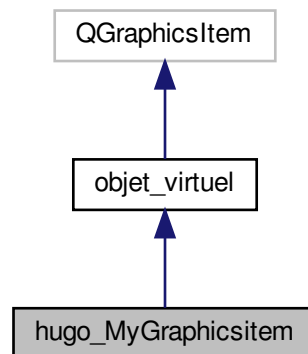
- serveur/Hugo/hugo_ecran.h
- serveur/Hugo/[hugo_ecran.cpp](#)

6.36 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

6.36.1 Constructor & Destructor Documentation

6.36.1.1 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.

6.36.2 Member Function Documentation

6.36.2.1 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

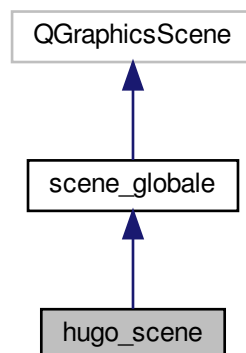
Affichage d'une icone jauge d'essence

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

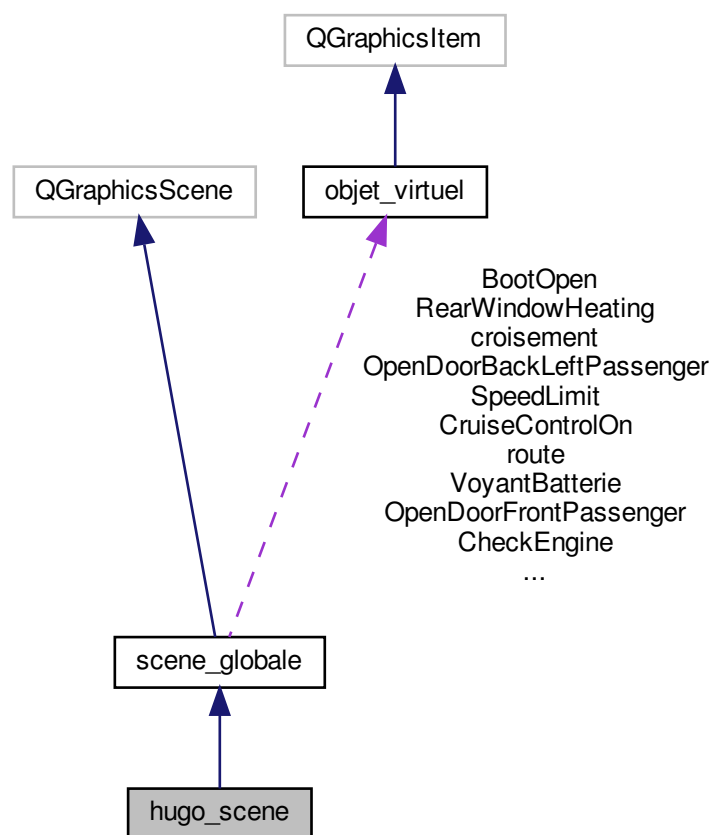
- serveur/Hugo/hugo_mygraphicsitem.h
- serveur/Hugo/hugo_mygraphicsitem.cpp

6.37 hugo_scene Class Reference

Inheritance diagram for hugo_scene:



Collaboration diagram for hugo_scene:



Public Member Functions

- **hugo_scene** ([scene_globale](#) *parent=nullptr)

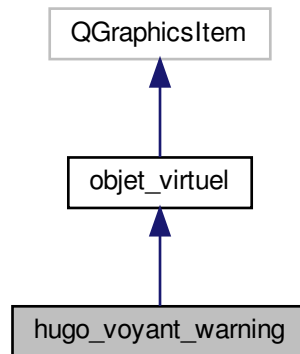
Additional Inherited Members

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

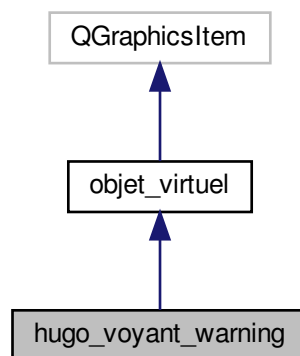
- serveur/Hugo/hugo_scene.h
- serveur/Hugo/hugo_scene.cpp

6.38 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

6.38.1 Constructor & Destructor Documentation

6.38.1.1 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.

6.38.2 Member Function Documentation

6.38.2.1 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

6.38.2.2 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.

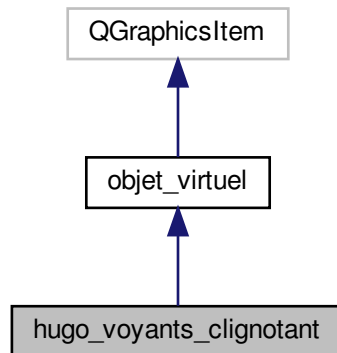
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

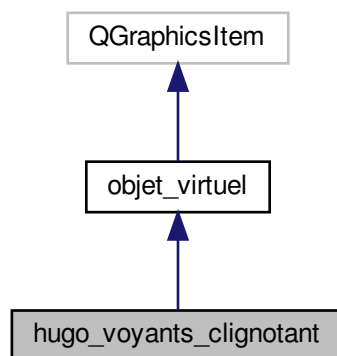
- serveur/Hugo/hugo_voyant_warning.h
- serveur/Hugo/[hugo_voyant_warning.cpp](#)

6.39 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

6.39.1 Constructor & Destructor Documentation

6.39.1.1 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.

6.39.2 Member Function Documentation

6.39.2.1 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.

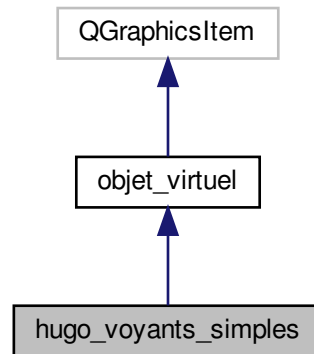
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné

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

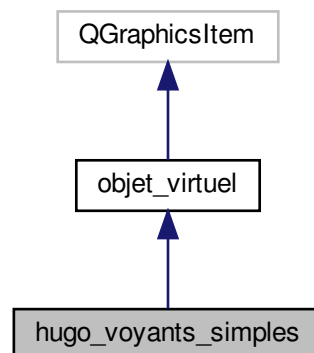
- serveur/Hugo/hugo_voyants_clignotant.h
- serveur/Hugo/hugo_voyants_clignotant.cpp

6.40 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

6.40.1 Constructor & Destructor Documentation

6.40.1.1 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

6.40.2 Member Function Documentation

6.40.2.1 paint()

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

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

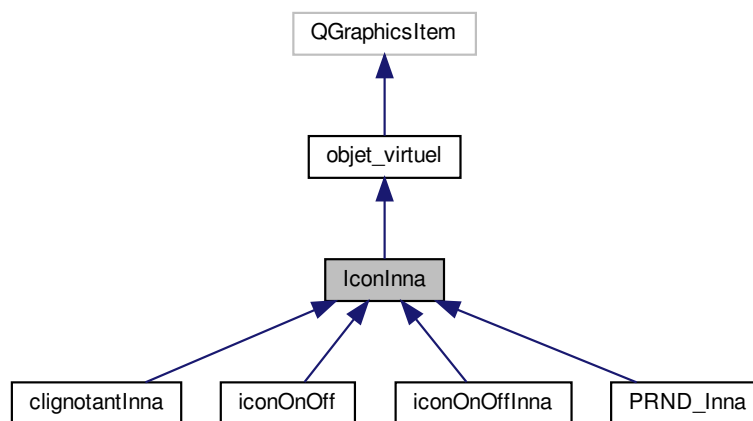
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché.

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

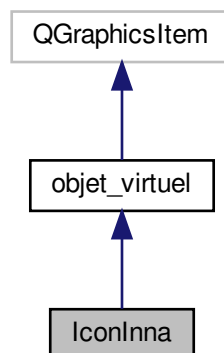
- serveur/Hugo/hugo_voyants_simples.h
- serveur/Hugo/[hugo_voyants_simples.cpp](#)

6.41 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

6.41.1 Member Function Documentation

6.41.1.1 [init\(\)](#)

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

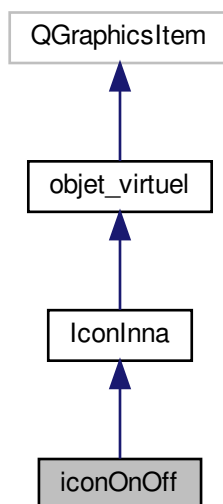
La fonction [init\(QPoint p, QSize s, QString pth\)](#) permet de reinitialiser un voyant.

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

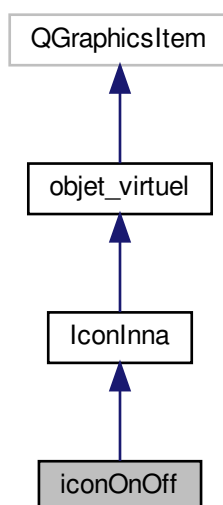
- serveur/Inna/iconinna.h
- serveur/Inna/[iconinna.cpp](#)

6.42 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)

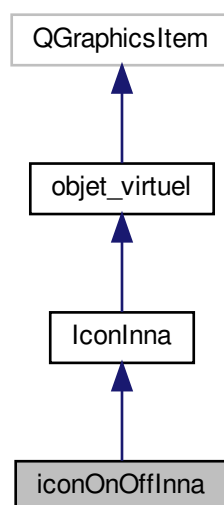
Additional Inherited Members

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

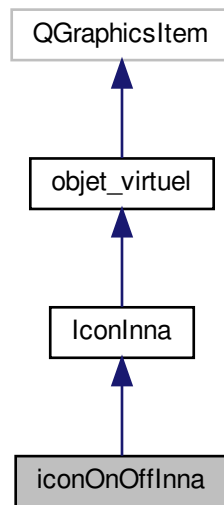
- serveur/lnna/icononoff.h
- serveur/lnna/icononoff.cpp

6.43 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)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)

Additional Inherited Members

6.43.1 Constructor & Destructor Documentation

6.43.1.1 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.

6.43.2 Member Function Documentation

6.43.2.1 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* :

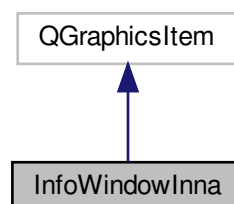
Si *value* = 1 => l'etait *allumé* , sinon l'etait *etaint*.

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

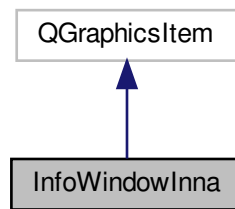
- serveur/Inna/icononoffinna.h
- serveur/Inna/[icononoffinna.cpp](#)

