

Doc car_dashboard

Generated by Doxygen 1.8.13

Contents

1	README	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	CadrantFlorian Class Reference	13
6.3	cadrantHenri Class Reference	14
6.4	cadrantVirtuel Class Reference	15
6.5	clignotant Class Reference	16
6.6	henri_scene Class Reference	18
6.7	hugo_Compteur Class Reference	20
6.7.1	Constructor & Destructor Documentation	21
6.7.1.1	hugo_Compteur()	21
6.7.2	Member Function Documentation	22
6.7.2.1	paint()	22
6.8	hugo_ecran Class Reference	23

6.8.1	Constructor & Destructor Documentation	24
6.8.1.1	hugo_ecran()	24
6.8.2	Member Function Documentation	24
6.8.2.1	paint()	24
6.9	hugo_MyGraphicsitem Class Reference	25
6.9.1	Constructor & Destructor Documentation	26
6.9.1.1	hugo_MyGraphicsitem()	26
6.9.2	Member Function Documentation	27
6.9.2.1	paint()	27
6.10	hugo_scene Class Reference	28
6.11	hugo_voyant_warning Class Reference	30
6.11.1	Constructor & Destructor Documentation	31
6.11.1.1	hugo_voyant_warning()	31
6.11.2	Member Function Documentation	31
6.11.2.1	paint()	31
6.12	hugo_voyants_clignotant Class Reference	32
6.12.1	Constructor & Destructor Documentation	33
6.12.1.1	hugo_voyants_clignotant()	33
6.12.2	Member Function Documentation	33
6.12.2.1	paint()	33
6.13	hugo_voyants_simples Class Reference	34
6.13.1	Constructor & Destructor Documentation	35
6.13.1.1	hugo_voyants_simples()	35
6.13.2	Member Function Documentation	35
6.13.2.1	paint()	35
6.14	jaugeBatterieHenri Class Reference	36
6.15	jaugeClignotantHenri Class Reference	37
6.16	jaugeEssenceHenri Class Reference	38
6.17	jaugeTemperatureHenri Class Reference	40
6.18	jaugeToursMinuteHenri Class Reference	41

6.19	jaugeVirtual Class Reference	42
6.20	Jonas_compteur Class Reference	43
6.21	Jonas_scene Class Reference	44
6.22	jonas_voyant_simple Class Reference	46
6.23	MainWindow Class Reference	47
6.23.1	Constructor & Destructor Documentation	48
6.23.1.1	MainWindow()	48
6.24	objet_virtuel Class Reference	49
6.24.1	Constructor & Destructor Documentation	50
6.24.1.1	objet_virtuel()	50
6.24.2	Member Function Documentation	51
6.24.2.1	getValue()	51
6.24.2.2	getValueMax()	51
6.24.2.3	setValue()	51
6.25	scene_globale Class Reference	52
6.25.1	Constructor & Destructor Documentation	53
6.25.1.1	scene_globale()	53
6.26	sceneDeFond Class Reference	53
6.27	sceneDeFondHenri Class Reference	54
6.28	SceneFlorian Class Reference	55
6.29	sceneGlobale Class Reference	57
6.30	Voyant Class Reference	58
6.31	warninghenri Class Reference	59
7	File Documentation	61
7.1	serveur/Hugo/hugo_compteur.cpp File Reference	61
7.1.1	Detailed Description	61
7.2	serveur/Hugo/hugo_ecran.cpp File Reference	62
7.2.1	Detailed Description	62
7.3	serveur/Hugo/hugo_scene.cpp File Reference	62
7.3.1	Detailed Description	63
7.4	serveur/Hugo/hugo_voyant_warning.cpp File Reference	63
7.4.1	Detailed Description	63
7.5	serveur/Hugo/hugo_voyants_simples.cpp File Reference	64
7.5.1	Detailed Description	64

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:

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 ou droit	CANN TURN X	1=clignotant droit, -1=clignotant gauche, 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 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
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	
cadrantVirtuel	15
jaugeVirtuel	42
objet_virtuel	49
afficheKmHenri	11
CadrantFlorian	13
cadrantHenri	14
clignotant	16
hugo_Compteur	20
hugo_ecran	23
hugo_MyGraphicsitem	25
hugo_voyant_warning	30
hugo_voyants_clignotant	32
hugo_voyants_simples	34
jaugeBatterieHenri	36
jaugeClignotantHenri	37
jaugeEssenceHenri	38
jaugeTemperatureHenri	40
jaugeToursMinuteHenri	41
Jonas_compteur	43
jonas_voyant_simple	46
Voyant	58
warninghenri	59
sceneDeFond	53
sceneDeFondHenri	54
QGraphicsScene	
scene_globale	52
henri_scene	18
hugo_scene	28
Jonas_scene	44
SceneFlorian	55
sceneGlobale	57
QMainWindow	
MainWindow	47
MainWindow	47
MainWindow	47

Chapter 4

Class Index

4.1 Class List

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

afficheKmHenri	11
CadrantFlorian	13
cadrantHenri	14
cadrantVirtuel	15
clignotant	16
henri_scene	18
hugo_Compteur	20
hugo_ecran	23
hugo_MyGraphicsitem	25
hugo_scene	28
hugo_voyant_warning	30
hugo_voyants_clignotant	32
hugo_voyants_simples	34
jaugeBatterieHenri	36
jaugeClignotantHenri	37
jaugeEssenceHenri	38
jaugeTemperatureHenri	40
jaugeToursMinuteHenri	41
jaugeVirtuel	42
Jonas_compteur	43
Jonas_scene	44
jonas_voyant_simple	46
MainWindow	47
objet_virtuel	49
scene_globale	52
sceneDeFond	53
sceneDeFondHenri	54
SceneFlorian	55
sceneGlobale	57
Voyant	58
warninghenri	59

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/clignotant.h	??
serveur/Florian/jaugevirtuel.h	??
serveur/Florian/mainwindow.h	??
serveur/Florian/scenedefond.h	??
serveur/Florian/sceneflorian.h	??
serveur/Florian/sceneglobale.h	??
serveur/Florian/voyant.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_compteur.cpp	
Classe compteur, permettant la création de compteurs paramétrables avec aiguille et effet de verre	61
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	62
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	62

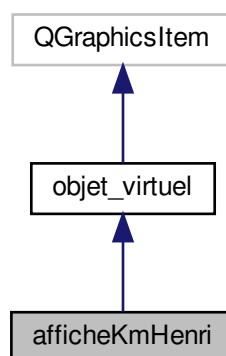
serveur/Hugo/ hugo_scene.h	??
serveur/Hugo/ hugo_voyant_warning.cpp	
Classe dérivée de objet_virtuel permettant l'affichage d'un voyant warning clignotant	63
serveur/Hugo/ hugo_voyant_warning.h	??
serveur/Hugo/ hugo_voyants_clignotant.h	??
serveur/Hugo/ hugo_voyants_simples.cpp	
Classe permettant l'affichage de voyants simples i.e. dont l'état est allumé ou éteint	64
serveur/Hugo/ hugo_voyants_simples.h	??
serveur/Jonas/ jonas_compteur.h	??
serveur/Jonas/ jonas_scene.h	??
serveur/Jonas/ jonas_voyant_simple.h	??

