

Doc car_dashboard

Generated by Doxygen 1.8.13

Contents

1	Projet dashboard	1
2	Bug List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Class Documentation	11
6.1	afficheKmHenri Class Reference	11
6.2	aiguille_Lea Class Reference	12
6.3	CadranEss Class Reference	14
6.4	CadrantFlorian Class Reference	15
6.5	cadrantHenri Class Reference	16
6.6	CadranTourParMin Class Reference	17
6.7	cadrantVirtuel Class Reference	18
6.8	CadranVitesse Class Reference	19
6.9	clignot Class Reference	20
6.10	deux_voyants_lea Class Reference	22
6.11	FlorianClignotant Class Reference	23
6.12	FlorianCompteurKm Class Reference	25

6.13 FlorianVoyant Class Reference	26
6.14 FuelGauge_lea Class Reference	27
6.15 henri_scene Class Reference	28
6.16 hugo_boite_vitesse Class Reference	30
6.16.1 Member Function Documentation	30
6.16.1.1 paint()	31
6.17 hugo_Compteur Class Reference	31
6.17.1 Constructor & Destructor Documentation	32
6.17.1.1 hugo_Compteur()	33
6.17.2 Member Function Documentation	33
6.17.2.1 paint()	33
6.18 hugo_ecran Class Reference	35
6.18.1 Constructor & Destructor Documentation	36
6.18.1.1 hugo_ecran()	36
6.18.2 Member Function Documentation	36
6.18.2.1 paint()	36
6.19 hugo_MyGraphicsitem Class Reference	37
6.19.1 Constructor & Destructor Documentation	38
6.19.1.1 hugo_MyGraphicsitem()	38
6.19.2 Member Function Documentation	38
6.19.2.1 paint()	38
6.20 hugo_scene Class Reference	39
6.21 hugo_voyant_warning Class Reference	41
6.21.1 Constructor & Destructor Documentation	42
6.21.1.1 hugo_voyant_warning()	42
6.21.2 Member Function Documentation	42
6.21.2.1 paint()	42
6.22 hugo_voyants_clignotant Class Reference	43
6.22.1 Constructor & Destructor Documentation	44
6.22.1.1 hugo_voyants_clignotant()	44

6.22.2	Member Function Documentation	44
6.22.2.1	paint()	44
6.23	hugo_voyants_simples Class Reference	45
6.23.1	Constructor & Destructor Documentation	46
6.23.1.1	hugo_voyants_simples()	46
6.23.2	Member Function Documentation	46
6.23.2.1	paint()	46
6.24	jaugeBatterieHenri Class Reference	47
6.25	jaugeClignotantHenri Class Reference	48
6.26	jaugeEssenceHenri Class Reference	49
6.27	jaugeTemperatureHenri Class Reference	51
6.28	jaugeToursMinuteHenri Class Reference	52
6.29	jaugeVirtuel Class Reference	53
6.30	Jonas_compteur Class Reference	54
6.31	Jonas_scene Class Reference	55
6.32	jonas_voyant_simple Class Reference	57
6.33	karim_scene Class Reference	58
6.34	lea_clignottants Class Reference	60
6.35	Lea_scene Class Reference	61
6.36	lea_sceneDeFond Class Reference	63
6.37	Leo_gauge Class Reference	64
6.38	Leo_indicator Class Reference	65
6.39	Leo_label Class Reference	66
6.40	Leo_scene Class Reference	67
6.41	MainWindow Class Reference	69
6.41.1	Constructor & Destructor Documentation	70
6.41.1.1	MainWindow()	70
6.41.2	Member Function Documentation	70
6.41.2.1	acceleration()	70
6.42	mygraphicItem Class Reference	71

6.43	objet_virtuel Class Reference	72
6.43.1	Constructor & Destructor Documentation	73
6.43.1.1	objet_virtuel()	73
6.43.2	Member Function Documentation	74
6.43.2.1	getValue()	74
6.43.2.2	getValueMax()	74
6.43.2.3	setValue()	74
6.44	phares_Lea Class Reference	75
6.45	Quatre_Voyants_Lea Class Reference	76
6.46	scene_globale Class Reference	77
6.46.1	Constructor & Destructor Documentation	79
6.46.1.1	scene_globale()	79
6.47	sceneDeFond Class Reference	79
6.48	sceneDeFondHenri Class Reference	80
6.49	SceneFlorian Class Reference	81
6.50	sceneGlobale Class Reference	83
6.51	speedometer_Lea Class Reference	84
6.52	TachometerGauge_Lea Class Reference	85
6.53	Voyant_batterie Class Reference	87
6.54	voyant_Lea Class Reference	88
6.55	Voyants Class Reference	90
6.56	warninghenri Class Reference	91
7	File Documentation	93
7.1	serveur/Hugo/hugo_compteur.cpp File Reference	93
7.1.1	Detailed Description	93
7.2	serveur/Hugo/hugo_ecran.cpp File Reference	94
7.2.1	Detailed Description	94
7.3	serveur/Hugo/hugo_scene.cpp File Reference	94
7.3.1	Detailed Description	95
7.4	serveur/Hugo/hugo_voyant_warning.cpp File Reference	95
7.4.1	Detailed Description	95
7.5	serveur/Hugo/hugo_voyants_simples.cpp File Reference	96
7.5.1	Detailed Description	96

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
à faire	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	12
cadrantVirtuel	18
jaugeVirtuel	53
lea_sceneDeFond	63
mygraphicItem	71
objet_virtuel	72
afficheKmHenri	11
CadrantEss	14
CadrantFlorian	15
cadrantHenri	16
CadrantTourParMin	17
CadrantVitesse	19
clignot	20
deux_voyants_lea	22
FlorianClignotant	23
FlorianCompteurKm	25
FlorianVoyant	26
FuelGauge_lea	27
hugo_boite_vitesse	30
hugo_Compteur	31
hugo_ecran	35
hugo_MyGraphicsitem	37
hugo_voyant_warning	41
hugo_voyants_clignotant	43
hugo_voyants_simples	45
jaugeBatterieHenri	47
jaugeClignotantHenri	48
jaugeEssenceHenri	49
jaugeTemperatureHenri	51
jaugeToursMinuteHenri	52
Jonas_compteur	54
jonas_voyant_simple	57
lea_clignottants	60
Leo_gauge	64

Leo_indicator	65
Quatre_Voyants_Lea	76
speedometer_Lea	84
TachometerGauge_Lea	85
voyant_Lea	88
Voyants	90
Voyant_batterie	87
warninghenri	91
phares_Lea	75
sceneDeFond	79
sceneDeFondHenri	80
QGraphicsScene	
scene_globale	77
henri_scene	28
hugo_scene	39
Jonas_scene	55
karim_scene	58
Lea_scene	61
Leo_scene	67
SceneFlorian	81
sceneGlobale	83
QGraphicsSimpleTextItem	
Leo_label	66
QMainWindow	
MainWindow	69
MainWindow	69
MainWindow	69
MainWindow	69

Chapter 4

Class Index

4.1 Class List

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

afficheKmHenri	11
aiguille_Lea	12
CadranEss	14
CadrantFlorian	15
cadrantHenri	16
CadranTourParMin	17
cadrantVirtuel	18
CadranVitesse	19
clignot	20
deux_voyants_lea	22
FlorianClignotant	23
FlorianCompteurKm	25
FlorianVoyant	26
FuelGauge_lea	27
henri_scene	28
hugo_boite_vitesse	30
hugo_Compteur	31
hugo_ecran	35
hugo_MyGraphicsitem	37
hugo_scene	39
hugo_voyant_warning	41
hugo_voyants_clignotant	43
hugo_voyants_simples	45
jaugeBatterieHenri	47
jaugeClignotantHenri	48
jaugeEssenceHenri	49
jaugeTemperatureHenri	51
jaugeToursMinuteHenri	52
jaugeVirtuel	53
Jonas_compteur	54
Jonas_scene	55
jonas_voyant_simple	57
karim_scene	58
lea_clignotants	60
Lea_scene	61

lea_sceneDeFond	63
Leo_gauge	64
Leo_indicator	65
Leo_label	66
Leo_scene	67
MainWindow	69
mygraphicItem	71
objet_virtuel	72
phares_Lea	75
Quatre_Voyants_Lea	76
scene_globale	77
sceneDeFond	79
sceneDeFondHenri	80
SceneFlorian	81
sceneGlobale	83
speedometer_Lea	84
TachometerGauge_Lea	85
Voyant_batterie	87
voyant_Lea	88
Voyants	90
warninghenri	91

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/Henri/affichekmhenri.h	??
serveur/Henri/cadranthenri.h	??
serveur/Henri/henri_scene.h	??
serveur/Henri/jaugebatteriehenri.h	??
serveur/Henri/jaugeclignotanthenri.h	??
serveur/Henri/jaugeessencehenri.h	??
serveur/Henri/jaugetemperaturehenri.h	??
serveur/Henri/jaugetoursminutehenri.h	??
serveur/Henri/mainwindow.h	??
serveur/Henri/scenedefondhenri.h	??
serveur/Henri/warninghenri.h	??
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	93
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	94
serveur/Hugo/hugo_ecran.h	??
serveur/Hugo/hugo_mygraphicsitem.h	??

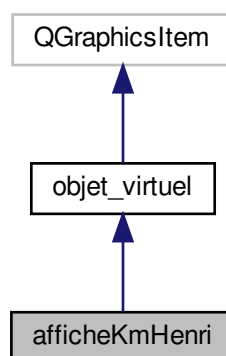
serveur/Hugo/hugo_scene.cpp	
Classe dérivée de scene_globale où sont ajoutés et paramétrés tous les composants du tableau de bord. Tous les objets doivent être définis au préalable dans scene_globale.h	94
serveur/Hugo/hugo_scene.h	??
serveur/Hugo/hugo_voyant_warning.cpp	
Classe dérivée de objet_virtuel permettant l'affichage d'un voyant warning clignotant	95
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	96
serveur/Hugo/hugo_voyants_simples.h	??
serveur/Jonas/jonas_compteur.h	??
serveur/Jonas/jonas_scene.h	??
serveur/Jonas/jonas_voyant_simple.h	??
serveur/Karim/cadranEss.h	??
serveur/Karim/cadrantourparmin.h	??
serveur/Karim/CadranVitesse.h	??
serveur/Karim/clignot.h	??
serveur/Karim/karim_scene.h	??
serveur/Karim/Voyant_batterie.h	??
serveur/Karim/voyants.h	??
serveur/Lea/aiguille_lea.h	??
serveur/Lea/deux_voyants_lea.h	??
serveur/Lea/fuelgauge_lea.h	??
serveur/Lea/lea_clignotants.h	??
serveur/Lea/lea_scene.h	??
serveur/Lea/lea_scenedefond.h	??
serveur/Lea/mainwindow.h	??
serveur/Lea/mygraphicitem.h	??
serveur/Lea/phares_lea.h	??
serveur/Lea/quatre_voyants_lea.h	??
serveur/Lea/speedometer_lea.h	??
serveur/Lea/tachometergauge_lea.h	??
serveur/Lea/voyant_lea.h	??
serveur/Leo/leo_gauge.h	??
serveur/Leo/leo_indicator.h	??
serveur/Leo/leo_label.h	??
serveur/Leo/leo_scene.h	??