6.44 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**

6.44.1 Member Function Documentation

6.44.1.1 paint()

```

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

[InfoWindowInna::paint.](#)

Parameters

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

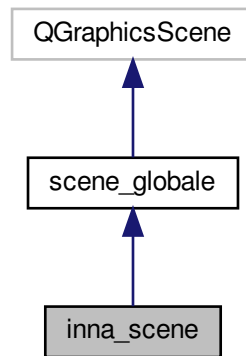
La fonction crée un objet static qui sert comme un fond pour des voyants et des messages d'info.

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

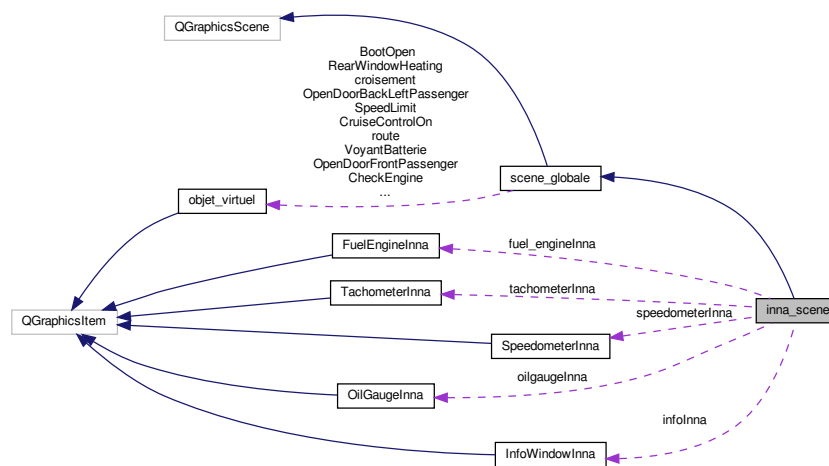
- serveur/Inna/infowindowinna.h
- serveur/Inna/[infowindowinna.cpp](#)

6.45 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

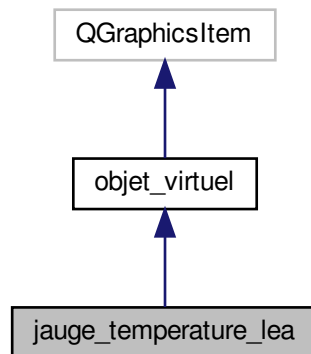
- [SpeedometerInna](#) * **speedometerInna**
- [TachometerInna](#) * **tachometerInna**
- [OilGaugeInna](#) * **oilgaugeInna**
- [InfoWindowInna](#) * **infoInna**
- [FuelEngineInna](#) * **fuel_engineInna**

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

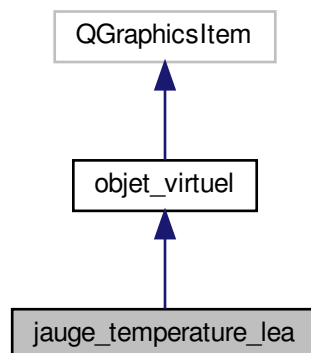
- [serveur/Inna/inna_scene.h](#)
- [serveur/Inna/inna_scene.cpp](#)

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

Public Attributes

- const double `pi` =3.14159265359
- double `x`
- double `y`
- double `r`
- int `angle_debut`
- int `angle_fin`
- int `span_angle`
- int `temperature_max` =130
- int `nb_graduation` =28
- int `v`

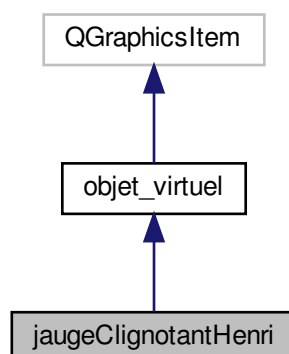
Additional Inherited Members

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

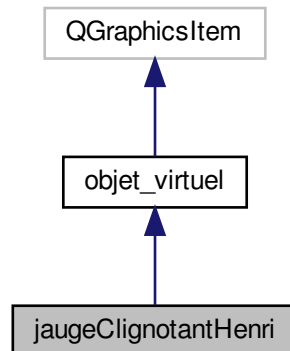
- `serveur/Lea/jauge_temperature_lea.h`
- `serveur/Lea/jauge_temperature_lea.cpp`

6.47 `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**

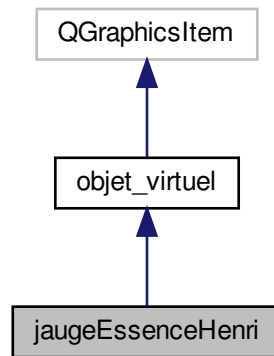
Additional Inherited Members

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

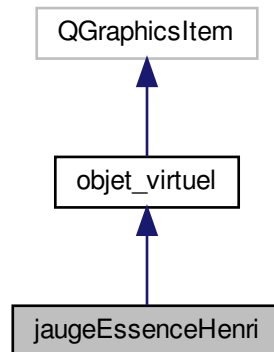
- `serveur/Henri/jaugeclignotantthenri.h`
- `serveur/Henri/jaugeclignotantthenri.cpp`

6.48 `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)

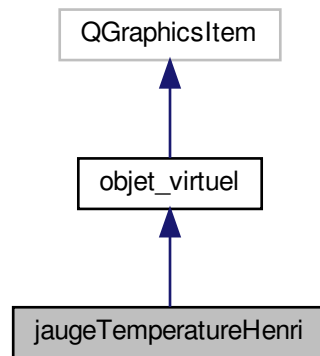
Additional Inherited Members

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

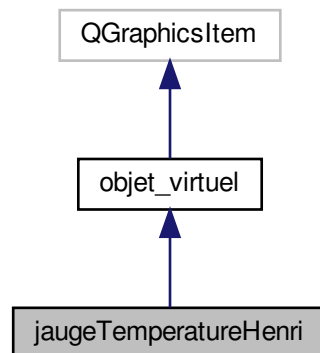
- `serveur/Henri/jaugeessencehenri.h`
- `serveur/Henri/jaugeessencehenri.cpp`

6.49 `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)`

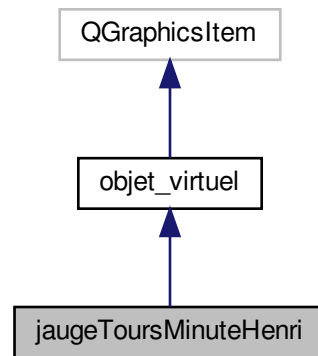
Additional Inherited Members

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

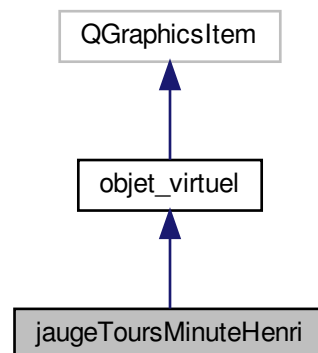
- `serveur/Henri/jaugetemperaturehenri.h`
- `serveur/Henri/jaugetemperaturehenri.cpp`

6.50 `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`

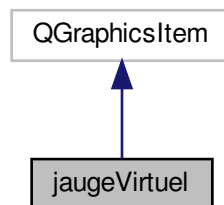
Additional Inherited Members

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

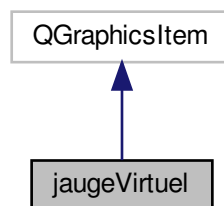
- serveur/Henri/jaugetoursminutehenri.h
- serveur/Henri/jaugetoursminutehenri.cpp

6.51 `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

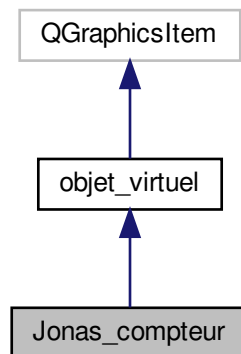
- int **valeur**
- int **valeurMax**

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

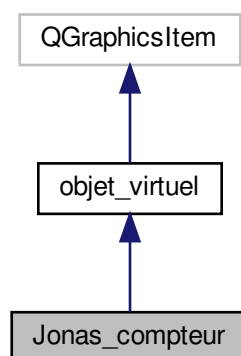
- serveur/Florian/jaugevirtuel.h
- serveur/Florian/jaugevirtuel.cpp

6.52 Jonas_compteur Class Reference

Inheritance diagram for Jonas_compteur:



Collaboration diagram for Jonas_compteur:



Public Member Functions

- **Jonas_compteur** (int max, QStringList gradList, float startAngle, float endAngle, int critic, QString text← Center, int ngrad, bool line=0, int mod=2, int size=150)
- QRectF **boundingRect** () const override
- void **paint** (QPainter *, const QStyleOptionGraphicsItem *, QWidget *) override
- 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)

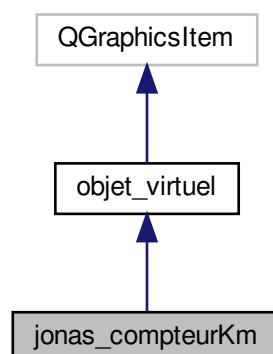
Additional Inherited Members

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

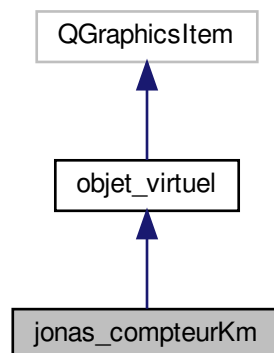
- serveur/Jonas/jonas_compteur.h
- serveur/Jonas/jonas_compteur.cpp

6.53 jonas_compteurKm Class Reference

Inheritance diagram for jonas_compteurKm:



Collaboration diagram for jonas_compteurKm:



Public Member Functions

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

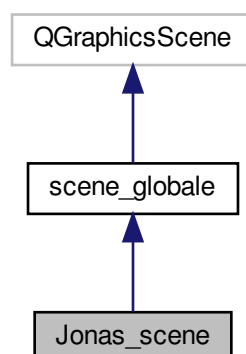
Additional Inherited Members

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

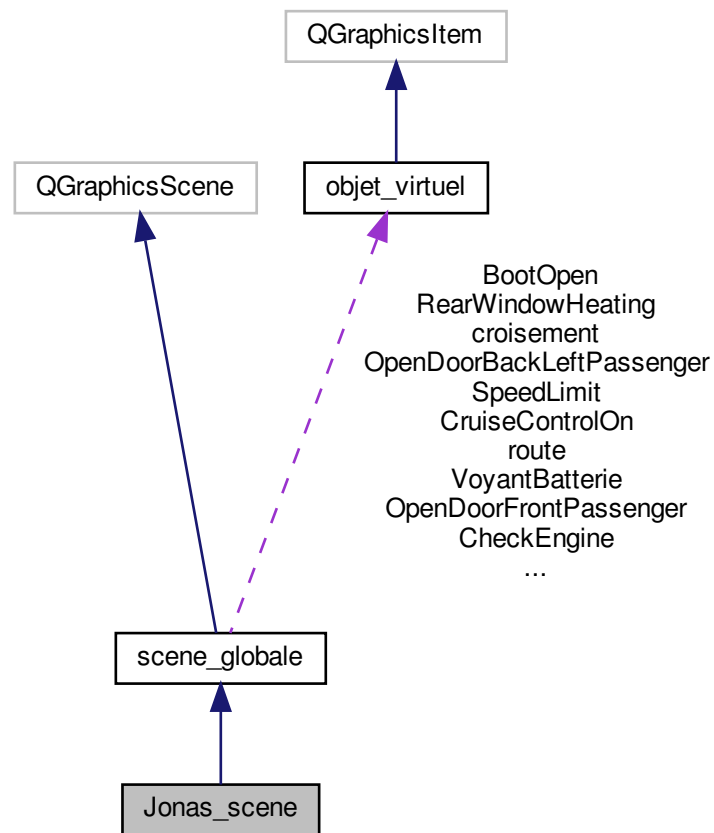
- serveur/Jonas/jonas_compteurkm.h
- serveur/Jonas/jonas_compteurkm.cpp

6.54 Jonas_scene Class Reference

Inheritance diagram for Jonas_scene:



Collaboration diagram for Jonas_scene:



Public Member Functions

- **Jonas_scene** ([scene_globale](#) *parent=nullptr)

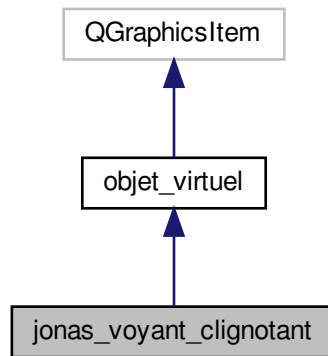
Additional Inherited Members

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

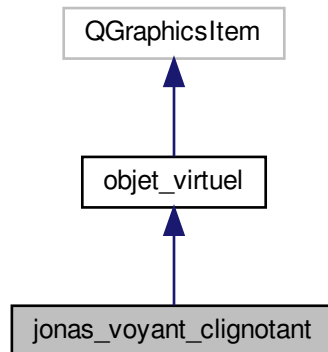
- `serveur/Jonas/jonas_scene.h`
- `serveur/Jonas/jonas_scene.cpp`

6.55 jonas_voyant_clignotant Class Reference

Inheritance diagram for jonas_voyant_clignotant:



Collaboration diagram for jonas_voyant_clignotant:



Public Member Functions

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

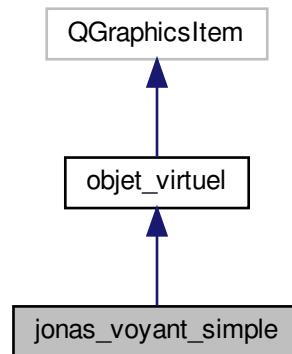
Additional Inherited Members

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

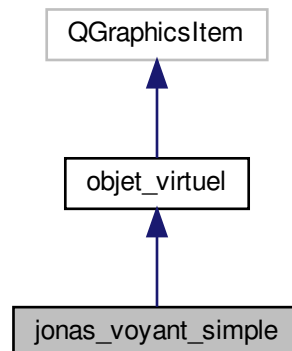