Chapter 6

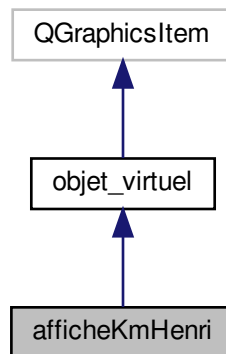
Class Documentation

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)

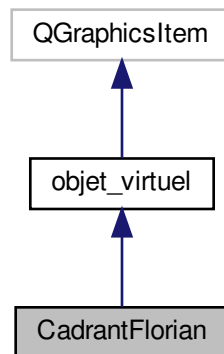
Additional Inherited Members

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

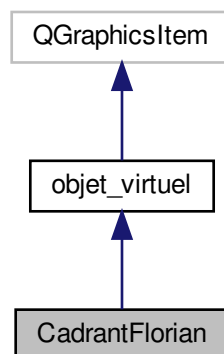
- serveur/Henri/affichekmhenri.h
- serveur/Henri/affichekmhenri.cpp

6.2 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** ()

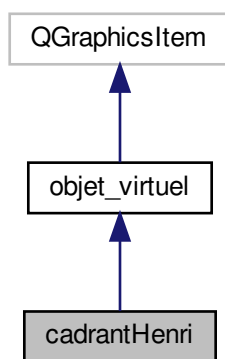
Additional Inherited Members

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

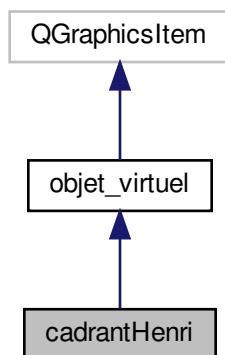
- serveur/Florian/cadrantflorian.h
- serveur/Florian/cadrantflorian.cpp

6.3 `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**

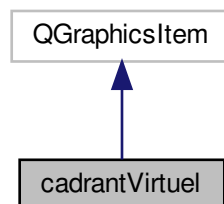
Additional Inherited Members

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

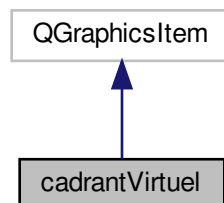
- serveur/Henri/cadranthenri.h
- serveur/Henri/cadranthenri.cpp

6.4 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

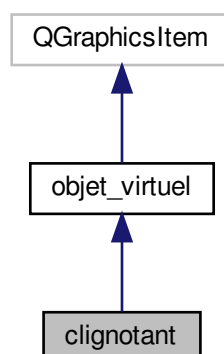
- int **valeur**
- int **valeurMax**

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

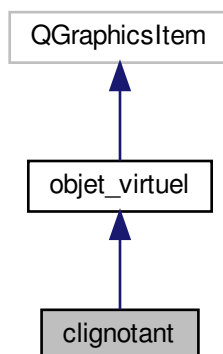
- serveur/Florian/cadrantvirtuel.h
- serveur/Florian/cadrantvirtuel.cpp

6.5 clignotant Class Reference

Inheritance diagram for clignotant:



Collaboration diagram for clignotant:



Public Member Functions

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

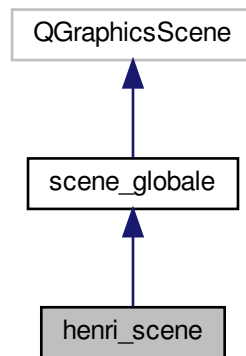
Additional Inherited Members

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

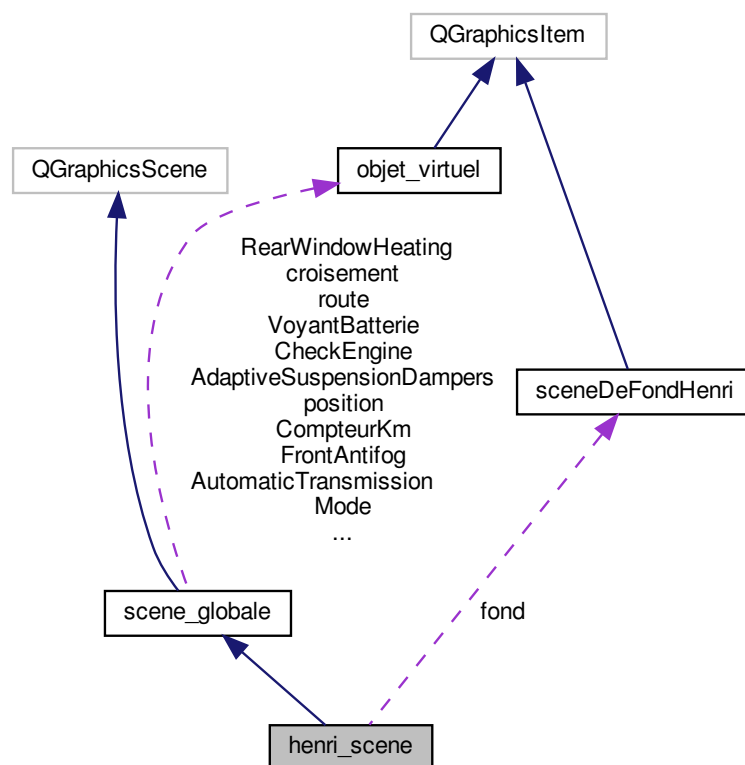
- `serveur/Florian/clignotant.h`
- `serveur/Florian/clignotant.cpp`