Chapter 6

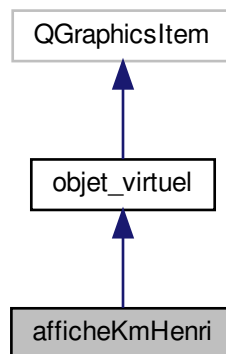
Class Documentation

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)

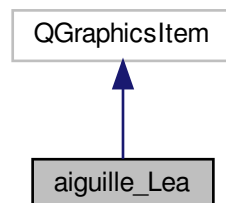
Additional Inherited Members

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

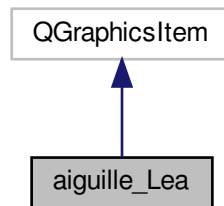
- serveur/Henri/affichekmhenri.h
- serveur/Henri/affichekmhenri.cpp

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

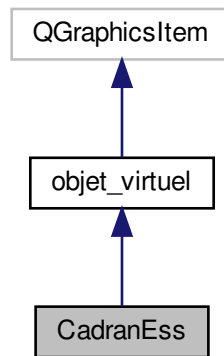
- double **x**
- double **y**
- double **r**
- double **taille_aiguille**
- int **vitesse**
- int **vitesse_max**
- int **angle_depart** =0
- int **angle_fin** =0
- int **angle** =0
- const double **pi** =3.14159265359

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

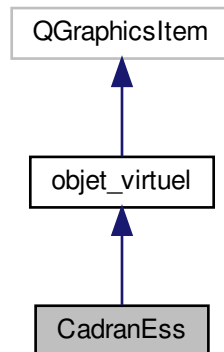
- serveur/Lea/aiguille_lea.h
- serveur/Lea/aiguille_lea.cpp

6.3 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)

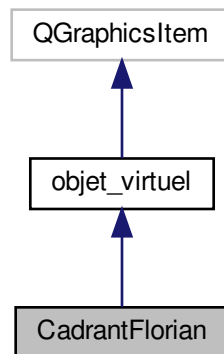
Additional Inherited Members

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

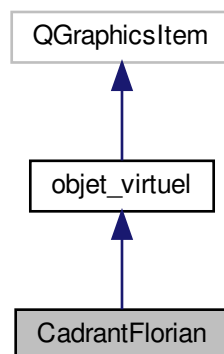
- serveur/Karim/cadranEss.h
- serveur/Karim/cadranEss.cpp

6.4 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)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- float **generateAngle** ()
- int **getSpeedMax** ()

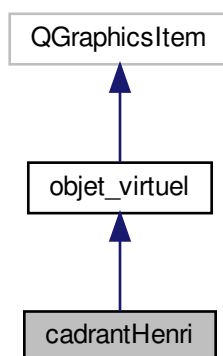
Additional Inherited Members

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

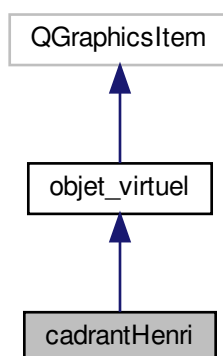
- serveur/Florian/cadrantflorian.h
- serveur/Florian/cadrantflorian.cpp

6.5 `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**

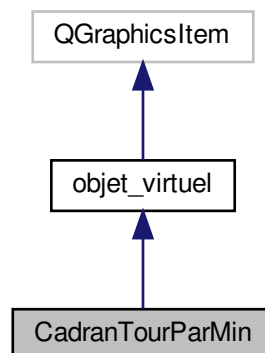
Additional Inherited Members

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

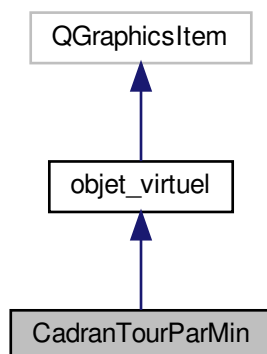
- serveur/Henri/cadranthenri.h
- serveur/Henri/cadranthenri.cpp

6.6 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)

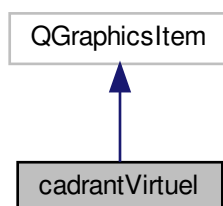
Additional Inherited Members

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

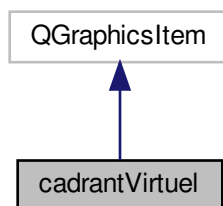
- serveur/Karim/cadrantourparmin.h
- serveur/Karim/cadrantourparmin.cpp

6.7 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

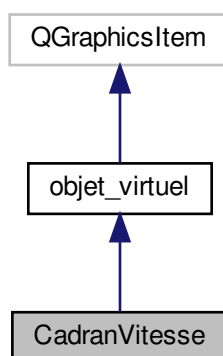
- `int` **`valeur`**
- `int` **`valeurMax`**

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

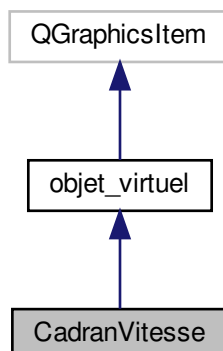
- `serveur/Florian/cadrantvirtuel.h`
- `serveur/Florian/cadrantvirtuel.cpp`

6.8 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)

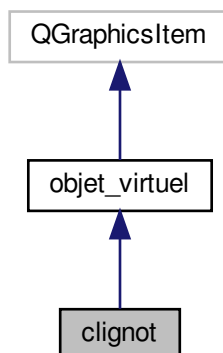
Additional Inherited Members

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

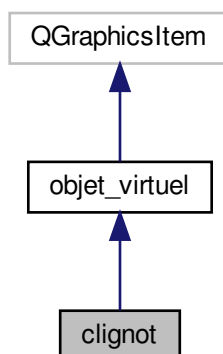
- serveur/Karim/CadranVitesse.h
- serveur/Karim/CadranVitesse.cpp

6.9 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**

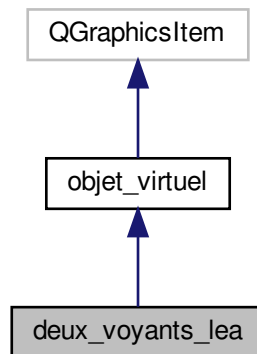
Additional Inherited Members

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

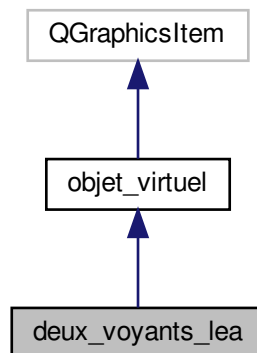
- serveur/Karim/clignot.h
- serveur/Karim/clignot.cpp

6.10 deux_voyants_lea Class Reference

Inheritance diagram for deux_voyants_lea:



Collaboration diagram for deux_voyants_lea:



Public Member Functions

- **deux_voyants_lea** ([objet_virtuel](#) *parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- void **setSize** (int, int)
- virtual void **parametrage** (int param_x, int param_y, QString param_chemin, QString param_chemin2)

Public Attributes

- QString **chemin**
- QString **chemin2**
- int **x**
- int **y**
- int **on** =0
- QSize **taille1**
- QSize **taille2**

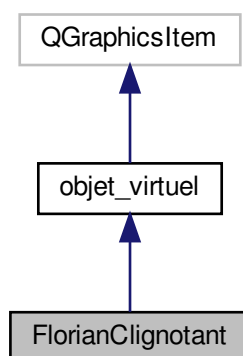
Additional Inherited Members

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

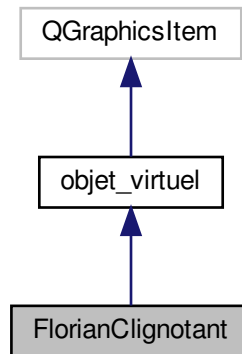
- serveur/Lea/deux_voyants_lea.h
- serveur/Lea/deux_voyants_lea.cpp

6.11 FlorianClignotant Class Reference

Inheritance diagram for FlorianClignotant:



Collaboration diagram for FlorianClignotant:



Public Member Functions

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

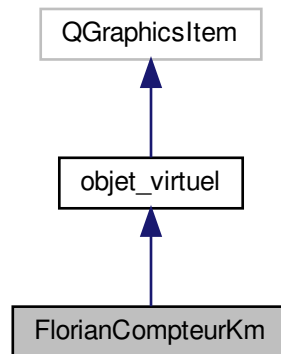
Additional Inherited Members

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

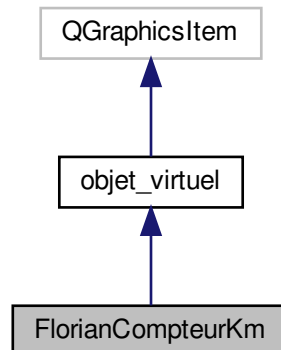
- `serveur/Florian/florianclignotant.h`
- `serveur/Florian/florianclignotant.cpp`

6.12 FlorianCompteurKm Class Reference

Inheritance diagram for FlorianCompteurKm:



Collaboration diagram for FlorianCompteurKm:



Public Member Functions

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

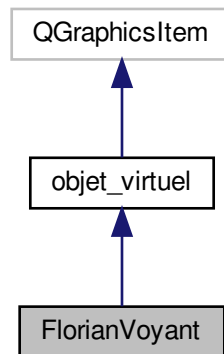
Additional Inherited Members

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

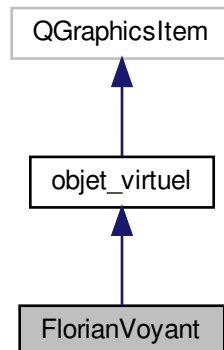
- `serveur/Florian/floriancompteurkm.h`
- `serveur/Florian/floriancompteurkm.cpp`

6.13 FlorianVoyant Class Reference

Inheritance diagram for FlorianVoyant:



Collaboration diagram for FlorianVoyant:



Public Member Functions

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

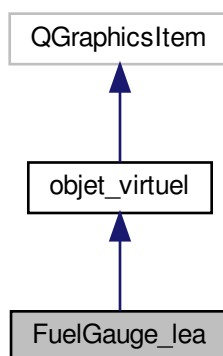
Additional Inherited Members

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

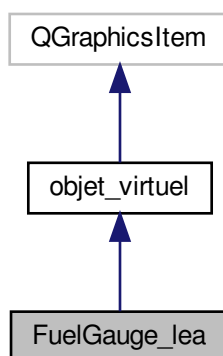
- serveur/Florian/florianvoyant.h
- serveur/Florian/florianvoyant.cpp

6.14 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)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- void **setValue** (int)
- virtual void **parametrage** (double param_x, double param_y, double param_width, double param_height, double param_x2, double param_y2, double param_x3, double param_y3)

Public Attributes

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

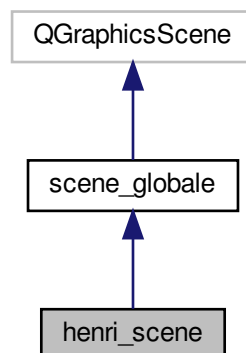
Additional Inherited Members

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

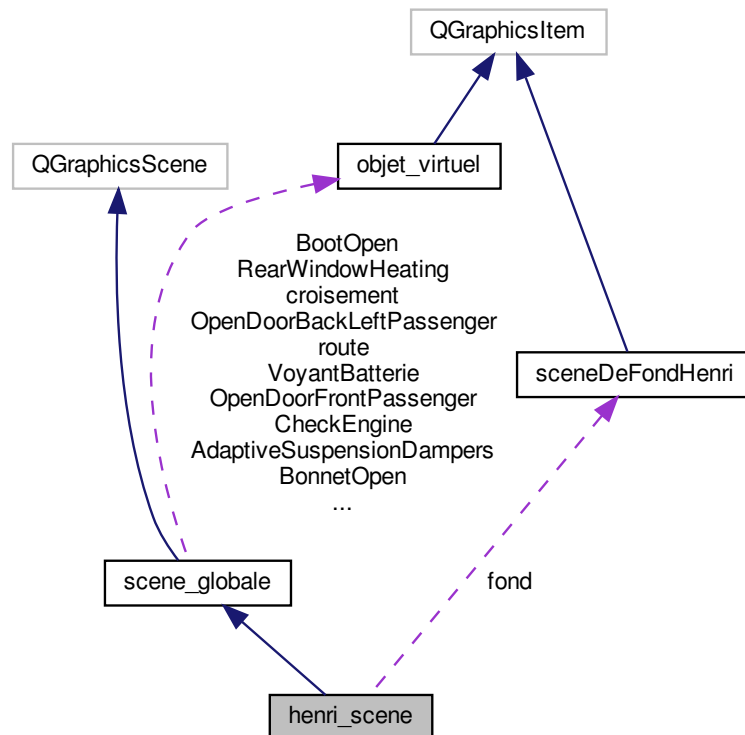
- serveur/Lea/fuelgauge_lea.h
- serveur/Lea/fuelgauge_lea.cpp

6.15 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

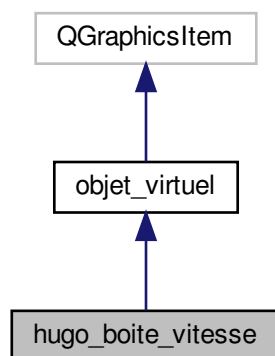
- [sceneDeFondHenri](#) * **fond**

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

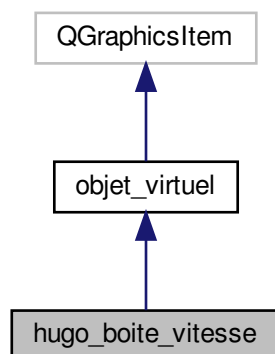
- serveur/Henri/henri_scene.h
- serveur/Henri/henri_scene.cpp

6.16 hugo_boite_vitesse Class Reference

Inheritance diagram for hugo_boite_vitesse:



Collaboration diagram for hugo_boite_vitesse:



Public Member Functions

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

Additional Inherited Members

6.16.1 Member Function Documentation

6.16.1.1 paint()

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

[hugo_ecran::paint](#)

Parameters

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

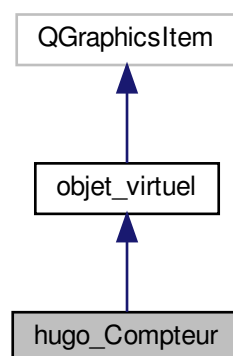
La fonction paint va permettre de réaliser un affichage pour le compteur kilometrique et l'heure. Dans un premier temps, affichage des objets fixes, puis on récupère et affiche les différentes variables.

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

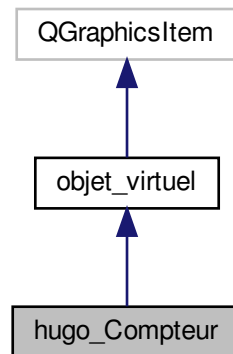
- serveur/Hugo/hugo_boite_vitesse.h
- serveur/Hugo/hugo_boite_vitesse.cpp

6.17 hugo_Compteur Class Reference

Inheritance diagram for hugo_Compteur:



Collaboration diagram for hugo_Compteur:



Public Member Functions

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

Protected Attributes

- int **x** =0
- int **y** =0
- int **r** =100
- int **start_angle** =0
- int **end_angle** =360
- int **nbre_graduations** =12
- int **critique**
- int **angle** =0
- int **value2** =0
- int **r_verre**
- int **direction_grad** =1
- QColor **couleur** =QColor(100,100,100)
- QColor **couleur2** =QColor(100,100,100)
- QColor **couleurgrad** =QColor(100,100,100,50)
- QColor **couleurgrad2** =QColor(100,100,100,50)
- QStringList **graduations**

Additional Inherited Members

6.17.1 Constructor & Destructor Documentation

6.17.1.1 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>	

6.17.2 Member Function Documentation

6.17.2.1 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 traduit 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

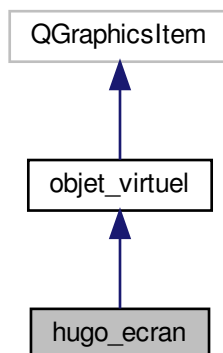
L'effet de verre est obtenu en superposant un cercle semi-transparent au compteur. L'option `direction_grad` permet de définir le sens du gradient.

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

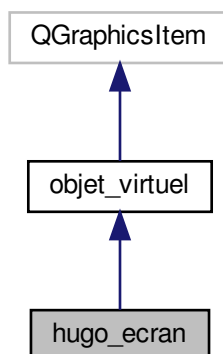
- `serveur/Hugo/hugo_compteur.h`
- `serveur/Hugo/hugo_compteur.cpp`

6.18 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

6.18.1 Constructor & Destructor Documentation

6.18.1.1 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.

6.18.2 Member Function Documentation

6.18.2.1 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

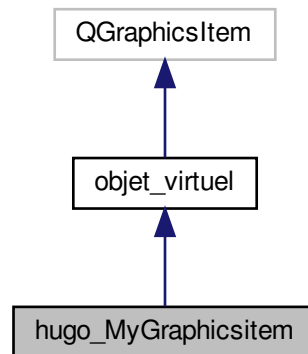
Affichage des différents textes. Le kilométrage est récupéré grâce à la fonction getValue.

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

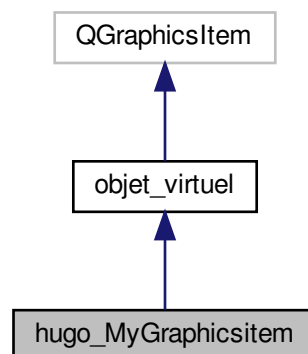
- serveur/Hugo/hugo_ecran.h
- serveur/Hugo/[hugo_ecran.cpp](#)

6.19 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

6.19.1 Constructor & Destructor Documentation

6.19.1.1 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.

6.19.2 Member Function Documentation

6.19.2.1 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

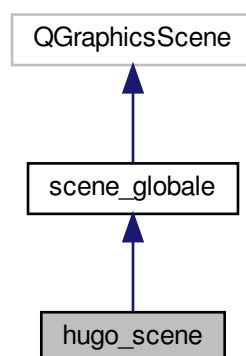
Affichage d'une icone jauge d'essence

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

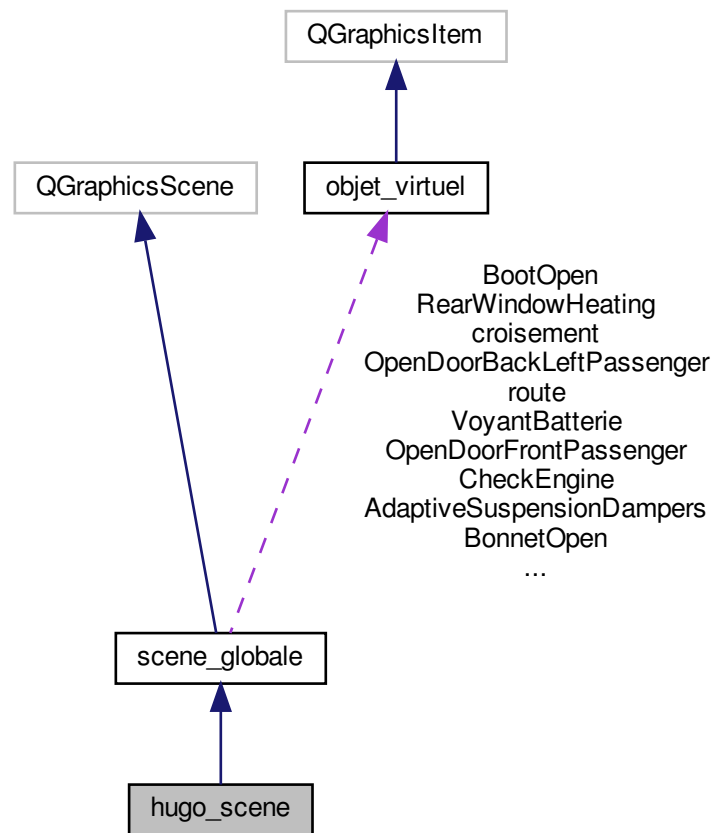
- serveur/Hugo/hugo_mygraphicsitem.h
- serveur/Hugo/hugo_mygraphicsitem.cpp

6.20 hugo_scene Class Reference

Inheritance diagram for hugo_scene:



Collaboration diagram for hugo_scene:



Public Member Functions

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

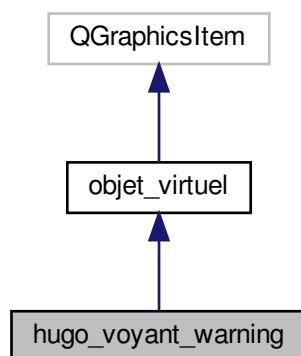
Additional Inherited Members

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

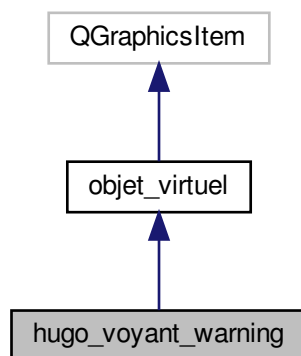
- `serveur/Hugo/hugo_scene.h`
- `serveur/Hugo/hugo_scene.cpp`

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

Public Attributes

- int **cligno**

Additional Inherited Members

6.21.1 Constructor & Destructor Documentation

6.21.1.1 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.