- serveur/Jonas/jonas_voyant_clignotant.h
- serveur/Jonas/jonas_voyant_clignotant.cpp

6.56 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
- void **paint** (QPainter *, const QStyleOptionGraphicsItem *, QWidget *) override
- int **getValue** ()

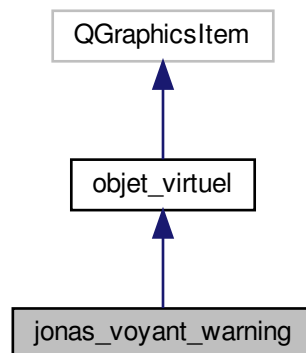
Additional Inherited Members

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

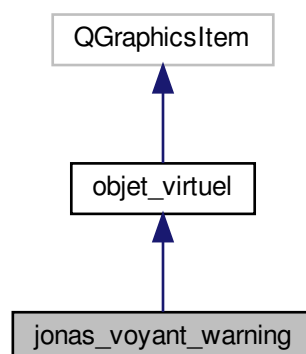
- serveur/Jonas/jonas_voyant_simple.h
- serveur/Jonas/jonas_voyant_simple.cpp

6.57 jonas_voyant_warning Class Reference

Inheritance diagram for jonas_voyant_warning:



Collaboration diagram for jonas_voyant_warning:



Public Member Functions

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

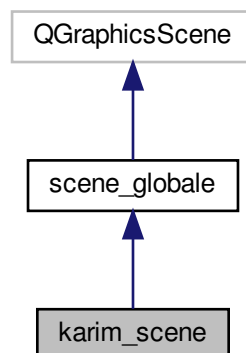
Additional Inherited Members

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

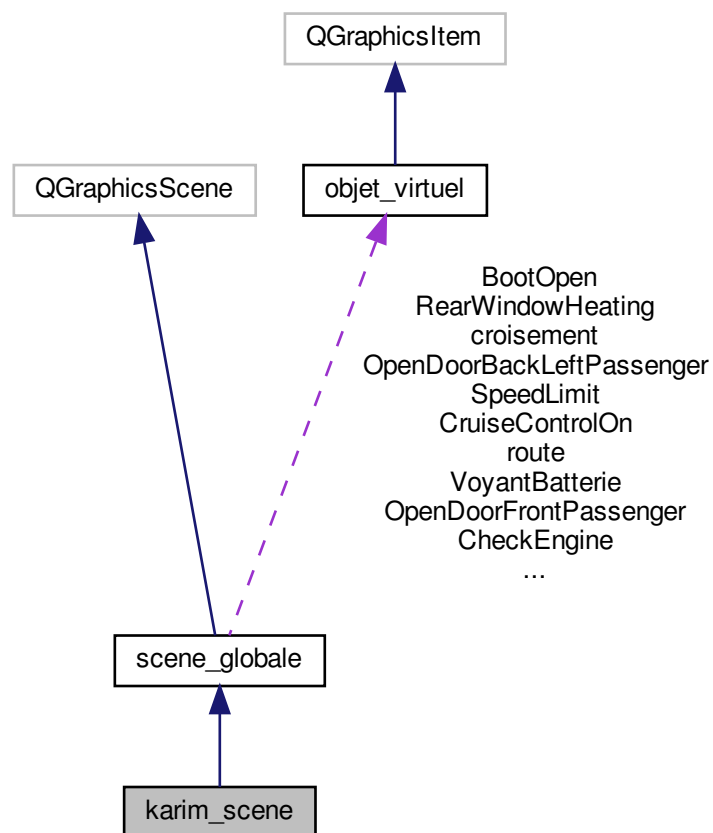
- serveur/Jonas/jonas_voyant_warning.h
- serveur/Jonas/jonas_voyant_warning.cpp

6.58 karim_scene Class Reference

Inheritance diagram for karim_scene:



Collaboration diagram for karim_scene:



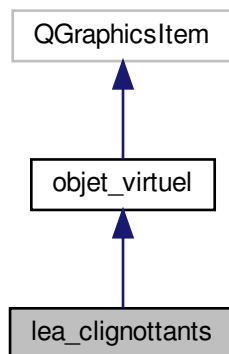
Additional Inherited Members

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

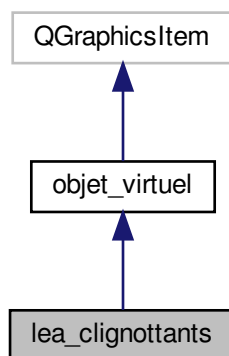
- `serveur/Karim/karim_scene.h`
- `serveur/Karim/karim_scene.cpp`

6.59 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

6.59.1 Constructor & Destructor Documentation

6.59.1.1 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.

6.59.2 Member Function Documentation

6.59.2.1 MAJ()

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

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

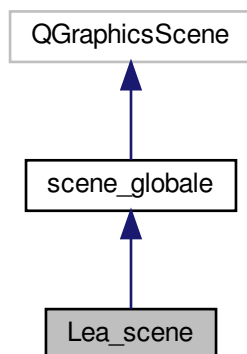
La valeur cligno controle l'opacité du painter, permettant de le rendre visible ou non. La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

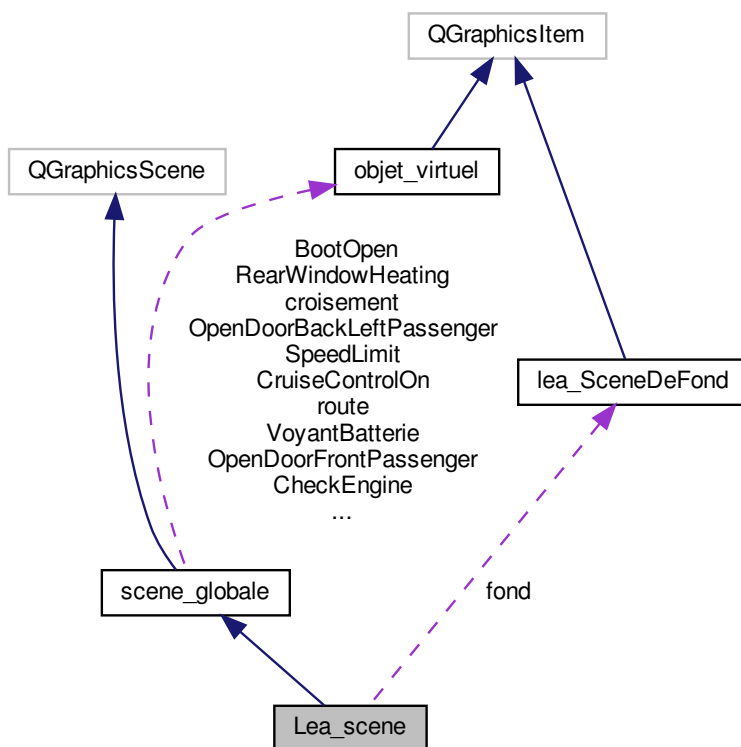
- serveur/Lea/lea_clignottants.h
- serveur/Lea/[lea_clignottants.cpp](#)

6.60 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

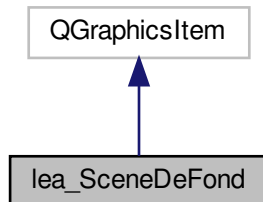
- [lea_SceneDeFond](#) * **fond**

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

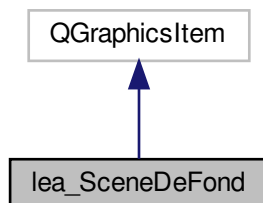
- serveur/Lea/lea_scene.h
- serveur/Lea/[lea_scene.cpp](#)

6.61 lea_SceneDeFond Class Reference

Inheritance diagram for lea_SceneDeFond:



Collaboration diagram for lea_SceneDeFond:



Public Member Functions

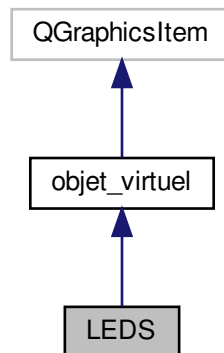
- **lea_SceneDeFond** (QGraphicsItem *parent=nullptr)
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- QRectF **boundingRect** () const

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

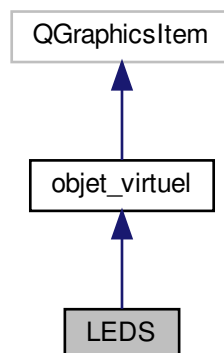
- serveur/Lea/lea_scenedefond.h
- serveur/Lea/lea_scenedefond.cpp

6.62 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

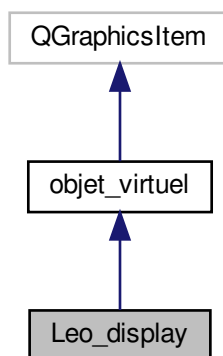
Additional Inherited Members

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

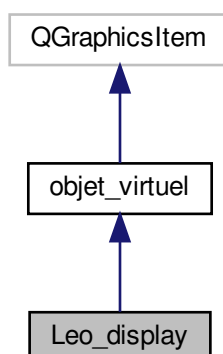
- serveur/Loto/leds.h
- serveur/Loto/leds.cpp

6.63 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

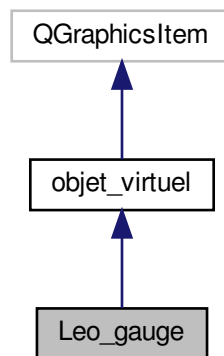
Additional Inherited Members

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

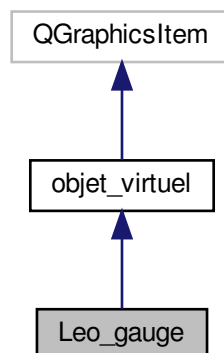
- serveur/Leo/leo_display.h
- serveur/Leo/leo_display.cpp

6.64 Leo_gauge Class Reference

Inheritance diagram for Leo_gauge:



Collaboration diagram for Leo_gauge:



Public Member Functions

- **Leo_gauge** (int maxValue, int spacing, qreal sizePx, [objet_virtuel](#) *parent=nullptr)
- QRectF **boundingRect** () const override
- void **paint** (QPainter *, const QStyleOptionGraphicsItem *, QWidget *) override

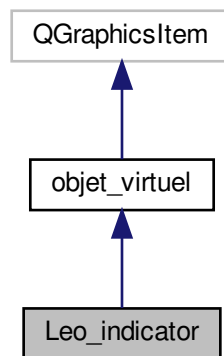
Additional Inherited Members

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

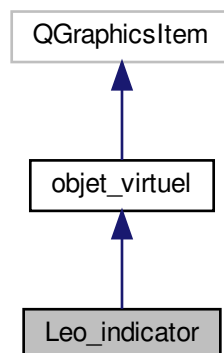
- serveur/Leo/leo_gauge.h
- serveur/Leo/leo_gauge.cpp

6.65 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**

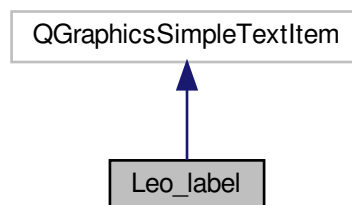
Additional Inherited Members

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

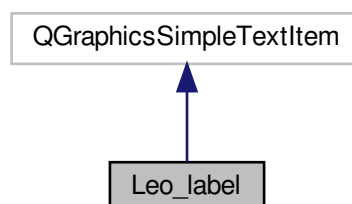
- serveur/Leo/leo_indicator.h
- serveur/Leo/leo_indicator.cpp

6.66 Leo_label Class Reference

Inheritance diagram for Leo_label:



Collaboration diagram for Leo_label:



Public Member Functions

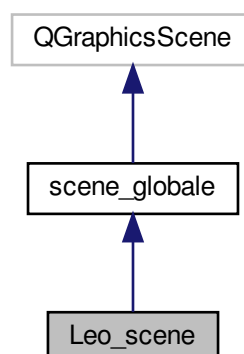
- **Leo_label** (const QString &text, const QRectF boundingRect, QGraphicsItem *parent=nullptr)
- QRectF **boundingRect** () const override
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *, QWidget *) override

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

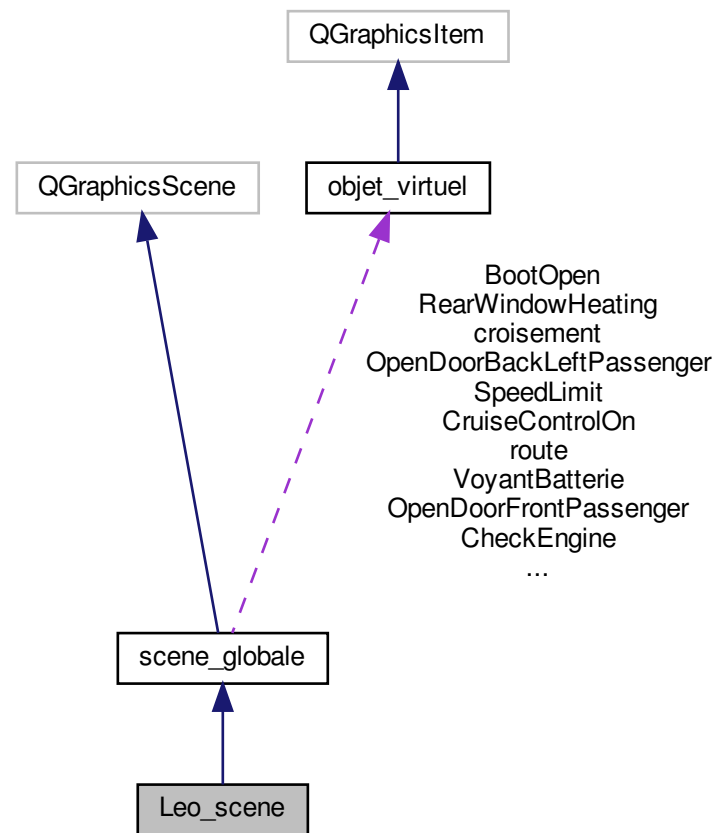
- serveur/Leo/leo_label.h
- serveur/Leo/leo_label.cpp

6.67 Leo_scene Class Reference

Inheritance diagram for Leo_scene:



Collaboration diagram for Leo_scene:



Public Member Functions

- **Leo_scene** ([scene_globale](#) *parent=nullptr)

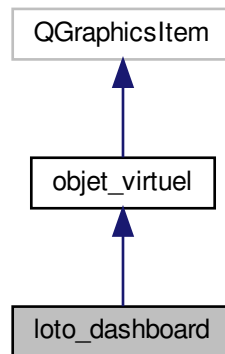
Additional Inherited Members

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

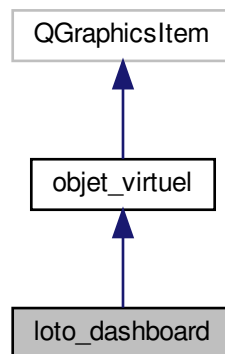
- serveur/Leo/leo_scene.h
- serveur/Leo/leo_scene.cpp

6.68 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