6.6 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

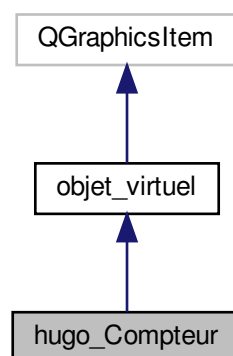
- [sceneDeFondHenri](#) * **fond**

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

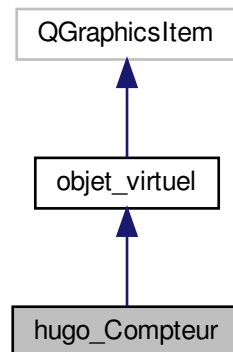
- serveur/Henri/henri_scene.h
- serveur/Henri/henri_scene.cpp

6.7 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

6.7.1 Constructor & Destructor Documentation

6.7.1.1 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>	

6.7.2 Member Function Documentation

6.7.2.1 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

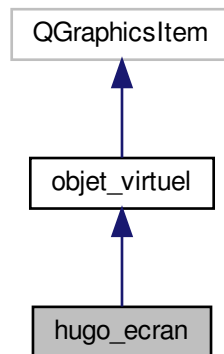
L'effet de verre est obtenu en superposant un cercle semi-transparent au compteur. L'option `direction_grad` permet de définir le sens du gradient.

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

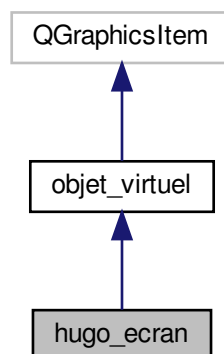
- `serveur/Hugo/hugo_compteur.h`
- `serveur/Hugo/hugo_compteur.cpp`

6.8 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

6.8.1 Constructor & Destructor Documentation

6.8.1.1 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.

6.8.2 Member Function Documentation

6.8.2.1 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

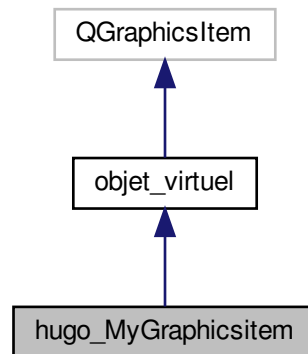
Affichage des différents textes. Le kilométrage est récupéré grâce à la fonction getValue.

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

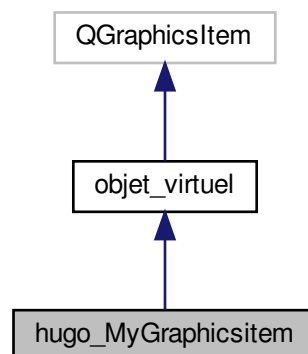
- serveur/Hugo/hugo_ecran.h
- serveur/Hugo/[hugo_ecran.cpp](#)

6.9 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

6.9.1 Constructor & Destructor Documentation

6.9.1.1 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.

6.9.2 Member Function Documentation

6.9.2.1 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 régime moteur

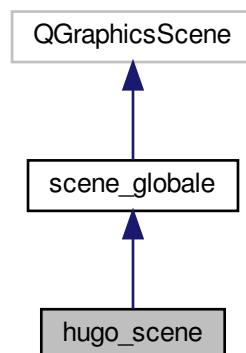
Affichage d'une icone jauge d'essence

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

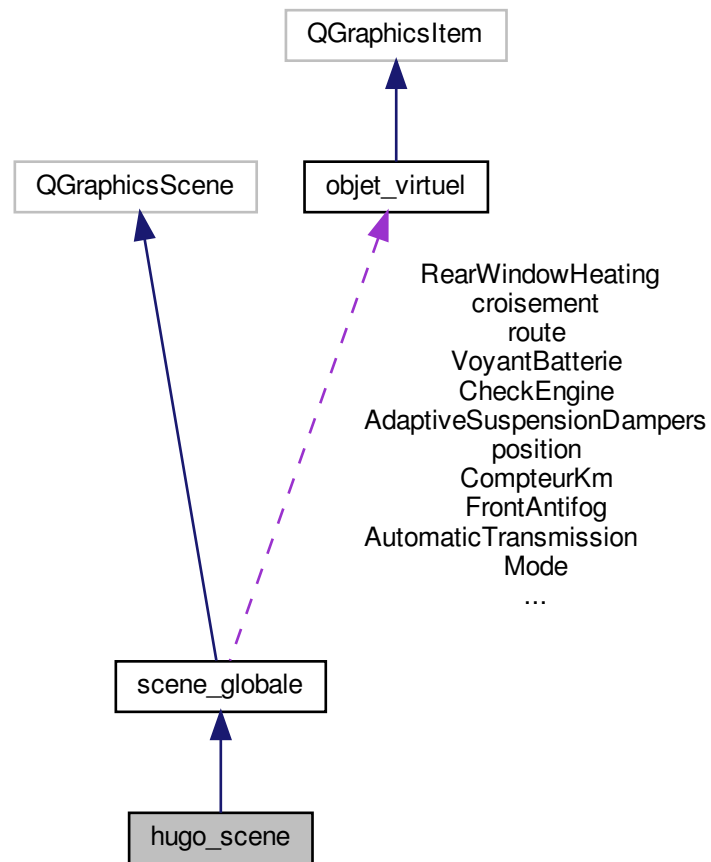
- serveur/Hugo/hugo_mygraphicsitem.h
- serveur/Hugo/hugo_mygraphicsitem.cpp

6.10 hugo_scene Class Reference

Inheritance diagram for hugo_scene:



Collaboration diagram for hugo_scene:



Public Member Functions

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

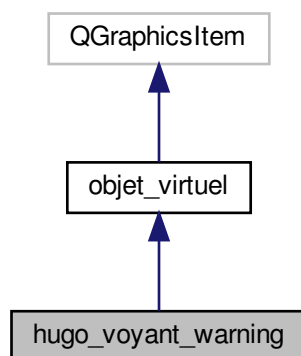
Additional Inherited Members

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

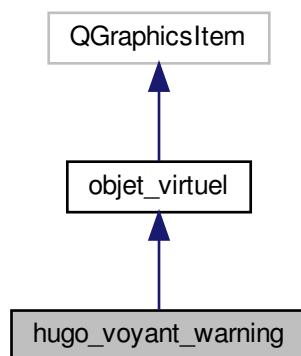
- serveur/Hugo/hugo_scene.h
- serveur/Hugo/[hugo_scene.cpp](#)

6.11 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

6.11.1 Constructor & Destructor Documentation

6.11.1.1 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.