6.21.2 Member Function Documentation

6.21.2.1 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.

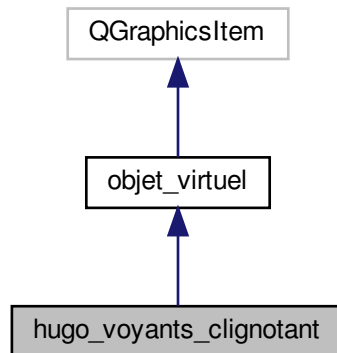
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

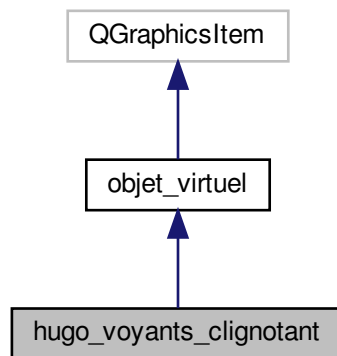
- serveur/Hugo/hugo_voyant_warning.h
- serveur/Hugo/[hugo_voyant_warning.cpp](#)

6.22 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

6.22.1 Constructor & Destructor Documentation

6.22.1.1 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.

6.22.2 Member Function Documentation

6.22.2.1 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.

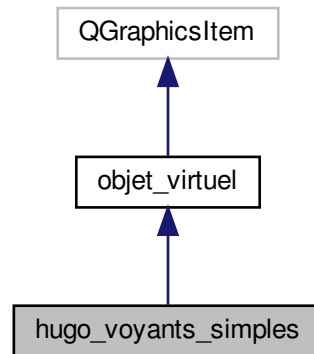
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné

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

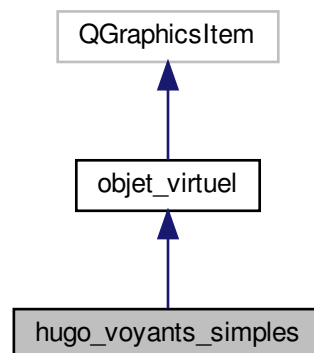
- serveur/Hugo/hugo_voyants_clignotant.h
- serveur/Hugo/hugo_voyants_clignotant.cpp

6.23 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

6.23.1 Constructor & Destructor Documentation

6.23.1.1 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

6.23.2 Member Function Documentation

6.23.2.1 paint()

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

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

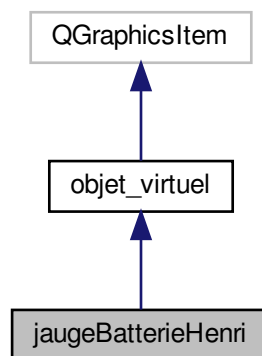
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché.

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

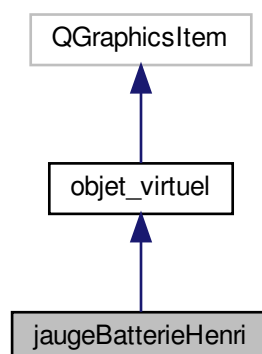
- serveur/Hugo/hugo_voyants_simples.h
- serveur/Hugo/[hugo_voyants_simples.cpp](#)

6.24 `jaugeBatterieHenri` Class Reference

Inheritance diagram for `jaugeBatterieHenri`:



Collaboration diagram for `jaugeBatterieHenri`:



Public Member Functions

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

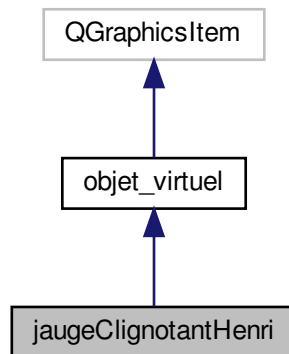
Additional Inherited Members

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

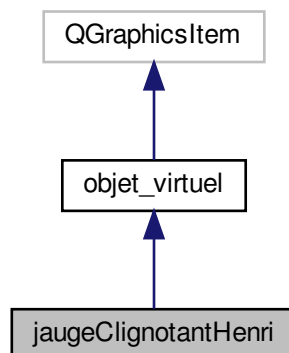
- serveur/Henri/jaugebatteriehenri.h
- serveur/Henri/jaugebatteriehenri.cpp

6.25 jaugeClignotantHenri Class Reference

Inheritance diagram for jaugeClignotantHenri:



Collaboration diagram for jaugeClignotantHenri:



Public Member Functions

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

Public Attributes

- int **cligno**

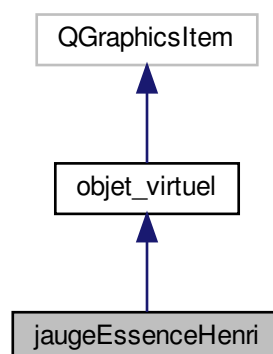
Additional Inherited Members

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

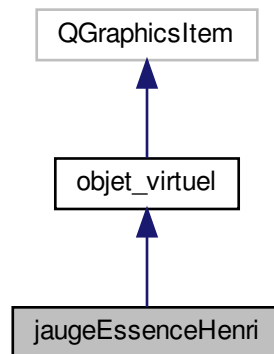
- serveur/Henri/jaugeclignotanthenri.h
- serveur/Henri/jaugeclignotanthenri.cpp

6.26 `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)

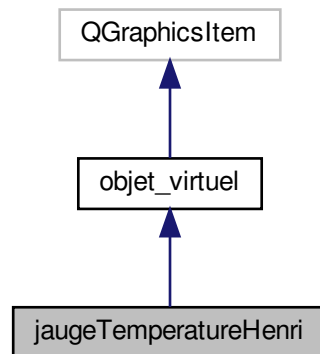
Additional Inherited Members

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

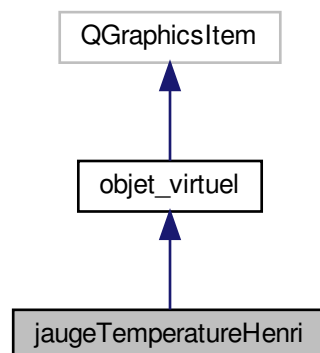
- `serveur/Henri/jaugeessencehenri.h`
- `serveur/Henri/jaugeessencehenri.cpp`

6.27 `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)`

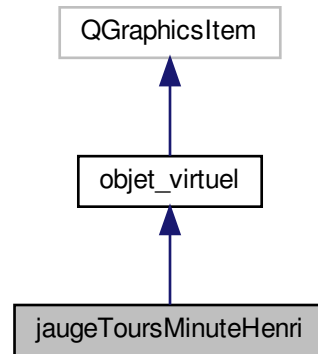
Additional Inherited Members

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

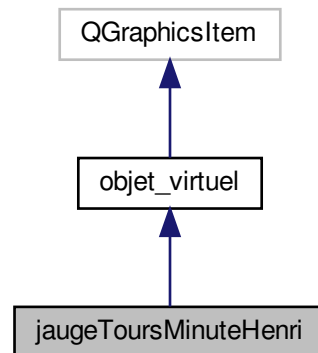
- `serveur/Henri/jaugetemperaturehenri.h`
- `serveur/Henri/jaugetemperaturehenri.cpp`

6.28 `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`

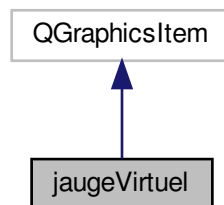
Additional Inherited Members

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

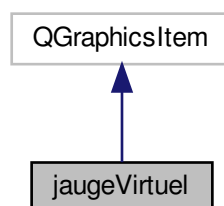
- serveur/Henri/jaugetoursminutehenri.h
- serveur/Henri/jaugetoursminutehenri.cpp

6.29 **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

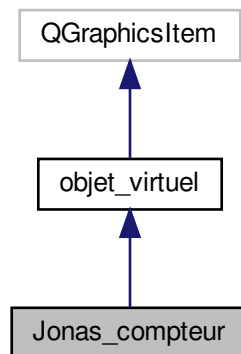
- int **valeur**
- int **valeurMax**

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

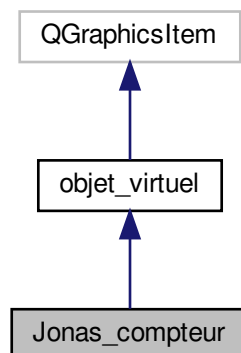
- serveur/Florian/jaugevirtuel.h
- serveur/Florian/jaugevirtuel.cpp

6.30 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, QString textCenter, 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)

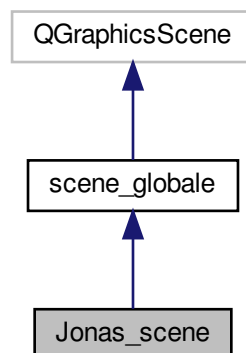
Additional Inherited Members

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

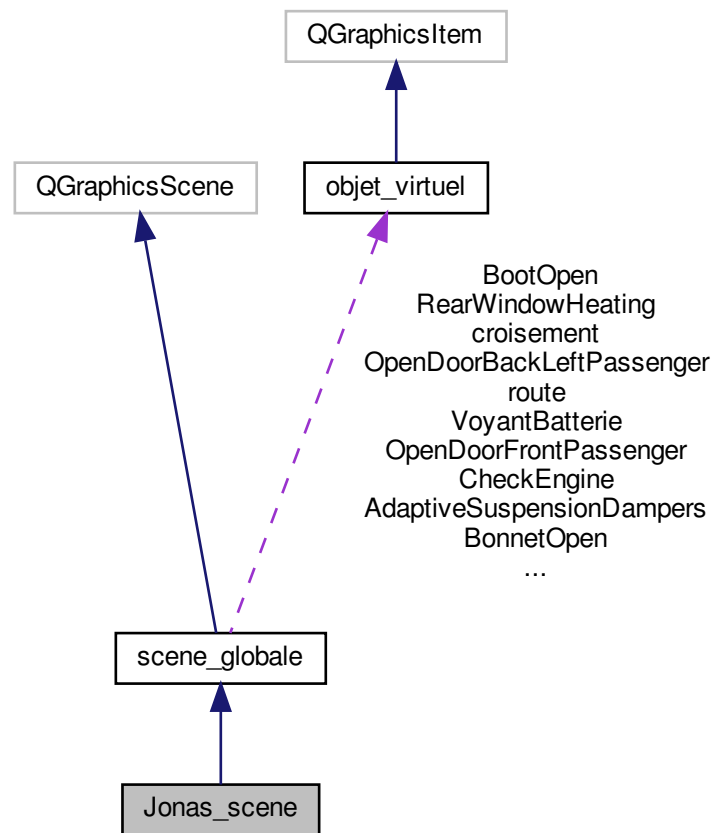
- serveur/Jonas/jonas_compteur.h
- serveur/Jonas/jonas_compteur.cpp