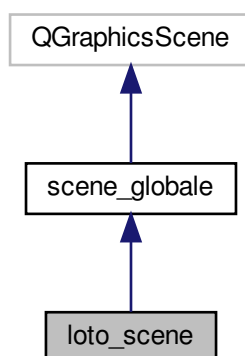
Additional Inherited Members

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

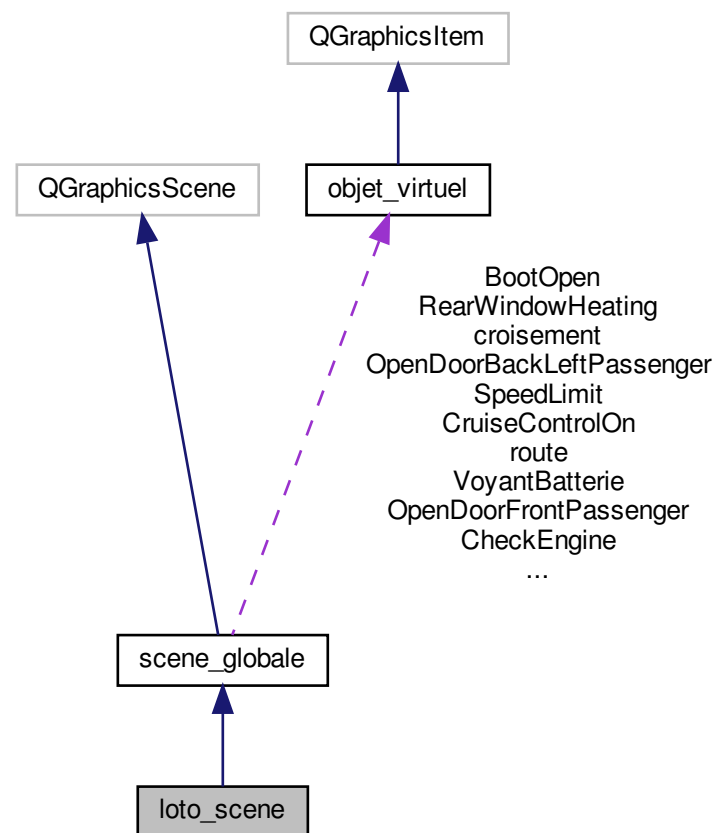
- serveur/Loto/loto_dashboard.h
- serveur/Loto/loto_dashboard.cpp

6.69 loto_scene Class Reference

Inheritance diagram for loto_scene:



Collaboration diagram for loto_scene:



Public Member Functions

- **loto_scene** (**scene_globale** *parent=nullptr)

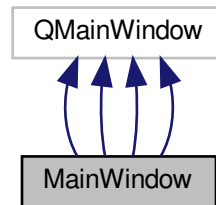
Additional Inherited Members

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

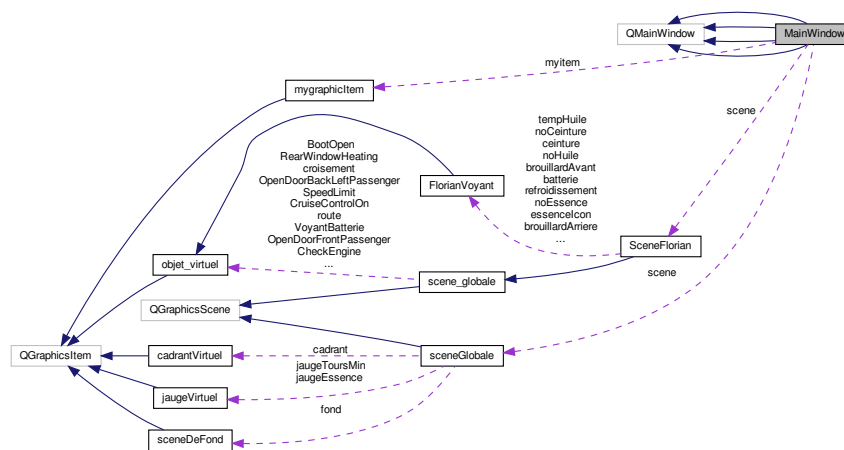
- serveur/Loto/loto_scene.h
- serveur/Loto/loto_scene.cpp

6.70 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](#) ()

6.70.1 Constructor & Destructor Documentation

6.70.1.1 MainWindow()

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

La scène par défaut est

6.70.2 Member Function Documentation

6.70.2.1 acceleration()

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

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

Fonction simulant une accélération linéaire avec changements de rapports de vitesse

Parameters

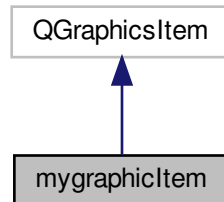
<i>time</i>	Durée de la simulation
-------------	------------------------

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

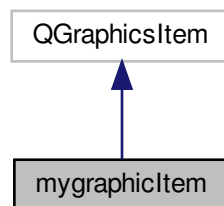
- serveur/Florian/mainwindow.h
- serveur/Florian/mainwindow.cpp

6.71 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:

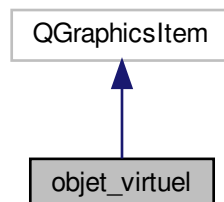
- serveur/Lea/mygraphicitem.h
- serveur/Lea/mygraphicitem.cpp

6.72 objet_virtuel Class Reference

Inherits QGraphicsItem.

Inherited by [afficheKmHenri](#), [ArrowOilInna](#), [ArrowOilTInna](#), [ArrowSpeedometerInna](#), [ArrowTachometerInna](#), [cadran](#), [CadranEss](#), [CadrantFlorian](#), [cadrantHenri](#), [CadranTourParMin](#), [CadranVitesse](#), [CarSpeedDial](#), [clignot](#), [Compteur_kilometre_lea](#), [compteurKmlInna](#), [EngineTInna](#), [FlorianClignotant](#), [FlorianCompteurKm](#), [FlorianVoyant](#), [fuel_guage](#), [FuelGauge_lea](#), [FuelLevellInna](#), [Harc](#), [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_gauge](#), [Leo_indicator](#), [loto_dashboard](#), [onoff](#), [onoffpaintHenri](#), [portesHenri](#), [speedometer_Lea](#), [TachometerGauge_Lea](#), [tachometre](#), [tempGauge](#), [voyant_Lea](#), [Voyants](#), [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**

6.72.1 Constructor & Destructor Documentation

6.72.1.1 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

6.72.2 Member Function Documentation

6.72.2.1 getValue()

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

Fonction renvoyant la variable value.

Returns

6.72.2.2 getValueMax()

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

[objet_virtuel::getValueMax](#) Fonction renvoyant la valeur de valueMax

Returns

6.72.2.3 setValue()

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

[objet_virtuel::setValue](#). Permet de modifier la valeur de la variable value

Parameters

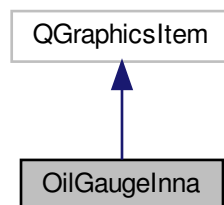
<code>set_value</code>	valeur à attribuer à value
------------------------	----------------------------

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

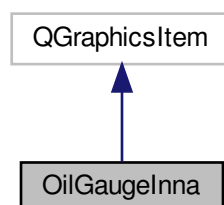
- serveur/objet_virtuel.h
- serveur/objet_virtuel.cpp

6.73 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**

6.73.1 Member Function Documentation

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

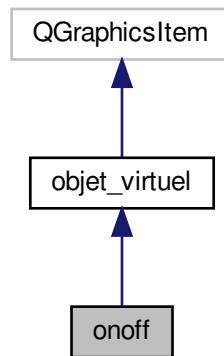
Affichage des traits et du texte pour le niveau d'huile.

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

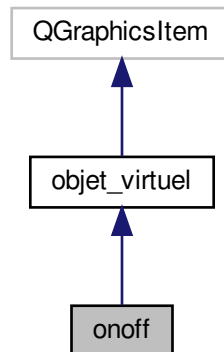
- serveur/Inna/oilgaugeinna.h
- serveur/Inna/oilgaugeinna.cpp

6.74 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)

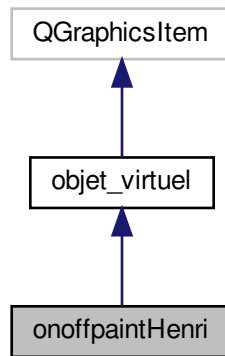
Additional Inherited Members

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

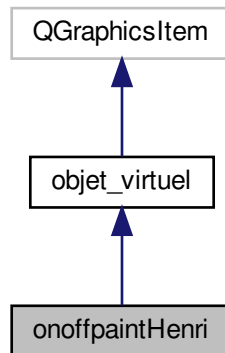
- serveur/Henri/onoff.h
- serveur/Henri/onoff.cpp

6.75 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**

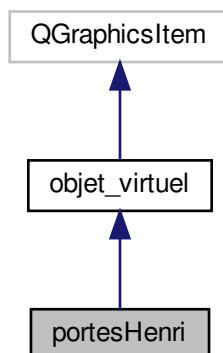
Additional Inherited Members

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

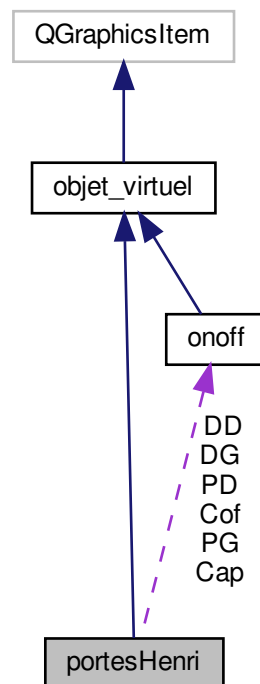
- serveur/Henri/onoffpainthenri.h
- serveur/Henri/onoffpainthenri.cpp

6.76 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**

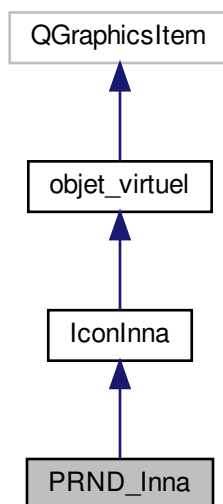
Additional Inherited Members

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

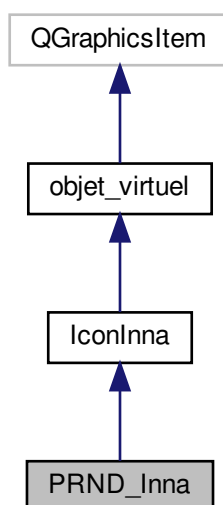
- serveur/Henri/porteshenri.h
- serveur/Henri/porteshenri.cpp

6.77 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

6.77.1 Member Function Documentation

6.77.1.1 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;

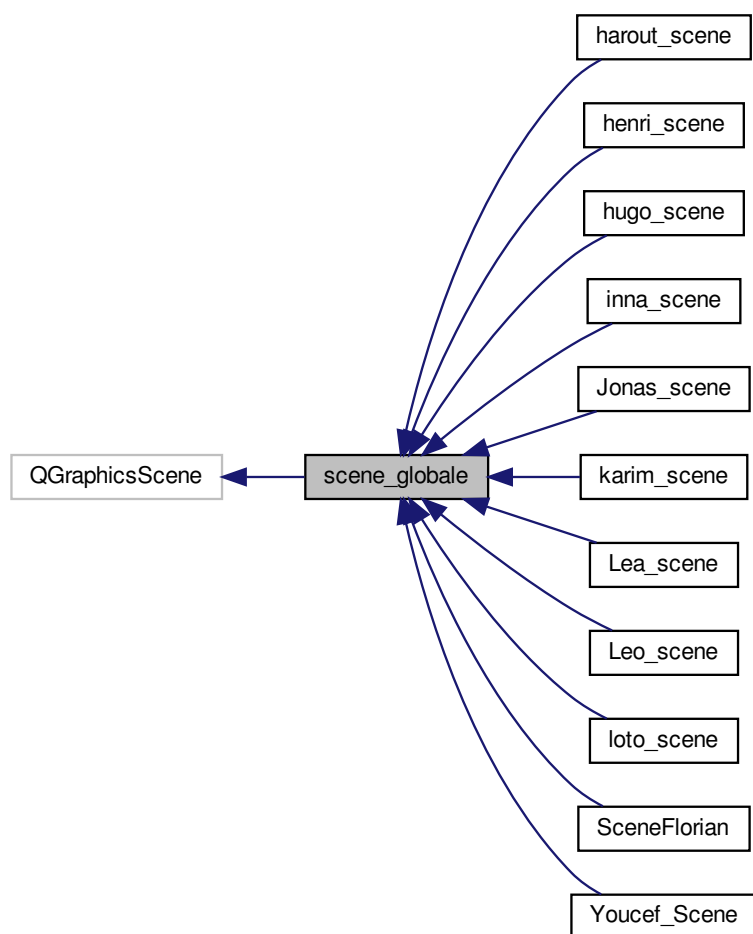
Dans autres cas le voyant "_probleme avec la boite de vitesse_" et affiché.

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

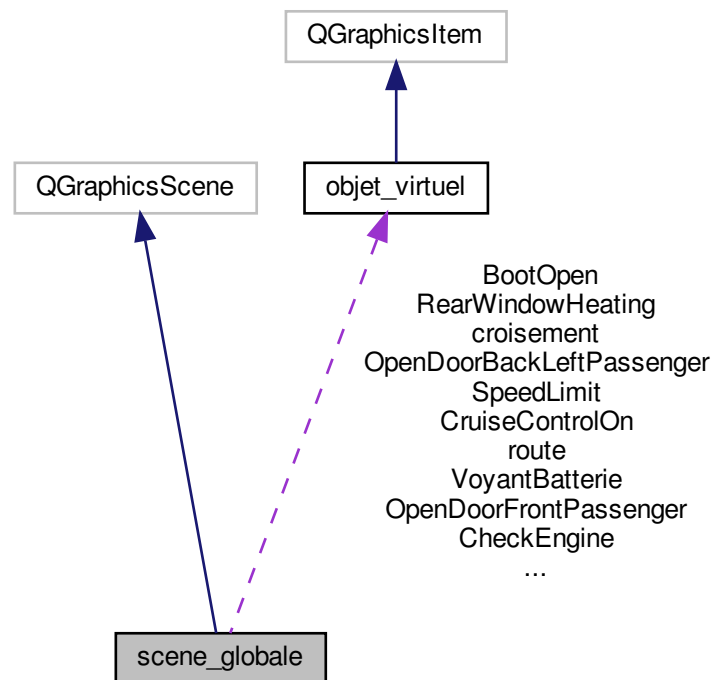
- serveur/Inna/prnd_inna.h
- serveur/Inna/[prnd_inna.cpp](#)

6.78 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**
- `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**