6.11.2 Member Function Documentation

6.11.2.1 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.

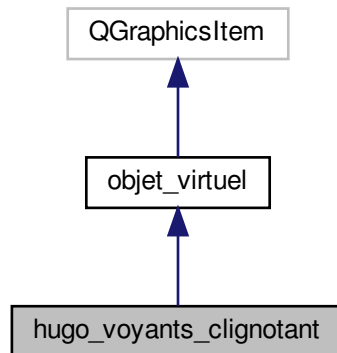
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné.

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

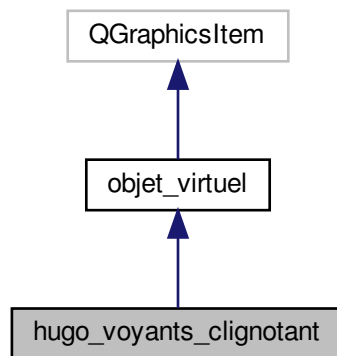
- serveur/Hugo/hugo_voyant_warning.h
- serveur/Hugo/[hugo_voyant_warning.cpp](#)

6.12 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

6.12.1 Constructor & Destructor Documentation

6.12.1.1 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.

6.12.2 Member Function Documentation

6.12.2.1 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.

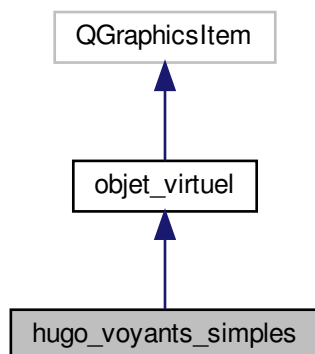
La mise à jour est effectuée à l'aide d'un timer, permettant un affichage alterné

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

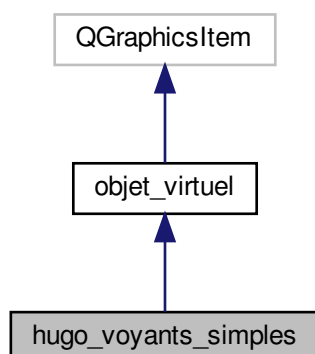
- serveur/Hugo/hugo_voyants_clignotant.h
- serveur/Hugo/hugo_voyants_clignotant.cpp

6.13 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

6.13.1 Constructor & Destructor Documentation

6.13.1.1 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

6.13.2 Member Function Documentation

6.13.2.1 paint()

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



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

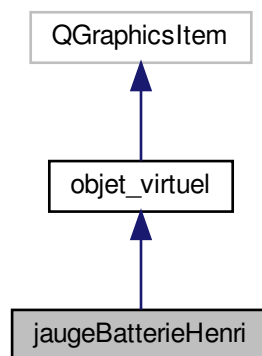
Va chercher la valeur de la variable "valeur" de la classe mère. Si ==1, le voyant doit être affiché.

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

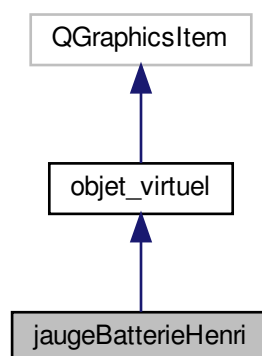
- serveur/Hugo/hugo_voyants_simples.h
- serveur/Hugo/[hugo_voyants_simples.cpp](#)

6.14 `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

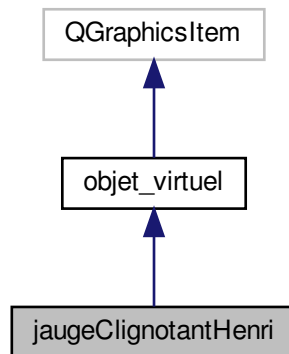
Additional Inherited Members

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

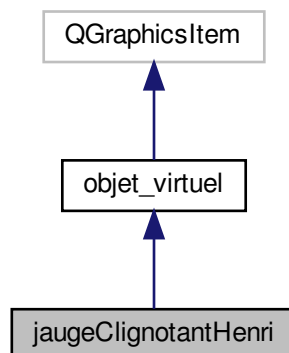
- serveur/Henri/jaugebatteriehenri.h
- serveur/Henri/jaugebatteriehenri.cpp

6.15 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**

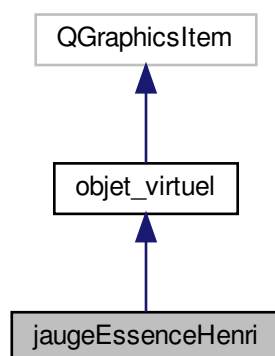
Additional Inherited Members

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

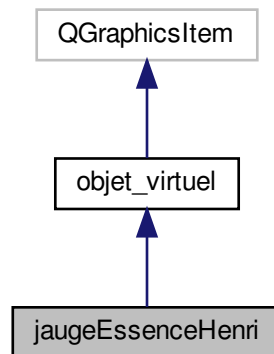
- serveur/Henri/jaugeclignotanthenri.h
- serveur/Henri/jaugeclignotanthenri.cpp

6.16 `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)

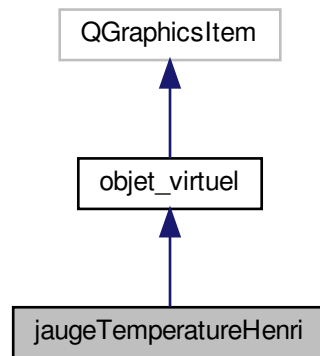
Additional Inherited Members

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

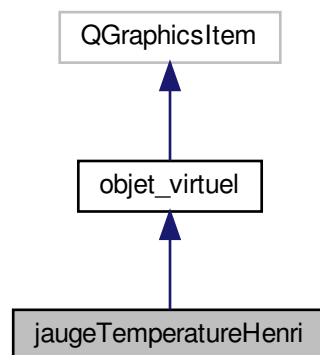
- `serveur/Henri/jaugeessencehenri.h`
- `serveur/Henri/jaugeessencehenri.cpp`