6.31 Jonas_scene Class Reference

Inheritance diagram for Jonas_scene:



Collaboration diagram for Jonas_scene:



Public Member Functions

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

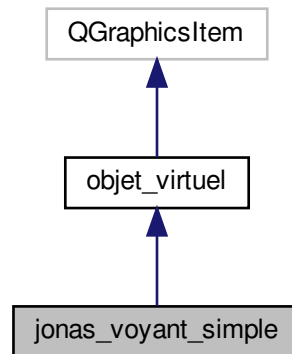
Additional Inherited Members

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

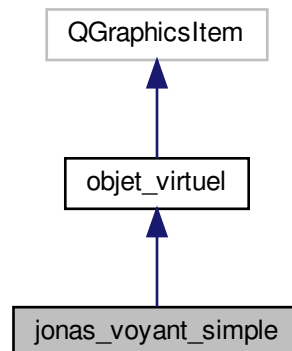
- serveur/Jonas/jonas_scene.h
- serveur/Jonas/jonas_scene.cpp

6.32 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** ()

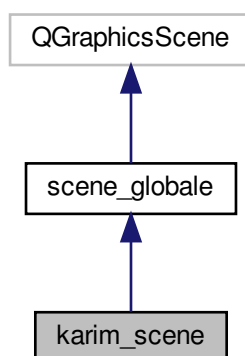
Additional Inherited Members

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

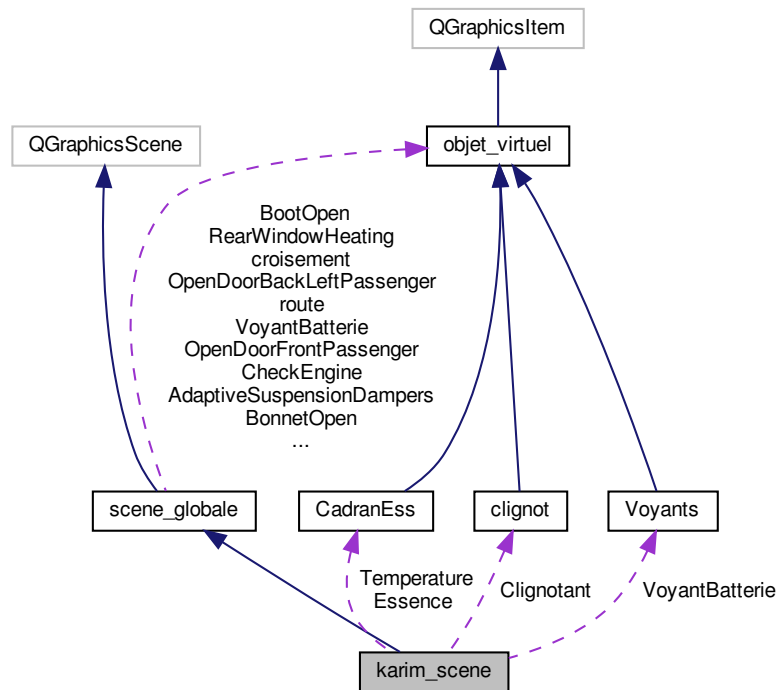
- serveur/Jonas/jonas_voyant_simple.h
- serveur/Jonas/jonas_voyant_simple.cpp

6.33 karim_scene Class Reference

Inheritance diagram for karim_scene:



Collaboration diagram for karim_scene:



Public Attributes

- [CadranEss](#) * **Essence**
- [CadranEss](#) * **Temperature**
- [clignot](#) * **Clignotant**
- [Voyants](#) * **VoyantBatterie**

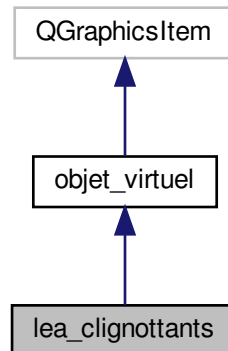
Additional Inherited Members

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

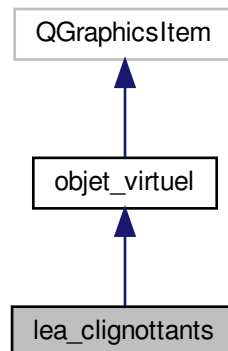
- `serveur/Karim/karim_scene.h`
- `serveur/Karim/karim_scene.cpp`

6.34 lea_clignottants Class Reference

Inheritance diagram for lea_clignottants:



Collaboration diagram for lea_clignottants:



Public Member Functions

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

Public Attributes

- int **cligno**

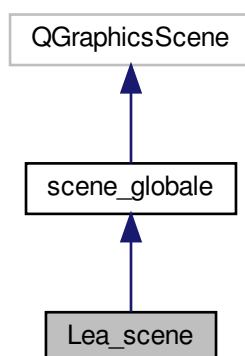
Additional Inherited Members

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

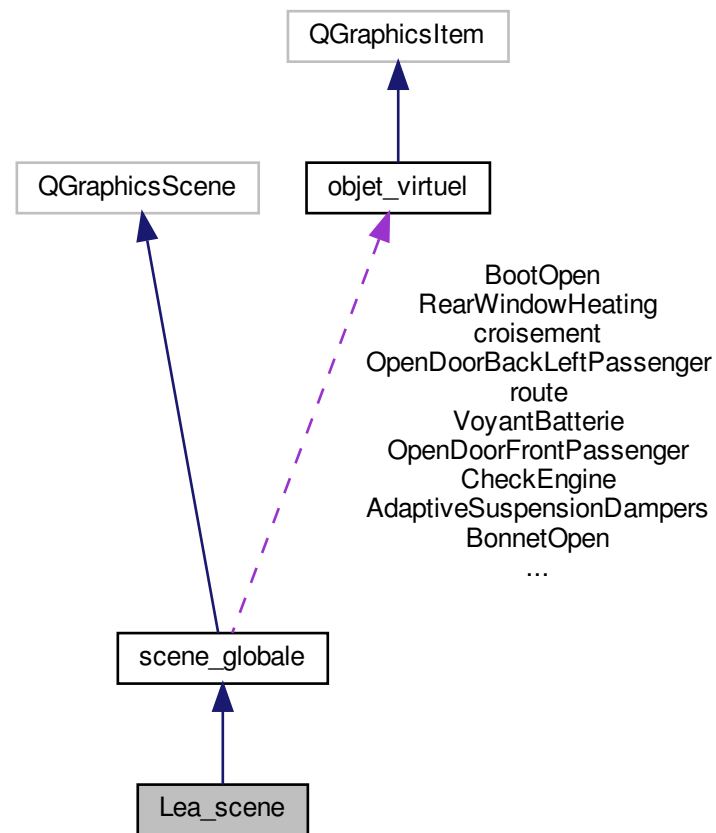
- serveur/Lea/lea_clignottants.h
- serveur/Lea/lea_clignottants.cpp

6.35 Lea_scene Class Reference

Inheritance diagram for Lea_scene:



Collaboration diagram for Lea_scene:



Public Member Functions

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

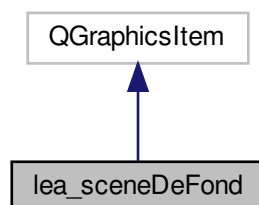
Additional Inherited Members

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

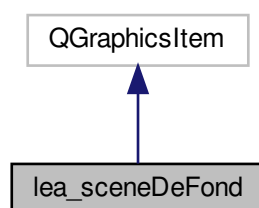
- serveur/Lea/lea_scene.h
- serveur/Lea/lea_scene.cpp

6.36 lea_sceneDeFond Class Reference

Inheritance diagram for lea_sceneDeFond:



Collaboration diagram for lea_sceneDeFond:



Public Member Functions

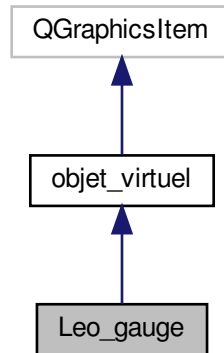
- **lea_sceneDeFond** (`QGraphicsItem *parent=nullptr`)
- `QRectF` **boundingRect** () const
- void **paint** (`QPainter *painter`, const `QStyleOptionGraphicsItem *option`, `QWidget *widget`)

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

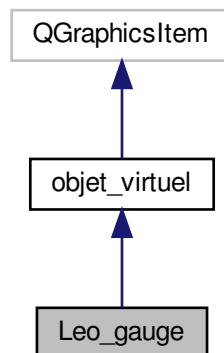
- `serveur/Lea/lea_scenedefond.h`
- `serveur/Lea/lea_scenedefond.cpp`

6.37 Leo_gauge Class Reference

Inheritance diagram for Leo_gauge:



Collaboration diagram for Leo_gauge:



Public Member Functions

- **Leo_gauge** (`objet_virtuel` *parent=nullptr)
- `QRectF` **boundingRect** () const override
- void **paint** (QPainter *, const QStyleOptionGraphicsItem *, QWidget *) override

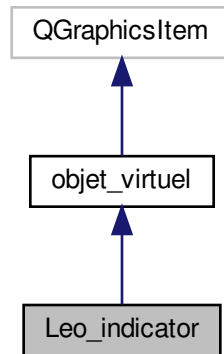
Additional Inherited Members

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

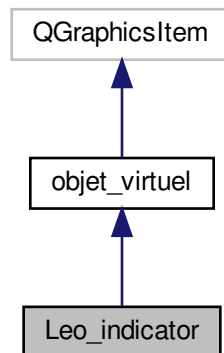
- serveur/Leo/leo_gauge.h

6.38 Leo_indicator Class Reference

Inheritance diagram for Leo_indicator:



Collaboration diagram for Leo_indicator:



Public Member Functions

- **Leo_indicator** ([objet_virtuel](#) *parent=nullptr)
- **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** = Qt::darkGreen
- QColor **ColorOn** = Qt::green