6.78.1 Constructor & Destructor Documentation

6.78.1.1 scene_globale()

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

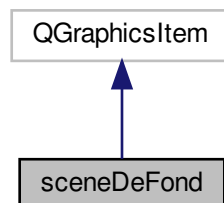
Classe dérivant de QGraphicsScene. Va servir de classe mère pour toutes les scènes, ce qui permettra de passer d'une scène à l'autre de façon dynamique. Tous les objets utilisés dans le dashboard seront définis dans "scène↵_globale.h"

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

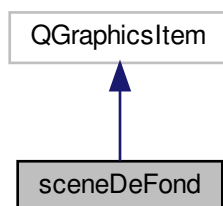
- serveur/scene_globale.h
- serveur/scene_globale.cpp

6.79 sceneDeFond Class Reference

Inheritance diagram for sceneDeFond:



Collaboration diagram for sceneDeFond:



Public Member Functions

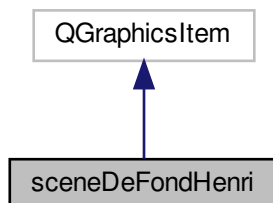
- **sceneDeFond** (`QGraphicsItem *parent=nullptr`)

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

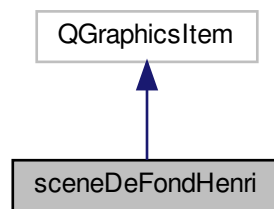
- `serveur/Florian/scenedefond.h`
- `serveur/Florian/scenedefond.cpp`

6.80 sceneDeFondHenri Class Reference

Inheritance diagram for sceneDeFondHenri:



Collaboration diagram for sceneDeFondHenri:



Public Member Functions

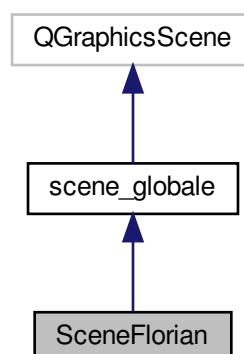
- **sceneDeFondHenri** (`QGraphicsItem *parent=nullptr`)
- void **paint** (`QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget`)
- `QRectF` **boundingRect** () const

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

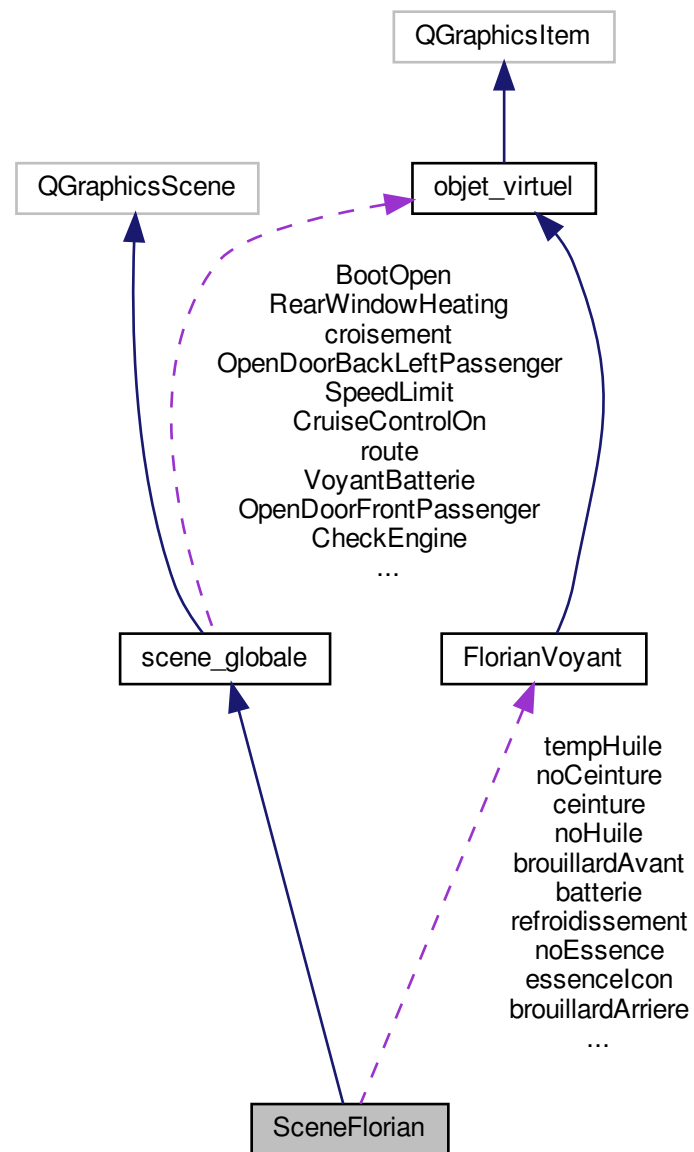
- `serveur/Henri/scenedefondhenri.h`
- `serveur/Henri/scenedefondhenri.cpp`

6.81 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**

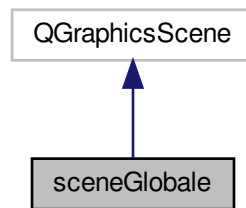
- [FlorianVoyant](#) * **brouillardArriere**
- [FlorianVoyant](#) * **noEssence**
- [FlorianVoyant](#) * **noHuile**
- [FlorianVoyant](#) * **refroidissement**
- [FlorianVoyant](#) * **batterie**
- [FlorianVoyant](#) * **ceinture**
- [FlorianVoyant](#) * **noCeinture**

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

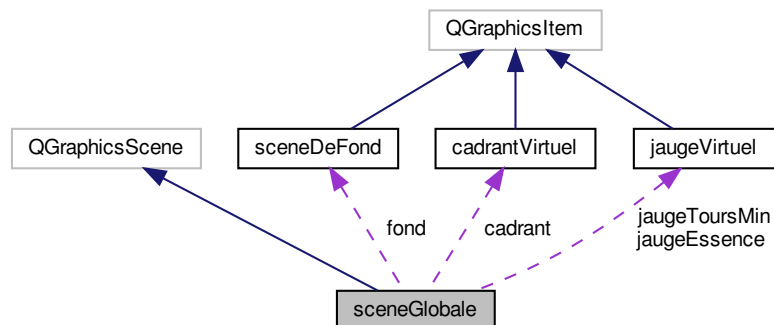
- serveur/Florian/sceneflorian.h
- serveur/Florian/sceneflorian.cpp

6.82 sceneGlobale Class Reference

Inheritance diagram for sceneGlobale:



Collaboration diagram for sceneGlobale:



Public Member Functions

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

Public Attributes

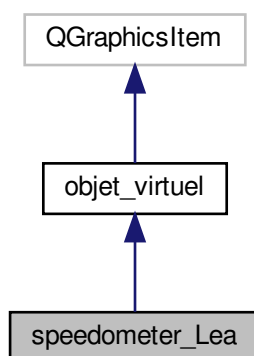
- [sceneDeFond](#) * **fond**
- [cadrantVirtuel](#) * **cadrant**
- [jaugeVirtuel](#) * **jaugeEssence**
- [jaugeVirtuel](#) * **jaugeToursMin**

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

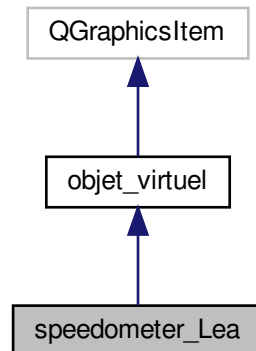
- serveur/Florian/sceneglobale.h
- serveur/Florian/sceneglobale.cpp

6.83 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
- 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

6.83.1 Constructor & Destructor Documentation

6.83.1.1 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

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 du cadran
<i>param_param_vitMax</i>	determine la vitesse maximum jusqu'à laquelle va le cadran

6.83.2 Member Function Documentation

6.83.2.1 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>	
----------------	--

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.

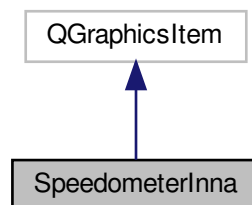
L'affichage de la vitesse se fait avec la fonction `drawText` et en utilisant la valeur donnée par la fonction `getValue()` de la classe `objet_virtuel`.

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

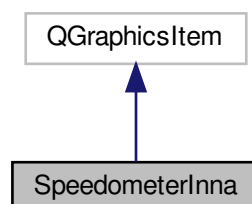
- serveur/Lea/speedometer_lea.h
- serveur/Lea/speedometer_lea.cpp

6.84 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**

6.84.1 Member Function Documentation

6.84.1.1 paint()

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

Dessin du circle encadrant

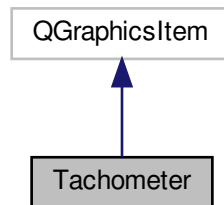
Dessin du texte et des traits

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

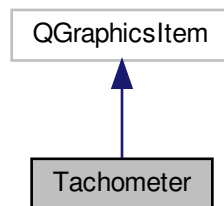
- serveur/lnna/speedometerinna.h
- serveur/lnna/speedometerinna.cpp

6.85 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

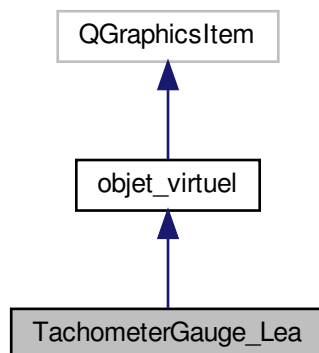
- float **pi** = 3.14159265359
- int **speed**

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

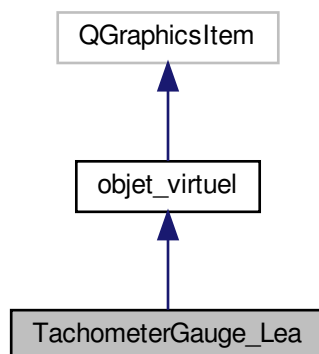
- serveur/Youcef/tachometer.h
- serveur/Youcef/tachometer.cpp

6.86 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.
- 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

6.86.1 Constructor & Destructor Documentation

6.86.1.1 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.

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

6.86.2 Member Function Documentation

6.86.2.1 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 drawPolygon. Puis affichage d'un petit cercle à la base de l'aiguille.

Création de l'affichage RPM.

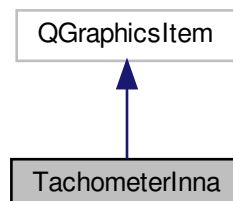
L'affichage RPM se fait avec la fonction drawText.

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

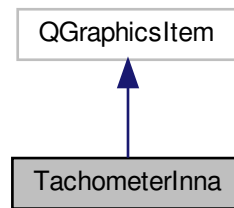
- serveur/Lea/tachometergauge_lea.h
- serveur/Lea/[tachometergauge_lea.cpp](#)

6.87 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**

6.87.1 Member Function Documentation

6.87.1.1 paint()