6.17 `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)`

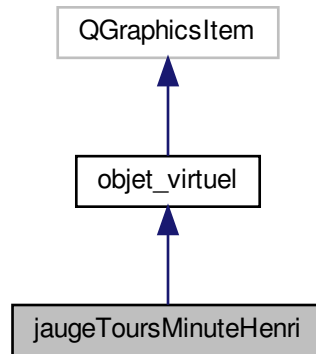
Additional Inherited Members

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

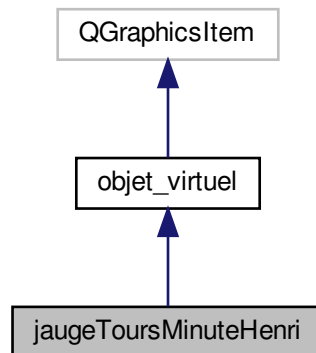
- `serveur/Henri/jaugetemperaturehenri.h`
- `serveur/Henri/jaugetemperaturehenri.cpp`

6.18 `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`

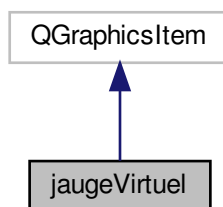
Additional Inherited Members

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

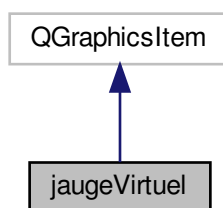
- serveur/Henri/jaugetoursminutehenri.h
- serveur/Henri/jaugetoursminutehenri.cpp

6.19 **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

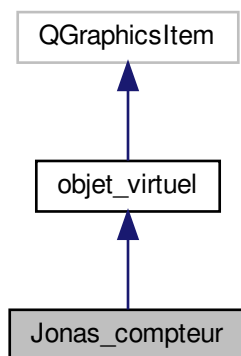
- int **valeur**
- int **valeurMax**

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

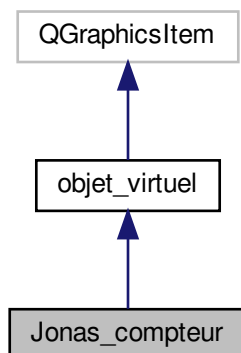
- serveur/Florian/jaugevirtuel.h
- serveur/Florian/jaugevirtuel.cpp

6.20 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)

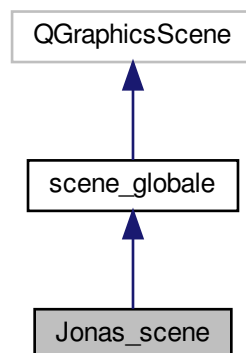
Additional Inherited Members

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

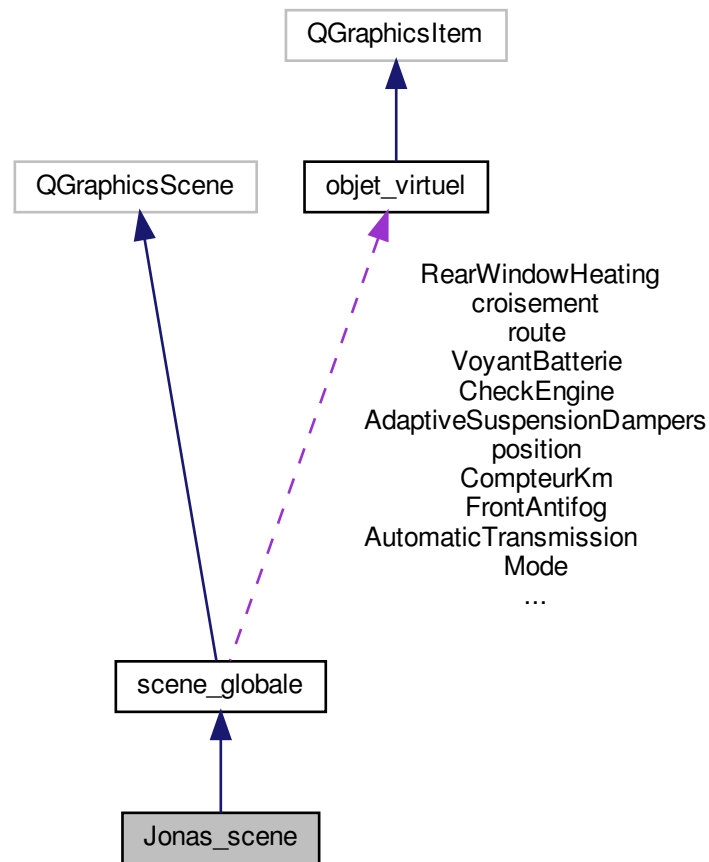
- serveur/Jonas/jonas_compteur.h
- serveur/Jonas/jonas_compteur.cpp

6.21 Jonas_scene Class Reference

Inheritance diagram for Jonas_scene:



Collaboration diagram for Jonas_scene:



Public Member Functions

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

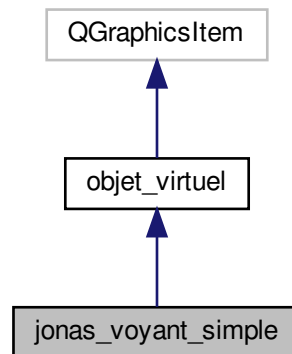
Additional Inherited Members

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

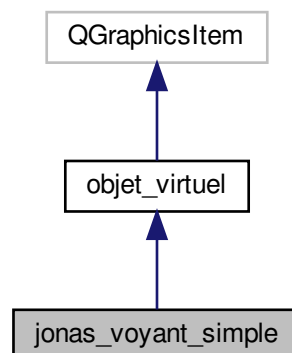
- serveur/Jonas/jonas_scene.h
- serveur/Jonas/jonas_scene.cpp

6.22 jonas_voyant_simple Class Reference

Inheritance diagram for jonas_voyant_simple:



Collaboration diagram for jonas_voyant_simple:



Public Member Functions

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

- **MainWindow** (QWidget *parent=0)
- void **cli** ()
- **MainWindow** (QWidget *parent=0)