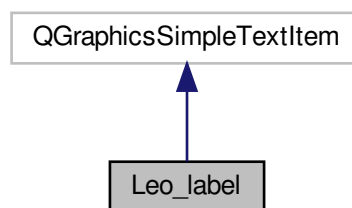
Additional Inherited Members

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

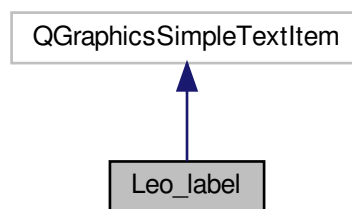
- serveur/Leo/leo_indicator.h
- serveur/Leo/leo_indicator.cpp

6.39 Leo_label Class Reference

Inheritance diagram for Leo_label:



Collaboration diagram for Leo_label:



Public Member Functions

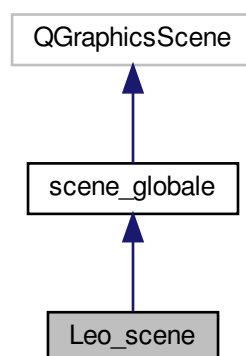
- **Leo_label** (`QGraphicsItem *parent=NULLPTR`)
- **Leo_label** (`const QString &text, QGraphicsItem *parent=NULLPTR`)
- void **SetFontSizePx** (`const int`)

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

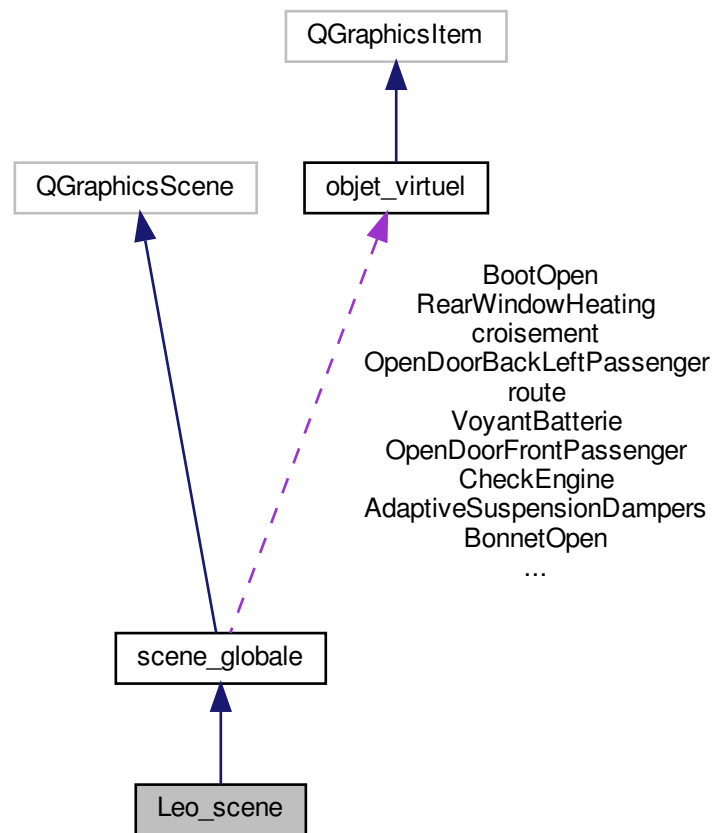
- serveur/Leo/leo_label.h
- serveur/Leo/leo_label.cpp

6.40 Leo_scene Class Reference

Inheritance diagram for Leo_scene:



Collaboration diagram for Leo_scene:



Public Member Functions

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

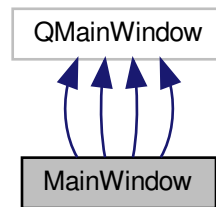
Additional Inherited Members

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

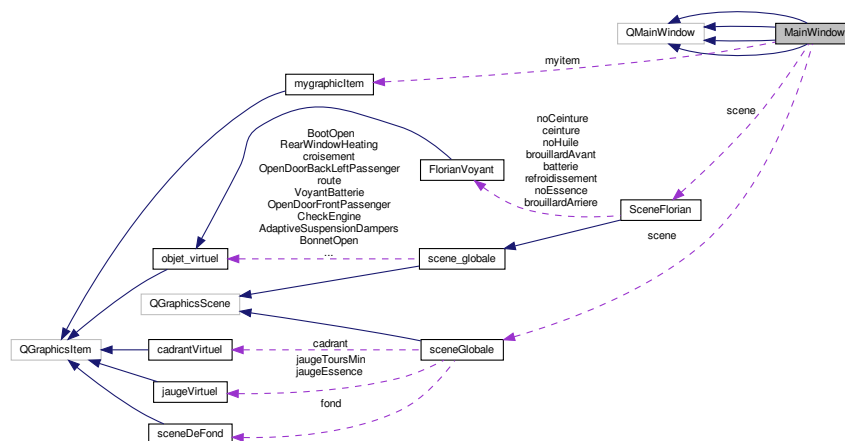
- serveur/Leo/leo_scene.h
- serveur/Leo/leo_scene.cpp

6.41 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.

Public Attributes

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

6.41.1 Constructor & Destructor Documentation

6.41.1.1 MainWindow()

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

La scène par défaut est

6.41.2 Member Function Documentation

6.41.2.1 acceleration()

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

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

Fonction simulant une accélération linéaire avec changements de rapports de vitesse

Parameters

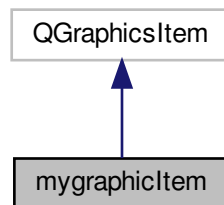
<i>time</i>	Durée de la simulation
-------------	------------------------

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

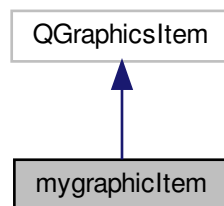
- serveur/Florian/mainwindow.h
- serveur/Florian/mainwindow.cpp

6.42 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**

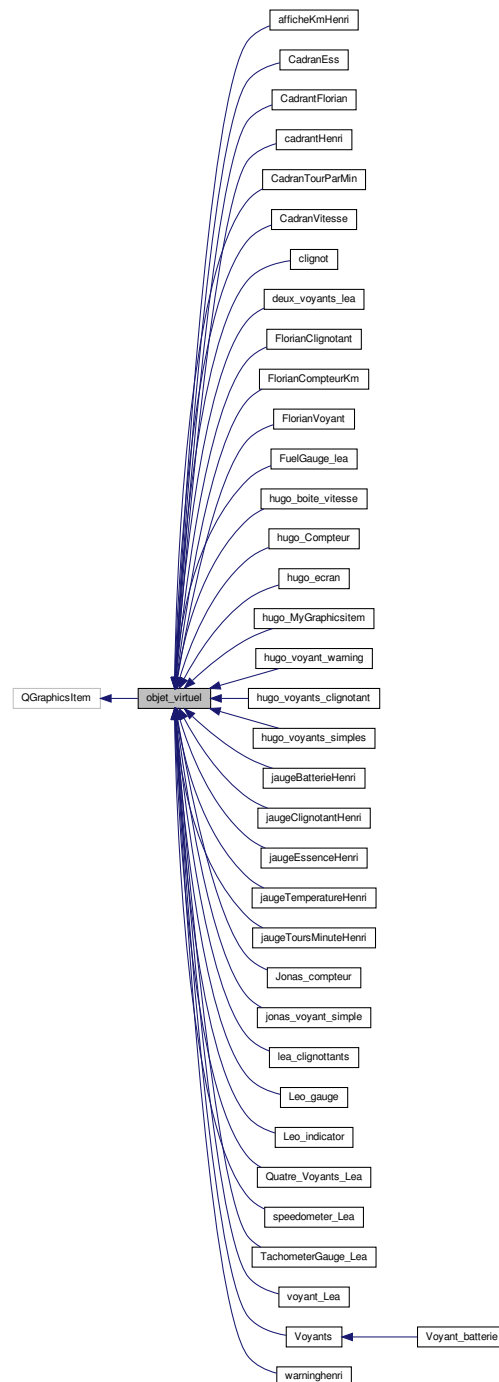
- const double **pi** =3.14159265359

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

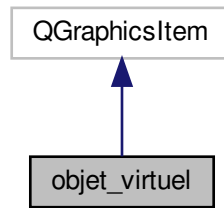
- serveur/Lea/mygraphicitem.h
- serveur/Lea/mygraphicitem.cpp

6.43 objet_virtuel Class Reference

Inheritance diagram for objet_virtuel:



Collaboration diagram for objet_virtuel:



Public Member Functions

- `objet_virtuel` (`QGraphicsItem *parent=nullptr`)
- float `getValue` () const
Fonction renvoyant la variable value.
- void `setValue` (float 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**

6.43.1 Constructor & Destructor Documentation

6.43.1.1 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

6.43.2 Member Function Documentation

6.43.2.1 `getValue()`

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

Fonction renvoyant la variable value.

Returns

6.43.2.2 `getValueMax()`

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

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

Returns

6.43.2.3 `setValue()`

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

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

Parameters

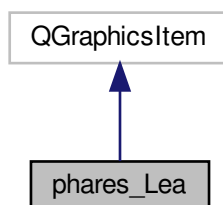
<i>set_value</i>	valeur à attribuer à value
------------------	----------------------------

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

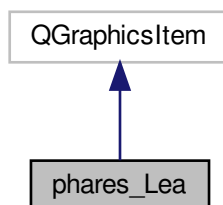
- serveur/objet_virtuel.h
- serveur/objet_virtuel.cpp

6.44 phares_Lea Class Reference

Inheritance diagram for phares_Lea:



Collaboration diagram for phares_Lea:

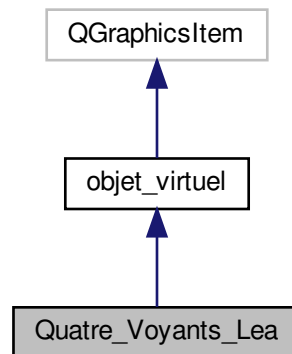


The documentation for this class was generated from the following files:

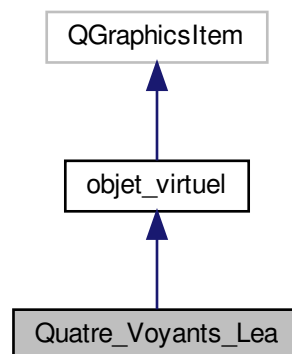
- serveur/Lea/phares_lea.h
- serveur/Lea/phares_lea.cpp

6.45 Quatre_Voyants_Lea Class Reference

Inheritance diagram for Quatre_Voyants_Lea:



Collaboration diagram for Quatre_Voyants_Lea:



Public Member Functions

- **Quatre_Voyants_Lea** ([objet_virtuel](#) *parent=nullptr)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- void **setSize** (int, int)
- virtual void **parametrage** (int param_x, int param_y, QString param_chemin, QString param_chemin2, Q<←String param_chemin3)

Public Attributes

- QString **chemin**
- QString **chemin2**
- QString **chemin3**
- int **x**
- int **y**
- int **on** =0
- QSize **taille1**
- QSize **taille2**

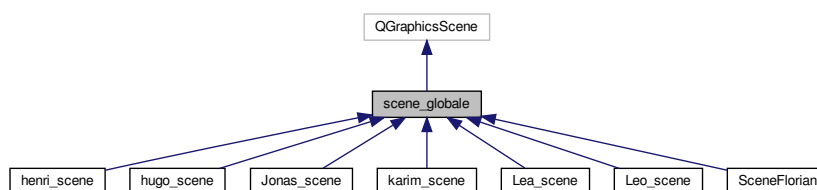
Additional Inherited Members

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

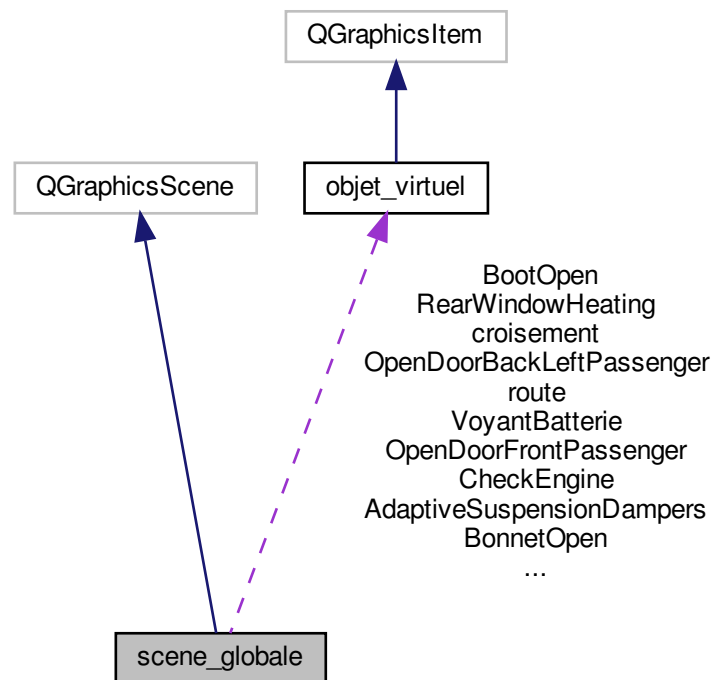
- serveur/Lea/quatre_voyants_lea.h
- serveur/Lea/quatre_voyants_lea.cpp

6.46 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**

6.46.1 Constructor & Destructor Documentation

6.46.1.1 scene_globale()

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

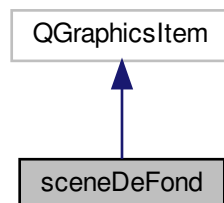
Classe dérivant de QGraphicsScene. Va servir de classe mère pour toutes les scènes, ce qui permettra de passer d'une scène à l'autre de façon dynamique. Tous les objets utilisés dans le dashboard seront définis dans "scène_↔_globale.h"

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

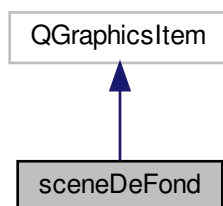
- serveur/scene_globale.h
- serveur/scene_globale.cpp

6.47 sceneDeFond Class Reference

Inheritance diagram for sceneDeFond:



Collaboration diagram for sceneDeFond:



Public Member Functions

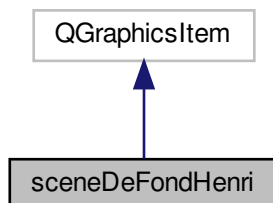
- **sceneDeFond** (`QGraphicsItem *parent=nullptr`)

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

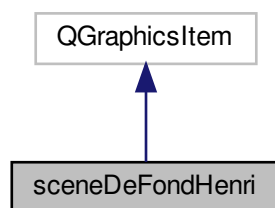
- `serveur/Florian/scenedefond.h`
- `serveur/Florian/scenedefond.cpp`

6.48 sceneDeFondHenri Class Reference

Inheritance diagram for sceneDeFondHenri:



Collaboration diagram for sceneDeFondHenri:



Public Member Functions

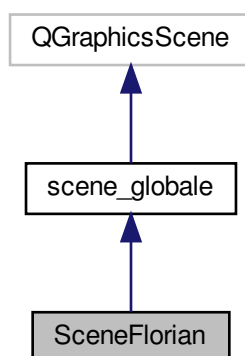
- **sceneDeFondHenri** (`QGraphicsItem *parent=nullptr`)
- void **paint** (`QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget`)
- `QRectF` **boundingRect** () const

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

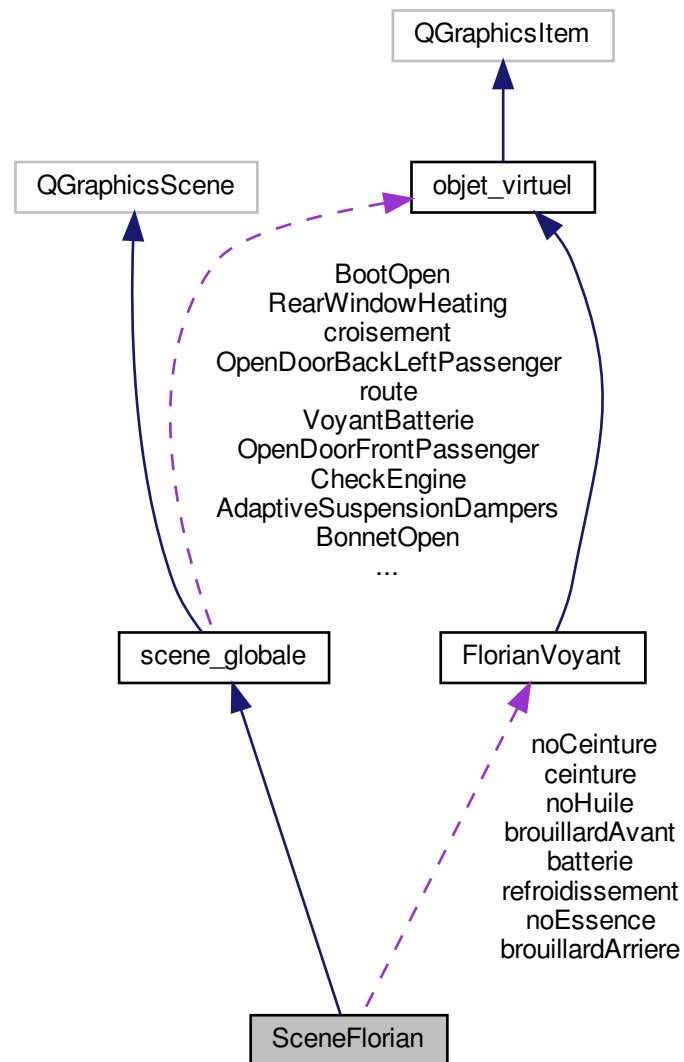
- `serveur/Henri/scenedefondhenri.h`
- `serveur/Henri/scenedefondhenri.cpp`

6.49 SceneFlorian Class Reference

Inheritance diagram for SceneFlorian:



Collaboration diagram for SceneFlorian:



Public Member Functions

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

Public Attributes

- **FlorianVoyant** * **brouillardAvant**
- **FlorianVoyant** * **brouillardArriere**
- **FlorianVoyant** * **noEssence**
- **FlorianVoyant** * **noHuile**
- **FlorianVoyant** * **refroidissement**

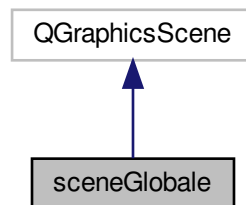
- [FlorianVoyant](#) * **batterie**
- [FlorianVoyant](#) * **ceinture**
- [FlorianVoyant](#) * **noCeinture**

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

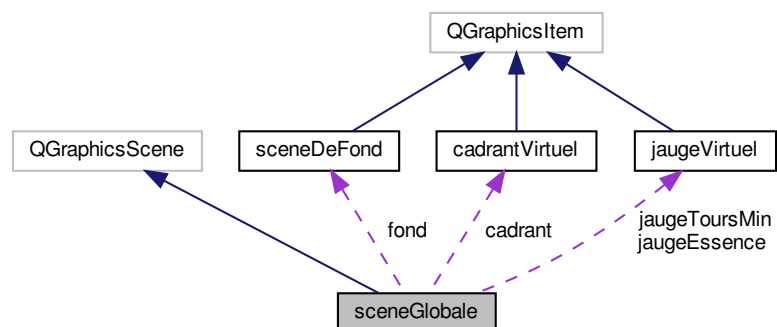
- serveur/Florian/sceneflorian.h
- serveur/Florian/sceneflorian.cpp

6.50 sceneGlobale Class Reference

Inheritance diagram for sceneGlobale:



Collaboration diagram for sceneGlobale:



Public Member Functions

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

Public Attributes

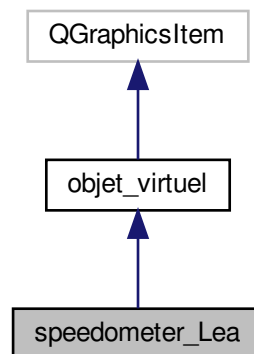
- [sceneDeFond](#) * **fond**
- [cadrantVirtuel](#) * **cadrant**
- [jaugeVirtuel](#) * **jaugeEssence**
- [jaugeVirtuel](#) * **jaugeToursMin**

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

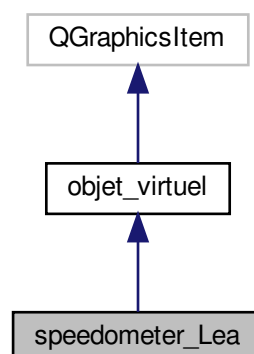
- serveur/Florian/sceneglobale.h
- serveur/Florian/sceneglobale.cpp

6.51 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, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- virtual void **parametrage** (double param_x, double param_y, double param_r, int param_start, int param_end, int param_spanAngle, int param_vitMax)
- int **getValueMax** ()
- void **setValue** (int)