```

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

Dessin du circle encadrant

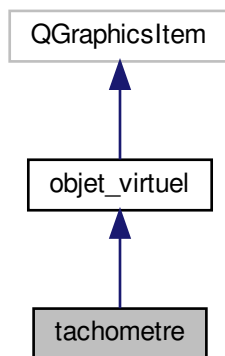
Dessin du texte et des traits

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

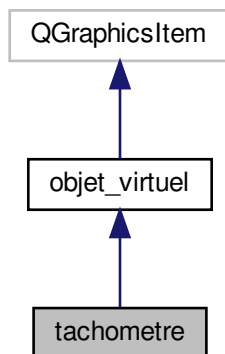
- serveur/Inna/tachometerinna.h
- serveur/Inna/[tachometerinna.cpp](#)

6.88 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

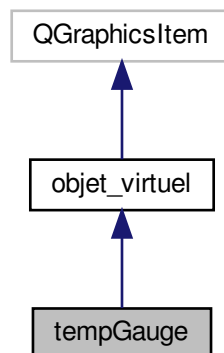
Additional Inherited Members

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

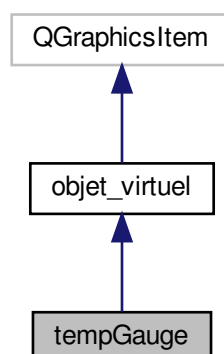
- serveur/Loto/tachometre.h
- serveur/Loto/tachometre.cpp

6.89 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)
- double **getEngineTemp** (double engineTemp)

Public Attributes

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

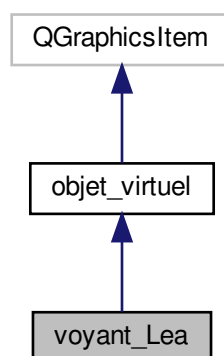
Additional Inherited Members

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

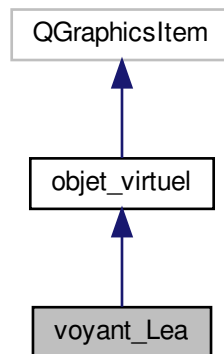
- serveur/Loto/tempgauge.h
- serveur/Loto/tempgauge.cpp

6.90 voyant_Lea Class Reference

Inheritance diagram for voyant_Lea:



Collaboration diagram for voyant_Lea:



Public Member Functions

- [voyant_Lea](#) (int, int, QString, int, int)
[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

Additional Inherited Members

6.90.1 Constructor & Destructor Documentation

6.90.1.1 voyant_Lea()

```

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

[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

6.90.2 Member Function Documentation

6.90.2.1 paint()

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

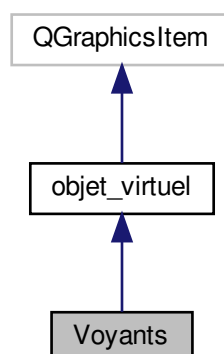
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché. Si ==0, le voyant doit être éteint

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

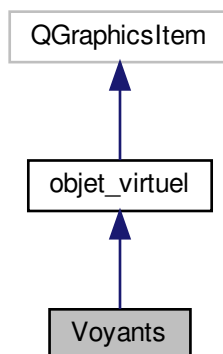
- serveur/Lea/voyant_lea.h
- serveur/Lea/[voyant_lea.cpp](#)

6.91 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)

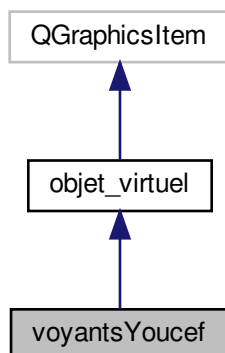
Additional Inherited Members

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

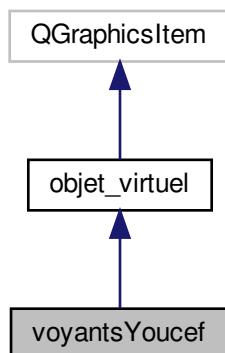
- serveur/Karim/voyants.h
- serveur/Karim/voyants.cpp

6.92 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**

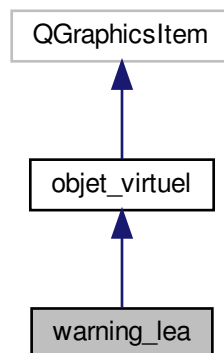
Additional Inherited Members

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

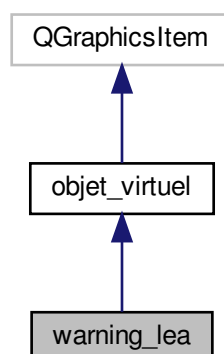
- serveur/Youcef/voyantsyoucef.h
- serveur/Youcef/voyantsyoucef.cpp

6.93 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**

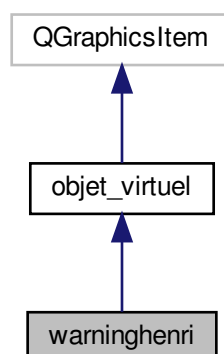
Additional Inherited Members

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

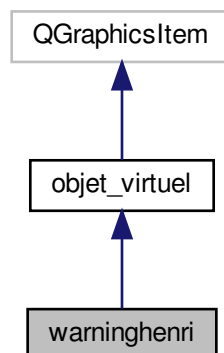
- serveurur/Lea/warning_lea.h
- serveurur/Lea/[warning_lea.cpp](#)

6.94 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)

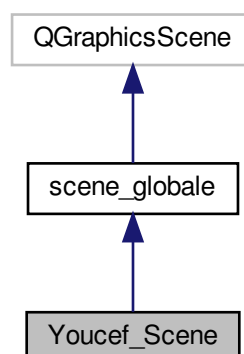
Additional Inherited Members

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

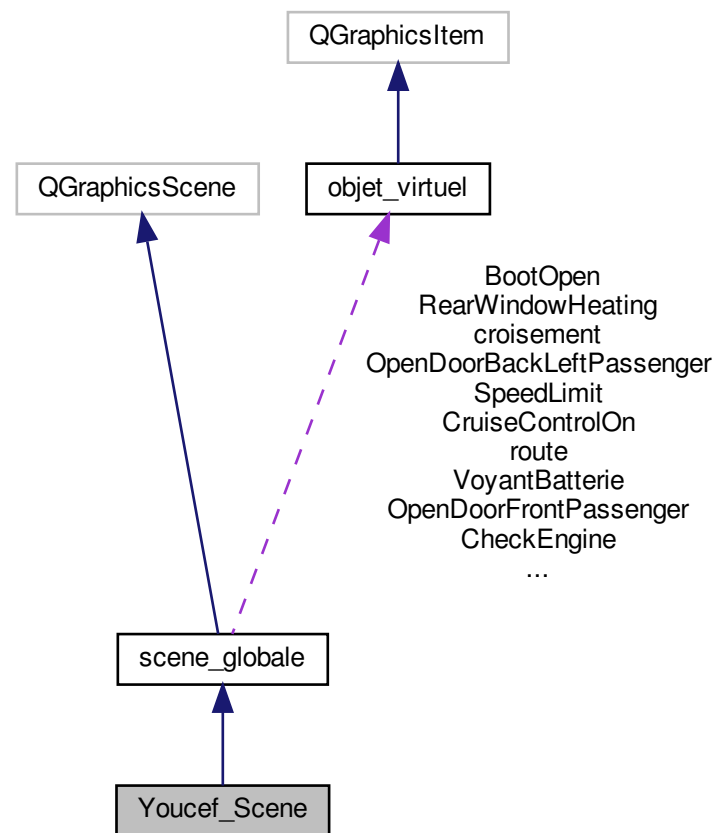
- serveur/Henri/warninghenri.h
- serveur/Henri/warninghenri.cpp

6.95 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

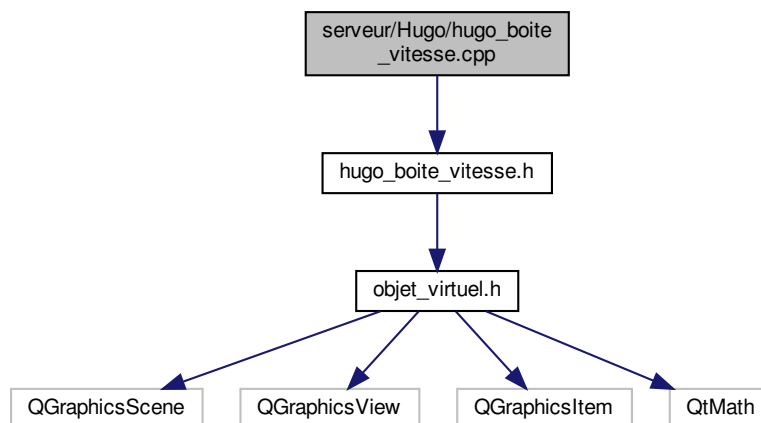
File Documentation

7.1 serveur/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:



7.1.1 Detailed Description

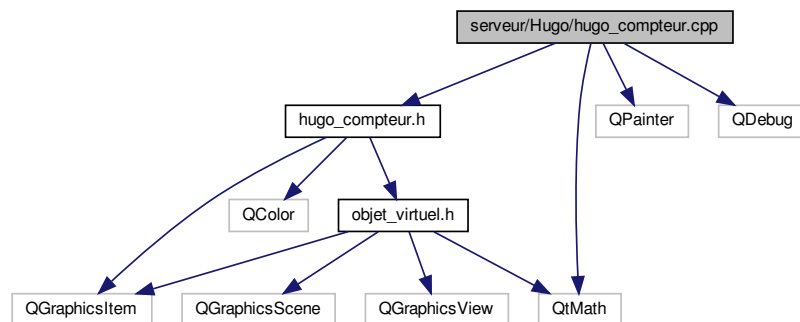
Classe dérivant de [objet_virtuel](#) permettant l'affichage du mode de transmission automatique du véhicule.

7.2 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`

7.2.1 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.

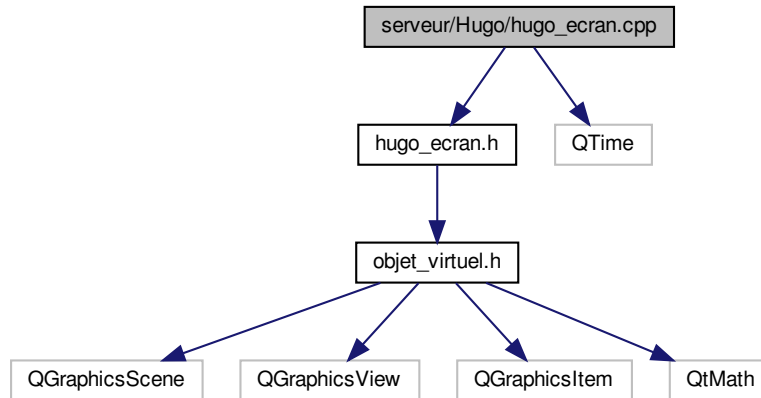
Bug L'utilisation du paramètre critique semble être soumis à certains bugs.

7.3 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:



7.3.1 Detailed Description

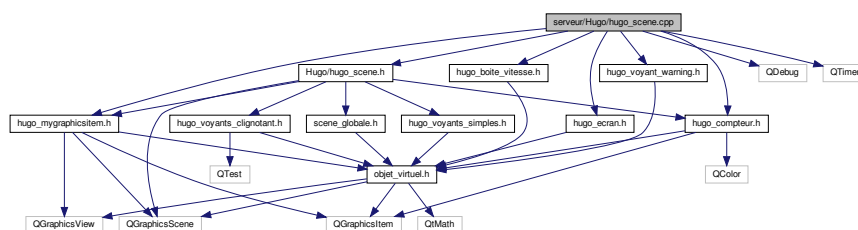
Classe dérivant de [objet_virtuel](#) permettant un affichage en temps réel du temps ainsi que de la distance parcourue depuis le redémarrage du programme.

7.4 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:



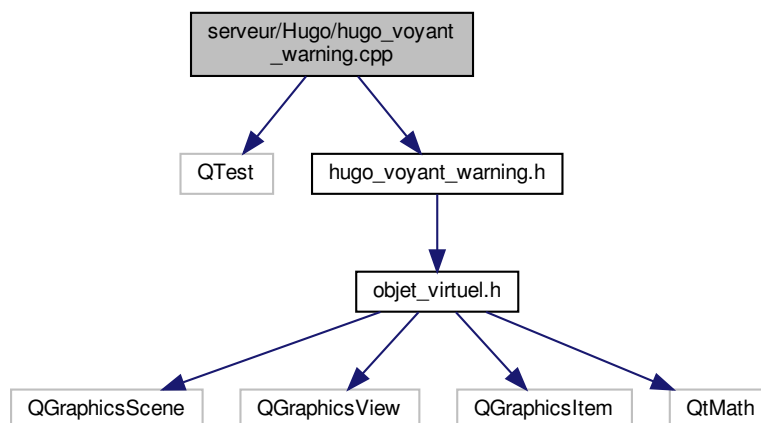
7.4.1 Detailed Description

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](#).

7.5 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:
```



7.5.1 Detailed Description

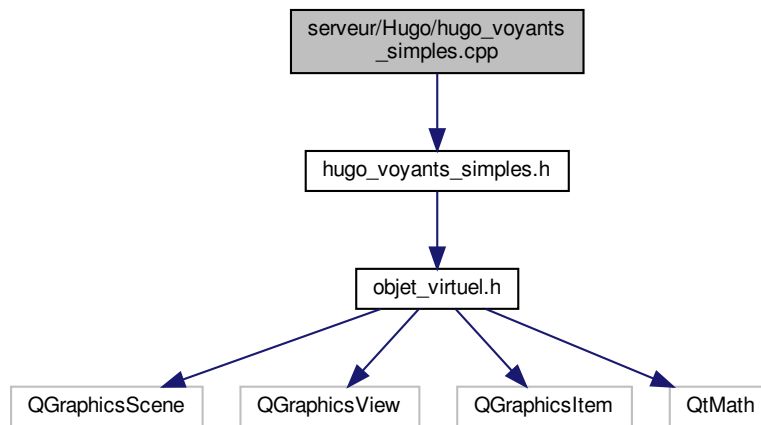
Classe dérivée de [objet_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

7.6 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:



7.6.1 Detailed Description

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

7.7 serveur/Inna/arrowoillinna.cpp File Reference

Classe [ArrowOilInna](#) dérivée de la classe [objet_virtuel](#).