Public Attributes

- [SceneFlorian](#) * **scene**
- [sceneGlobale](#) * **scene**

6.23.1 Constructor & Destructor Documentation

6.23.1.1 MainWindow()

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

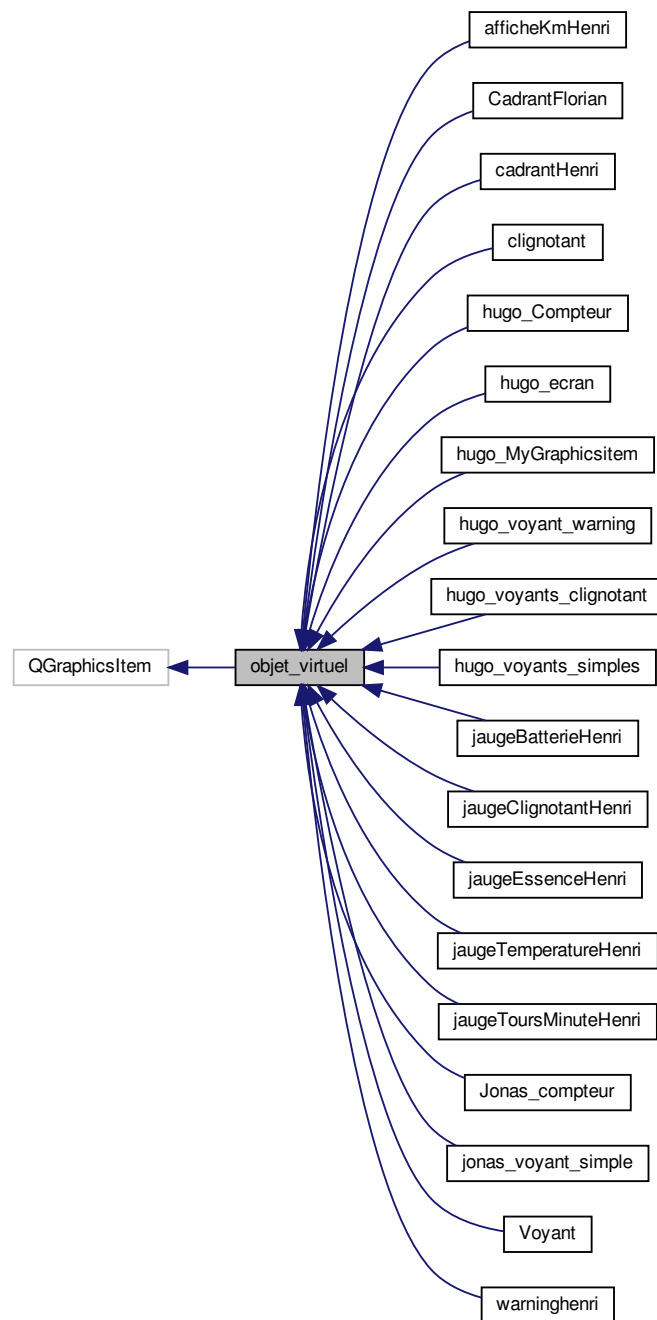
La scène par défaut est

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

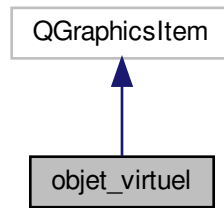
- serveur/Florian/mainwindow.h
- serveur/Florian/mainwindow.cpp

6.24 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**

6.24.1 Constructor & Destructor Documentation

6.24.1.1 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

6.24.2 Member Function Documentation

6.24.2.1 `getValue()`

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

Fonction renvoyant la variable value.

Returns

6.24.2.2 `getValueMax()`

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

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

Returns

6.24.2.3 `setValue()`

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

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

Parameters

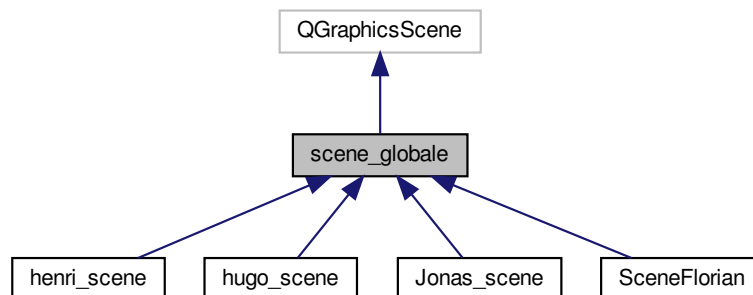
<i>set_value</i>	valeur à attribuer à value
------------------	----------------------------

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

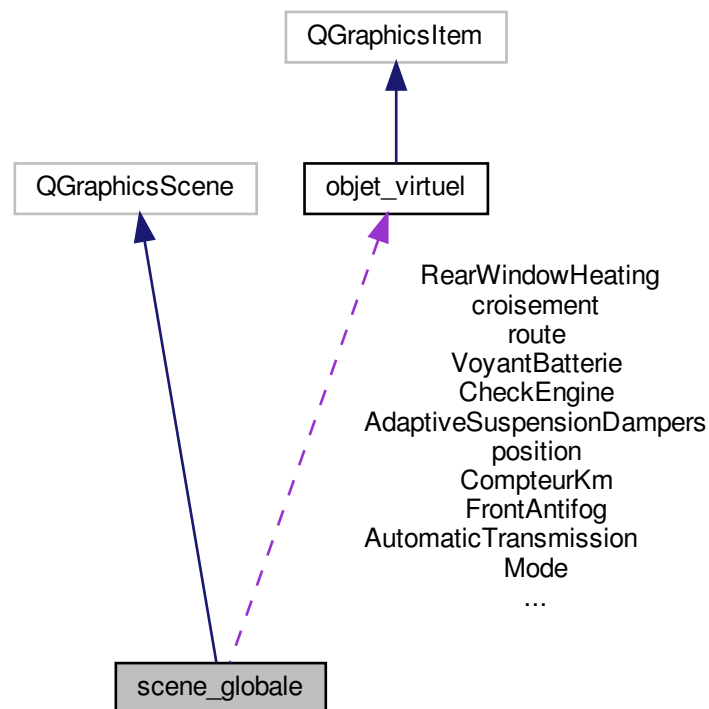
- serveur/objet_virtuel.h
- serveur/objet_virtuel.cpp

6.25 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**

6.25.1 Constructor & Destructor Documentation

6.25.1.1 `scene_globale()`

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

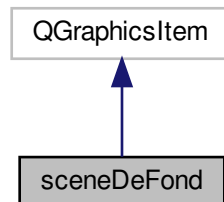
Classe dérivant de QGraphicsScene. Va servir de classe mère pour toutes les scènes, ce qui permettra de passer d'une scène à l'autre de façon dynamique. Tous les objets utilisés dans le dashboard seront définis dans "scène↔_globale.h"

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

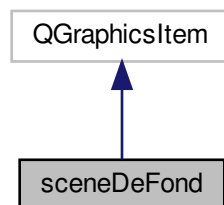
- serveur/scene_globale.h
- serveur/scene_globale.cpp