Public Attributes

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

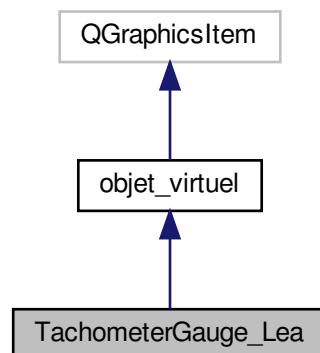
Additional Inherited Members

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

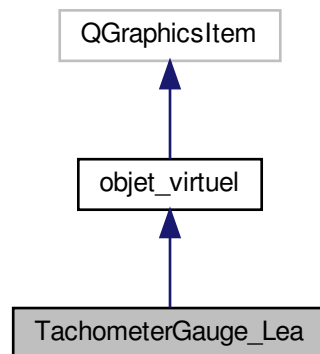
- serveur/Lea/speedometer_lea.h
- serveur/Lea/speedometer_lea.cpp

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

Protected Member Functions

- int **getValueMax** ()
- void **setValue** (int)
- virtual void **parametrage** (double param_x, double param_y, double param_r, int param_start, int param_end, int param_spanAngle, int param_graduation, int param_vitMax)

Protected Attributes

- double **r**
- double **x**
- double **y**
- int **angle_debut**
- int **angle_fin**
- int **nb_graduation** =41
- int **span_angle**
- int **vitesse_max** =270
- const double **pi** =3.14159265359

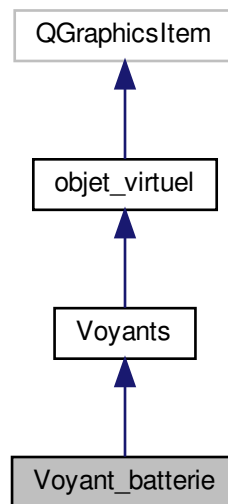
Additional Inherited Members

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

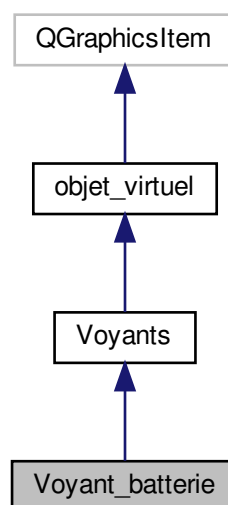
- serveur/Lea/tachometergauge_lea.h
- serveur/Lea/tachometergauge_lea.cpp

6.53 Voyant_batterie Class Reference

Inheritance diagram for Voyant_batterie:



Collaboration diagram for Voyant_batterie:



Public Member Functions

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

Protected Attributes

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

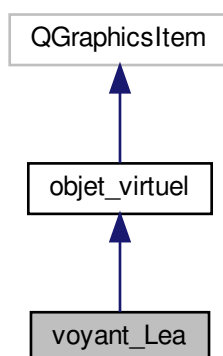
Additional Inherited Members

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

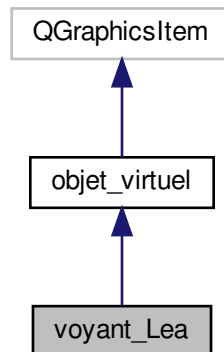
- serveur/Karim/Voyant_batterie.h
- serveur/Karim/Voyant_batterie.cpp

6.54 voyant_Lea Class Reference

Inheritance diagram for voyant_Lea:



Collaboration diagram for voyant_Lea:



Public Member Functions

- **voyant_Lea** (int, int, QString, int, int)
- QRectF **boundingRect** () const
- void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- virtual void **parametrage** (int param_x, int param_y, QString param_chemin)

Public Attributes

- QString **chemin**
- int **x**
- int **y**
- int **on** =0
- int **width**
- int **height**

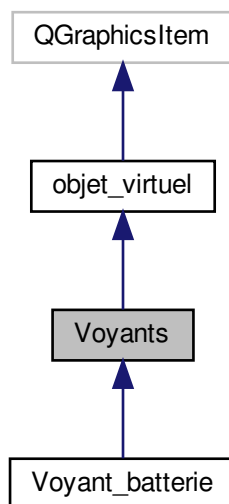
Additional Inherited Members

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

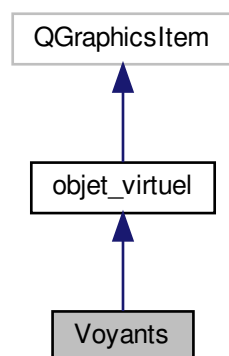
- serveur/Lea/voyant_lea.h
- serveur/Lea/voyant_lea.cpp

6.55 Voyants Class Reference

Inheritance diagram for Voyants:



Collaboration diagram for Voyants:



Public Member Functions

- int **getValue** () const
- void **setValue** (int value)
- int **getValueMax** () const

Protected Attributes

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

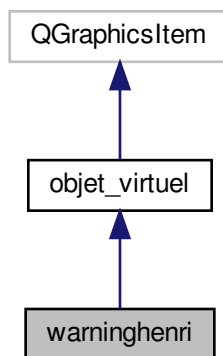
Additional Inherited Members

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

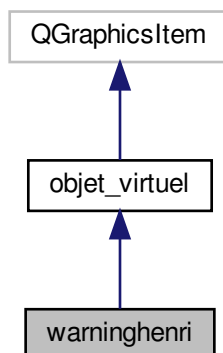
- serveur/Karim/voyants.h
- serveur/Karim/voyants.cpp

6.56 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

Chapter 7

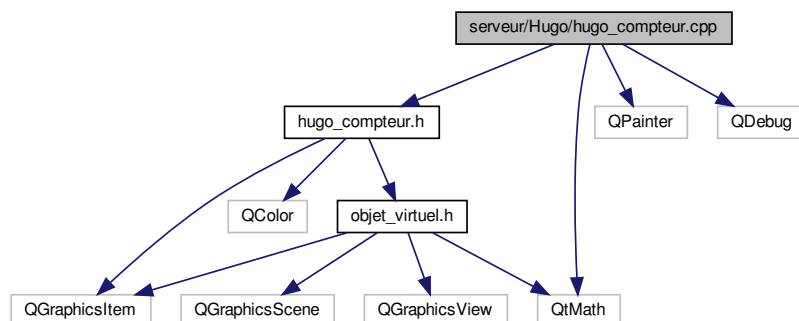
File Documentation

7.1 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`

7.1.1 Detailed Description

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

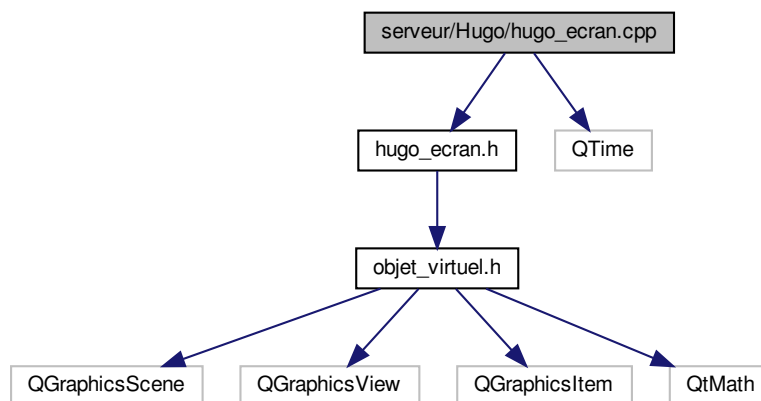
Classe héritée de `objet_virtuel`. Utilisée pour l'affichage du compteur de vitesse, du compteur rpm, de la jauge d'essence ainsi que de la jauge de température. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction `paint` permet l'affichage.

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

7.2 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:
```



7.2.1 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.

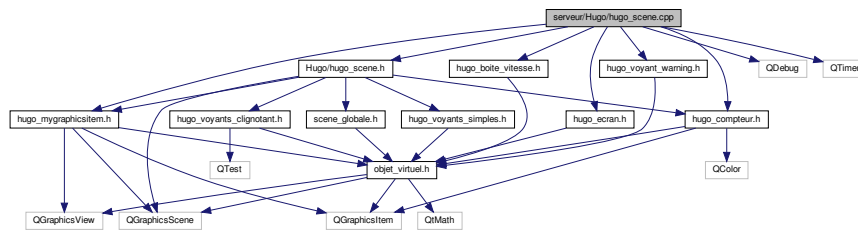
7.3 serveur/Hugo/hugo_scene.cpp File Reference

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:



7.3.1 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](#).

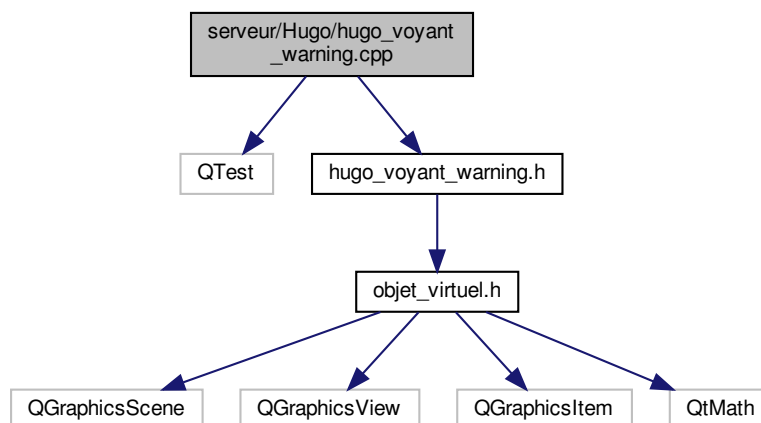
7.4 serveur/Hugo/hugo_voyant_warning.cpp File Reference

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:



7.4.1 Detailed Description

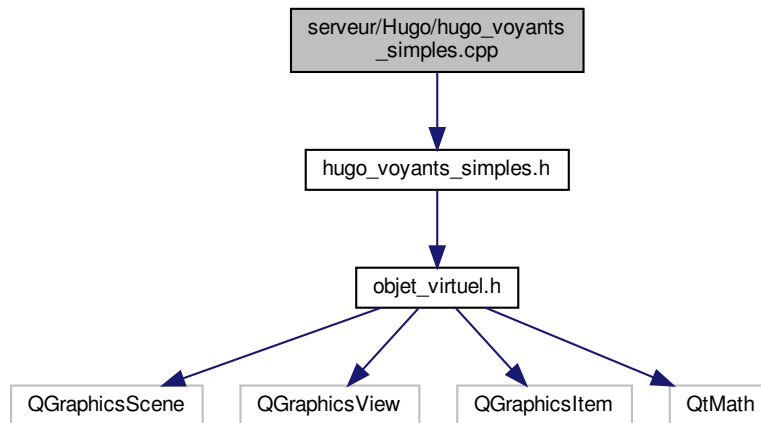
Classe dérivée de [objet_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

7.5 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:



7.5.1 Detailed Description

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