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

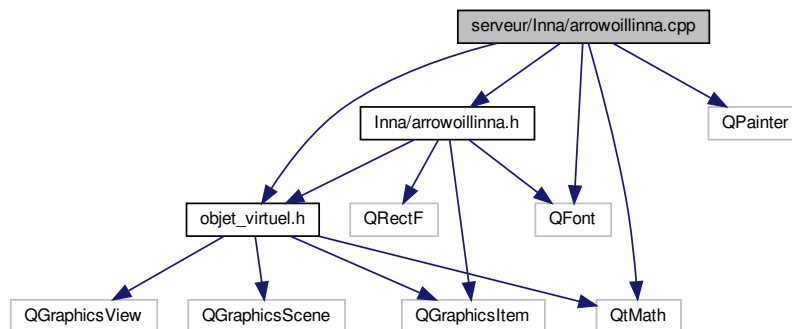
```
#include "objet_virtuel.h"
```

```
#include <QFont>
```

```
#include <QtMath>
```

```
#include <QPainter>
```

Include dependency graph for arrowoillinna.cpp:



7.7.1 Detailed Description

Classe [ArrowOilInna](#) dérivée de la classe [objet_virtuel](#).

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

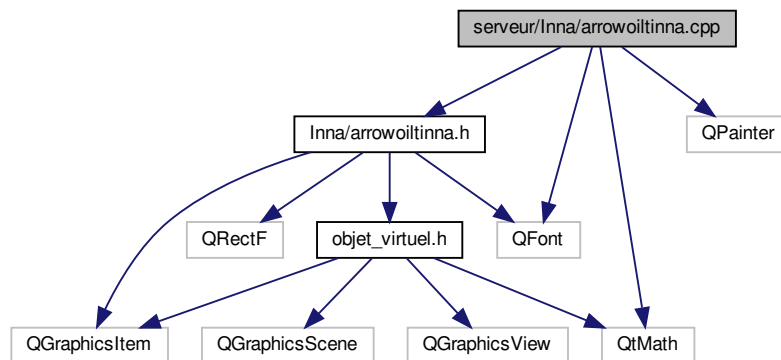
Parameters

<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

7.8 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:
```



7.8.1 Detailed Description

Classe [ArrowOilTinna](#) dérivée de la classe [objet_virtuel](#).

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

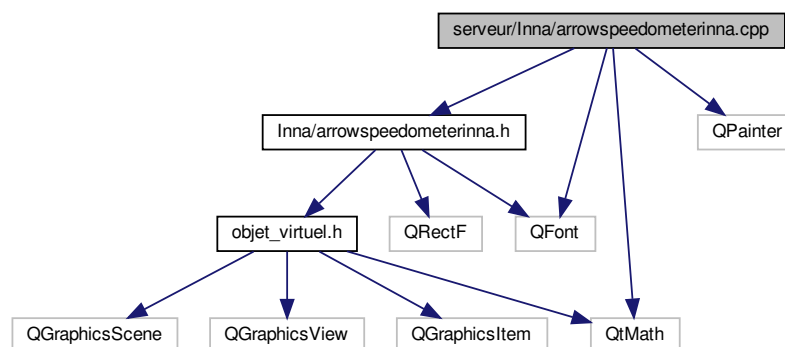
Parameters

<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

7.9 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:
```



7.9.1 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".

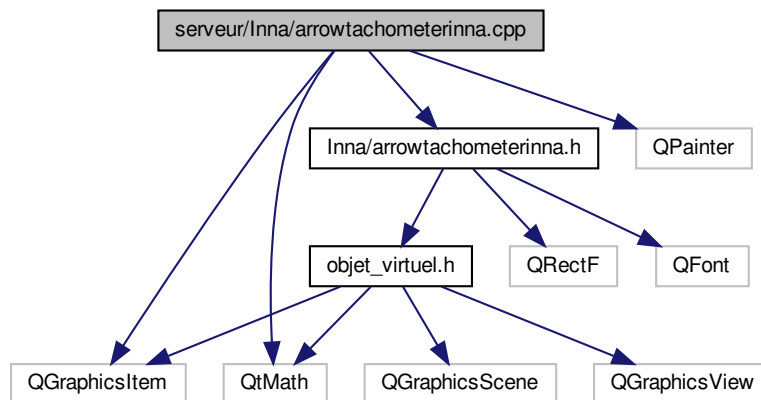
Parameters

<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

7.10 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:
```



7.10.1 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

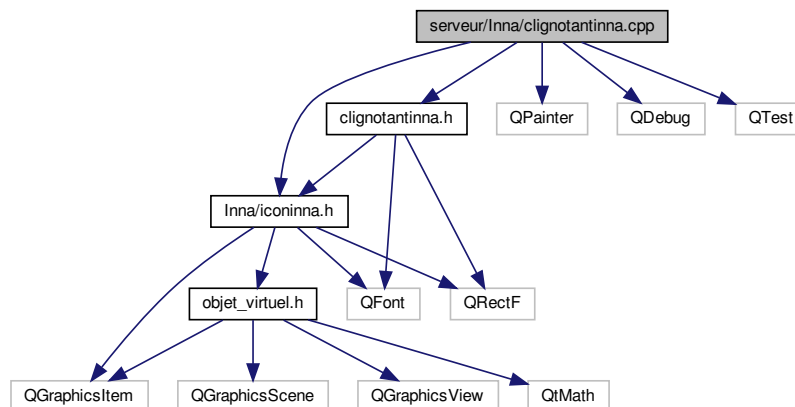
7.11 serveur/Inna/clignotantinna.cpp File Reference

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:



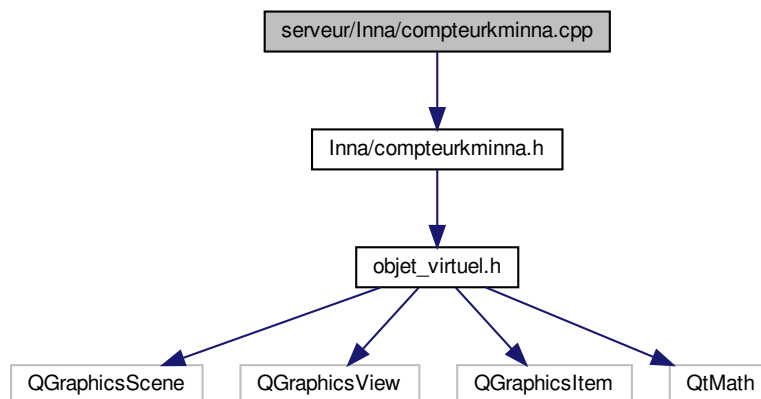
7.11.1 Detailed Description

Classe permettant d'afficher les voyants clignotants.

7.12 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:
```



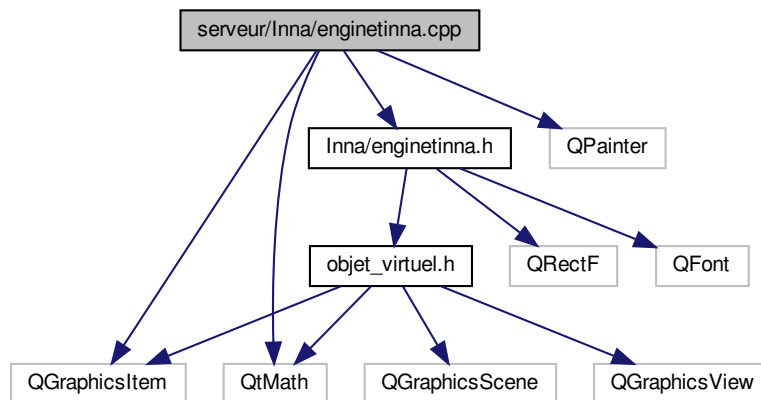
7.12.1 Detailed Description

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

7.13 serveur/Inna/engineinna.cpp File Reference

Classe [EngineTInna](#) dérivée de la classe [objet_virtuel](#).

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



7.13.1 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".

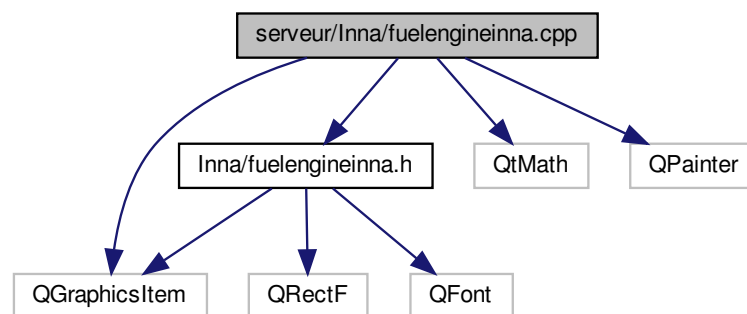
Parameters

<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

7.14 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:
```



7.14.1 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

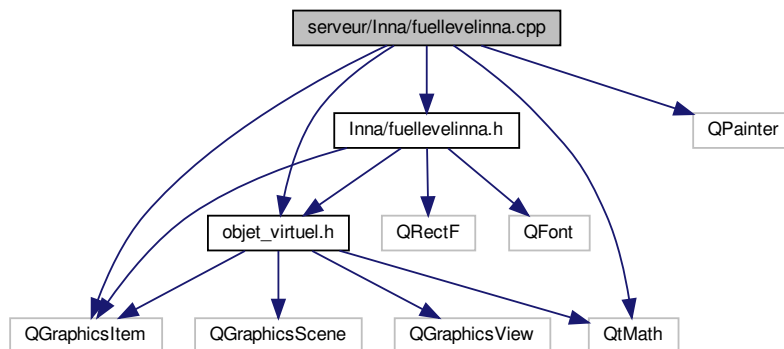
7.15 serveur/Inna/fuellevelinna.cpp File Reference

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`:



7.15.1 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

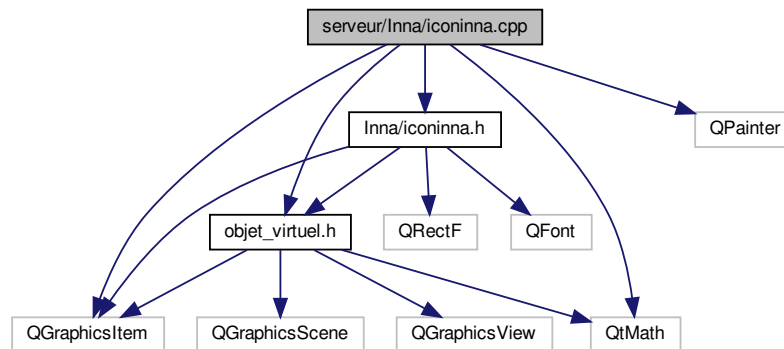
<i>value</i>	- la valeur du niveau d'essence
<i>valueMax</i>	- le niveau d'essence maximum
<i>A0</i>	- l'aunge de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

7.16 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:



7.16.1 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.

7.17 serveur/Inna/icononoffinna.cpp File Reference

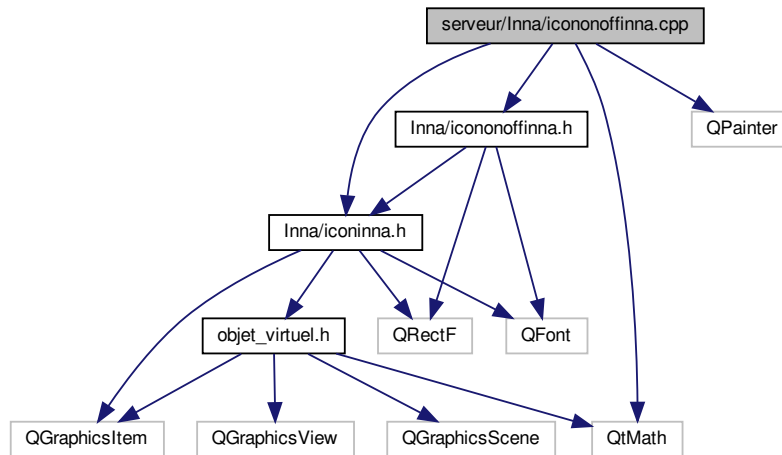
Classe `iconOnOffInna` dérivée de la classe `IconInna`.