6.26 sceneDeFond Class Reference

Inheritance diagram for sceneDeFond:



Collaboration diagram for sceneDeFond:



Public Member Functions

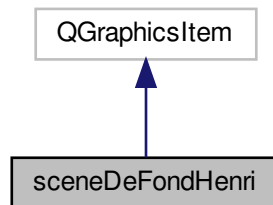
- **sceneDeFond** (QGraphicsItem *parent=nullptr)

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

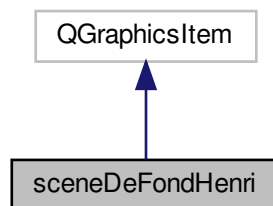
- serveur/Florian/scenedefond.h
- serveur/Florian/scenedefond.cpp

6.27 sceneDeFondHenri Class Reference

Inheritance diagram for sceneDeFondHenri:



Collaboration diagram for sceneDeFondHenri:



Public Member Functions

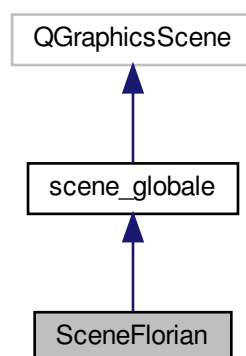
- **sceneDeFondHenri** (`QGraphicsItem *parent=nullptr`)
- void **paint** (`QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget`)
- `QRectF` **boundingRect** () const

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

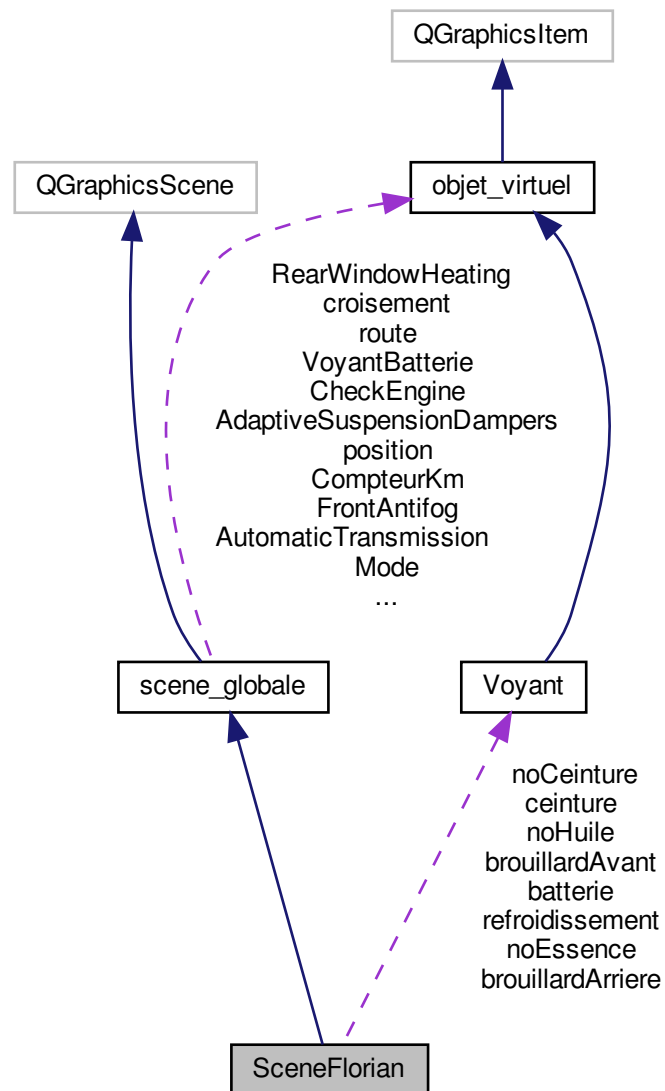
- `serveur/Henri/scenedefondhenri.h`
- `serveur/Henri/scenedefondhenri.cpp`

6.28 SceneFlorian Class Reference

Inheritance diagram for SceneFlorian:



Collaboration diagram for SceneFlorian:



Public Member Functions

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

Public Attributes

- **Voyant** * **brouillardAvant**
- **Voyant** * **brouillardArriere**
- **Voyant** * **noEssence**
- **Voyant** * **noHuile**

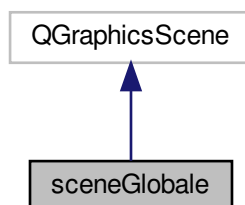
- [Voyant](#) * refroidissement
- [Voyant](#) * batterie
- [Voyant](#) * ceinture
- [Voyant](#) * noCeinture

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

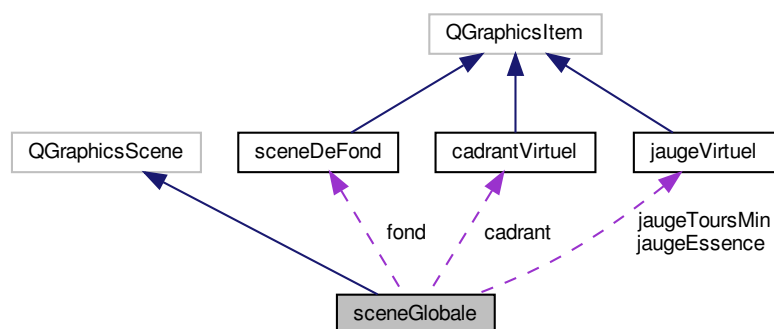
- serveur/Florian/sceneflorian.h
- serveur/Florian/sceneflorian.cpp

6.29 sceneGlobale Class Reference

Inheritance diagram for sceneGlobale:



Collaboration diagram for sceneGlobale:



Public Member Functions

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

Public Attributes

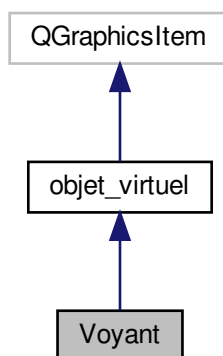
- [sceneDeFond](#) * **fond**
- [cadrantVirtuel](#) * **cadrant**
- [jaugeVirtuel](#) * **jaugeEssence**
- [jaugeVirtuel](#) * **jaugeToursMin**

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

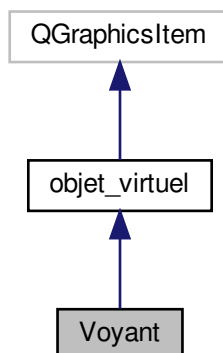
- serveur/Florian/sceneglobale.h
- serveur/Florian/sceneglobale.cpp

6.30 Voyant Class Reference

Inheritance diagram for Voyant:



Collaboration diagram for Voyant:



Public Member Functions

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

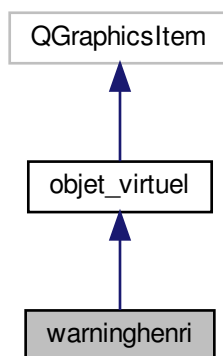
Additional Inherited Members

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

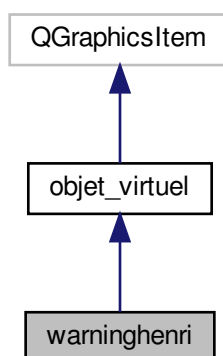
- serveur/Florian/voyant.h
- serveur/Florian/voyant.cpp

6.31 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

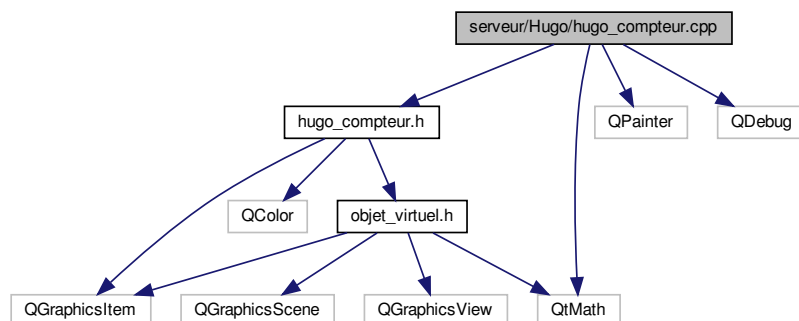
File Documentation

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.

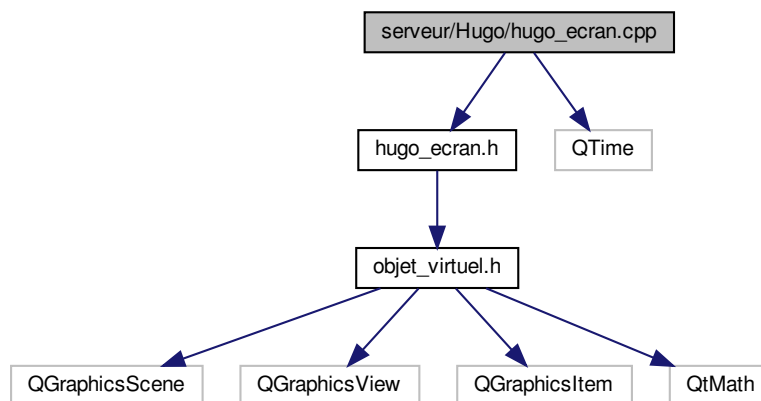
Classe héritée de `objet_virtuel`. Utilisée pour l'affichage du compteur de vitesse, du compteur rpm, de la jauge d'essence ainsi que de la jauge de température. L'appel au constructeur permet d'initialiser la totalité des paramètres, et la fonction `paint` permet l'affichage.

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.

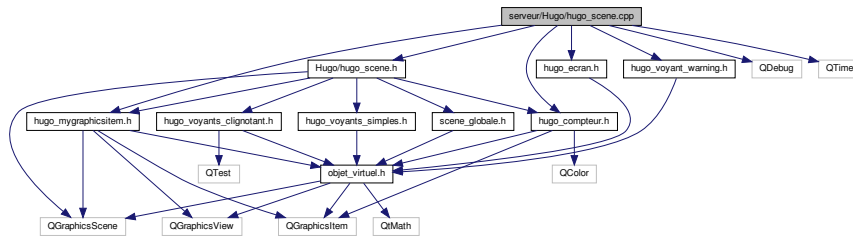
7.3 serveur/Hugo/hugo_scene.cpp File Reference

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 <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](#).

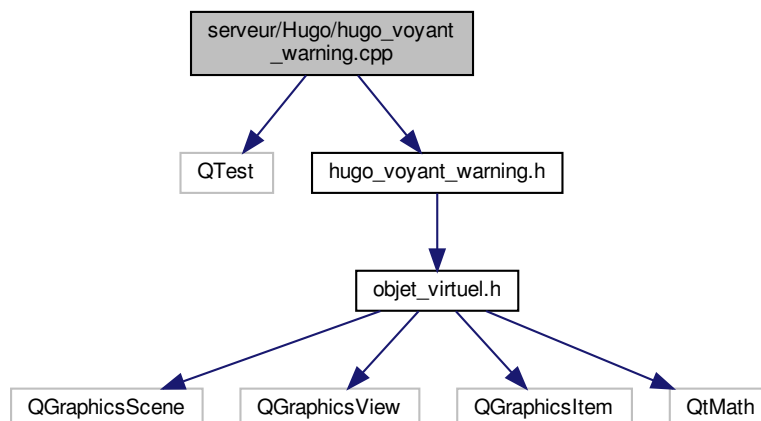
7.4 serveur/Hugo/hugo_voyant_warning.cpp File Reference

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

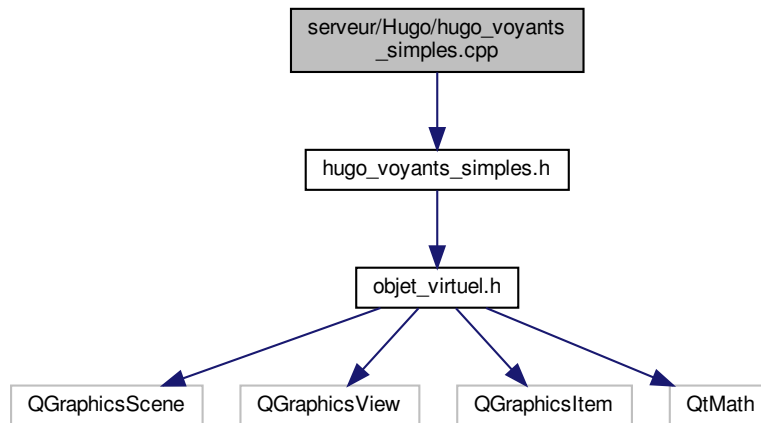
Classe dérivée de [objet_virtuel](#) permettant l'affichage d'un voyant warning clignotant.

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.