```

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

```

Include dependency graph for `icononoffinna.cpp`:



7.17.1 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.

7.18 serveur/Inna/infowindowinna.cpp File Reference

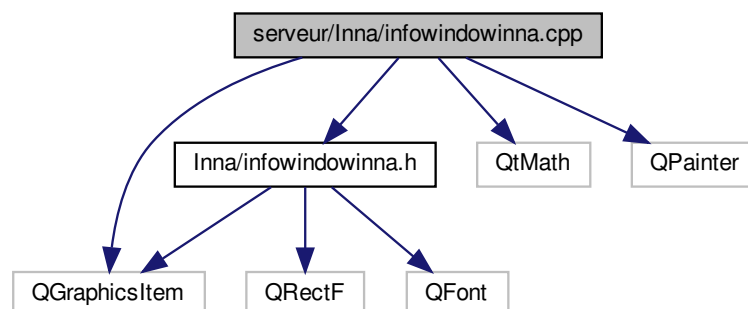
Classe `InfoWindowInna` dérivée de la classe `QGraphicsItem`.

```

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

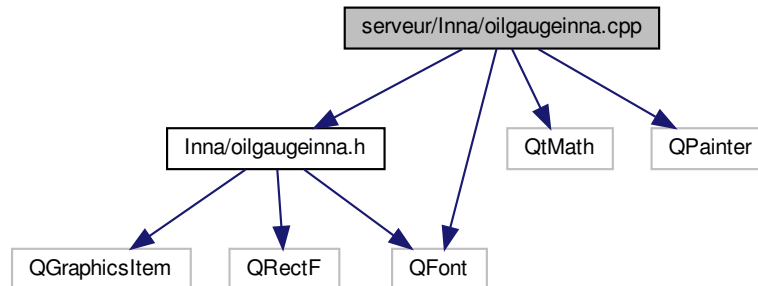
```

Include dependency graph for `infowindowinna.cpp`:




```
#include <QPainter>
```

Include dependency graph for oilgaugeinna.cpp:



7.20.1 Detailed Description

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

Parameters

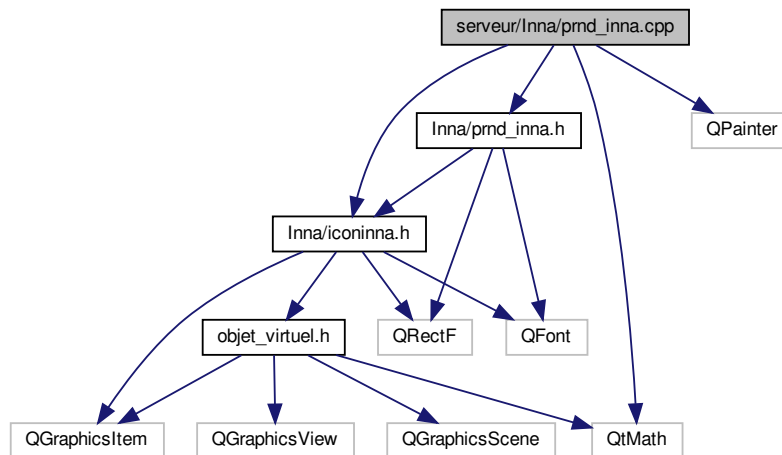
<i>tmax</i>	- la température maximum;
<i>lmax</i>	- le niveau maximum
<i>r</i>	- rayon de l'arche de la jauge;
<i>A0</i>	- l'angle du début de la jauge;
<i>Amax</i>	- l'angle total
<i>k</i>	- coefficient pour convertir la valeur en degrés

7.21 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:



7.21.1 Detailed Description

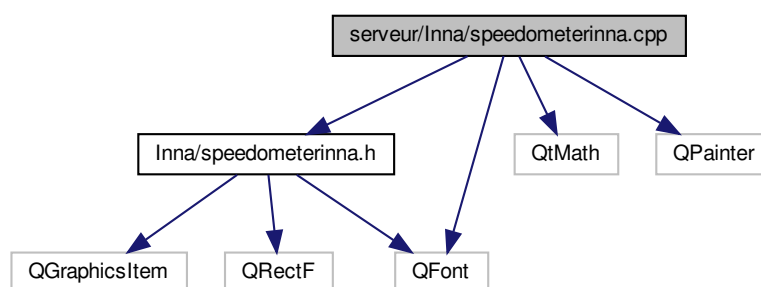
Classe [PRND_Inna](#) dérivée de la classe [IconInna](#) pour afficher les modes de la transmission.

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

7.22 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:
```



7.22.1 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

7.23 serveur/Inna/tachometerinna.cpp File Reference

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

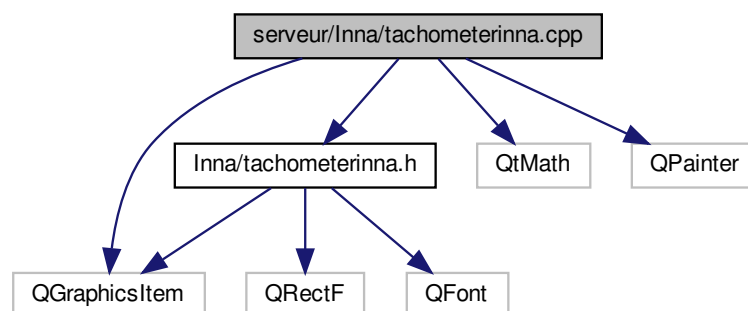
```
#include "Inna/tachometerinna.h"
```

```
#include <QGraphicsItem>
```

```
#include <QtMath>
```

```
#include <QPainter>
```

Include dependency graph for tachometerinna.cpp:



7.23.1 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

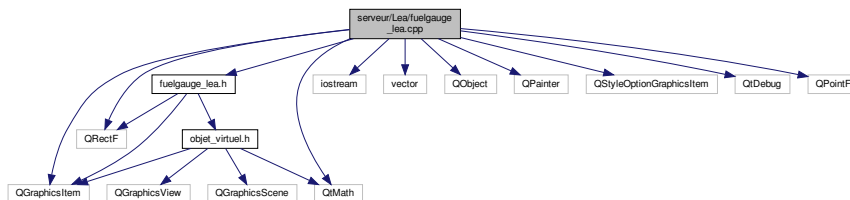
<i>valueMax</i>	- la valeur de tours par minutemaximum
<i>A0</i>	- l'angle de debut de la jauge
<i>Amax</i>	- l'ampleur de la jauge
<i>r</i>	- le rayon de l'arche de la jauge
<i>k</i>	- le coefficient pour convertir la valeur en degrés

7.24 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 <QGraphicsItem>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QRectF>
#include <QtMath>
#include <QtDebug>
#include <QPointF>
```

Include dependency graph for fuelgauge_lea.cpp:



7.24.1 Detailed Description

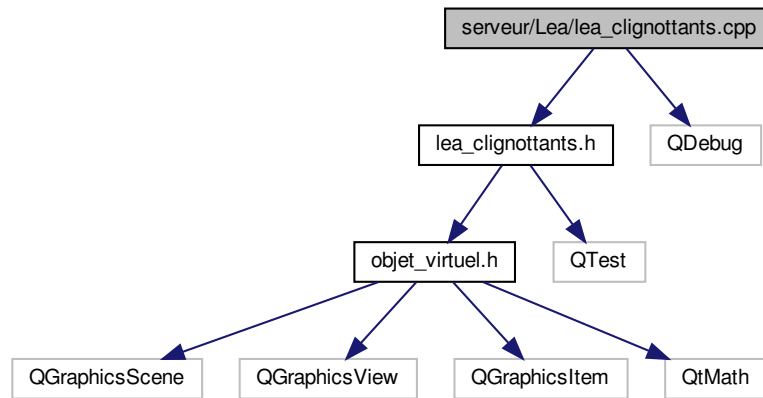
Classe jauge d'essence, permettant la création d'une jauge paramétrables .

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.

7.25 serveur/Lea/lea_clignottants.cpp File Reference

Classe dérivée de [objet_virtuel](#) permettant l'affichage d'un voyant clignotant droite ou gauche.

```
#include "lea_clignottants.h"
#include <QDebug>
Include dependency graph for lea_clignottants.cpp:
```



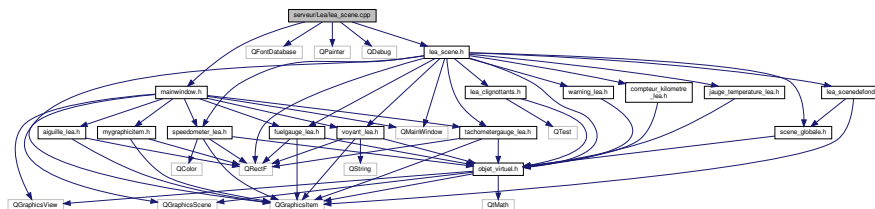
7.25.1 Detailed Description

Classe dérivée de [objet_virtuel](#) permettant l'affichage d'un voyant clignotant droite ou gauche.

7.26 serveur/Lea/lea_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 "lea_scene.h"
#include <QFontDatabase>
#include <QPainter>
#include <QDebug>
#include "mainwindow.h"
Include dependency graph for lea_scene.cpp:
```



7.26.1 Detailed Description

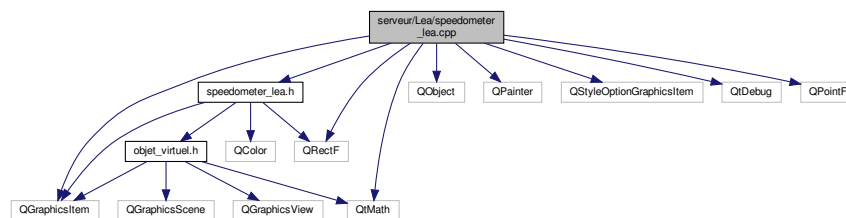
Classe dérivée de [scene_globale](#) où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans [scene_globale.h](#).

7.27 serveur/Lea/speedometer_lea.cpp File Reference

Classe speedometer, permettant la création du compteur vitesse paramétrable avec aiguille.

```
#include "speedometer_lea.h"
#include <QGraphicsItem>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QRectF>
#include <QtMath>
#include <QtDebug>
#include <QPointF>
```

Include dependency graph for speedometer_lea.cpp:



7.27.1 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.

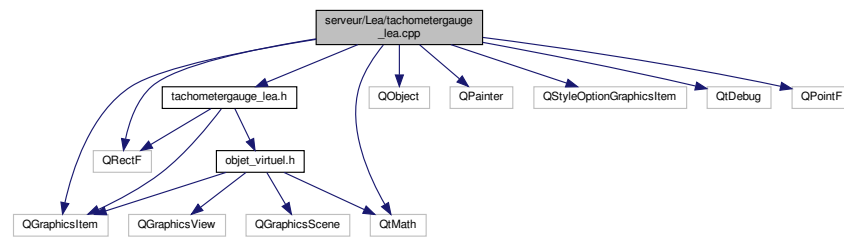
7.28 serveur/Lea/tachometergauge_lea.cpp File Reference

Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille.

```
#include "tachometergauge_lea.h"
#include <QGraphicsItem>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QRectF>
#include <QtMath>
#include <QtDebug>
```

```
#include <QPointF>
```

Include dependency graph for tachometergauge_lea.cpp:



7.28.1 Detailed Description

Classe jauge de tours/minute, permettant la création d'un compteur paramétrable avec aiguille.

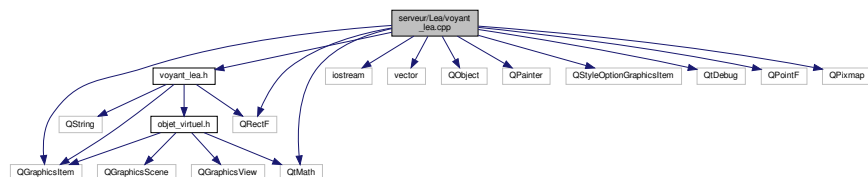
Classe héritée de [objet_virtuel](#). Utilisée pour l'affichage du compteur rpm. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction paint permet l'affichage.

7.29 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 <QGraphicsItem>
#include <QObject>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <QRectF>
#include <QtMath>
#include <QtDebug>
#include <QPointF>
#include <QPixmap>
```

Include dependency graph for voyant_lea.cpp:



7.29.1 Detailed Description

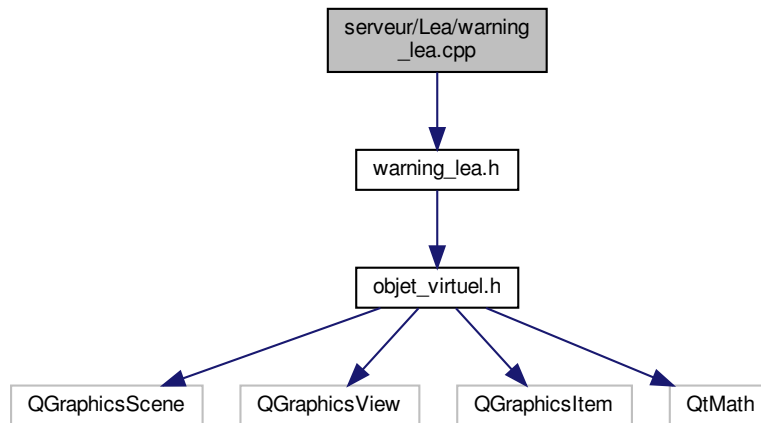
Classe permettant l'affichage de voyants dont l'état est allumé ou éteint.

7.30 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:



7.30.1 Detailed Description

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

