

My Project

Generated by Doxygen 1.8.13

Contents

1	README	1
2	Hierarchical Index	5
2.1	Class Hierarchy	5
3	Class Index	7
3.1	Class List	7
4	Class Documentation	9
4.1	afficheKmHenri Class Reference	9
4.2	CadrantFlorian Class Reference	9
4.3	cadrantHenri Class Reference	9
4.4	cadrantVirtuel Class Reference	10
4.5	clignotant Class Reference	10
4.6	henri_scene Class Reference	11
4.7	hugo_Compteur Class Reference	11
4.7.1	Constructor & Destructor Documentation	12
4.7.1.1	hugo_Compteur()	12
4.7.2	Member Function Documentation	13
4.7.2.1	paint()	13
4.8	hugo_ecran Class Reference	14
4.8.1	Member Function Documentation	14
4.8.1.1	paint()	14
4.9	hugo_MyGraphicsitem Class Reference	14
4.9.1	Member Function Documentation	15

4.9.1.1	paint()	15
4.10	hugo_scene Class Reference	15
4.10.1	Constructor & Destructor Documentation	15
4.10.1.1	hugo_scene()	16
4.11	hugo_voyant_warning Class Reference	16
4.11.1	Member Function Documentation	16
4.11.1.1	paint()	17
4.12	hugo_voyants_clignotant Class Reference	17
4.12.1	Constructor & Destructor Documentation	17
4.12.1.1	hugo_voyants_clignotant()	18
4.12.2	Member Function Documentation	18
4.12.2.1	paint()	18
4.13	hugo_voyants_simples Class Reference	18
4.13.1	Constructor & Destructor Documentation	19
4.13.1.1	hugo_voyants_simples()	19
4.13.2	Member Function Documentation	19
4.13.2.1	paint()	19
4.14	jaugBatterieHenri Class Reference	20
4.15	jaugClignotantHenri Class Reference	20
4.16	jaugEssenceHenri Class Reference	21
4.17	jaugTemperatureHenri Class Reference	21
4.18	jaugToursMinuteHenri Class Reference	22
4.19	jaugVirtuel Class Reference	22
4.20	Jonas_compteur Class Reference	23
4.21	Jonas_scene Class Reference	23
4.22	jonas_voyant_simple Class Reference	24
4.23	MainWindow Class Reference	24
4.23.1	Constructor & Destructor Documentation	24
4.23.1.1	MainWindow()	25
4.24	objet_virtuel Class Reference	25

4.24.1	Constructor & Destructor Documentation	25
4.24.1.1	objet_virtuel()	25
4.24.2	Member Function Documentation	26
4.24.2.1	getValue()	26
4.24.2.2	getValueMax()	26
4.24.2.3	setValue()	26
4.25	scene_globale Class Reference	26
4.25.1	Constructor & Destructor Documentation	27
4.25.1.1	scene_globale()	27
4.26	sceneDeFond Class Reference	27
4.27	sceneDeFondHenri Class Reference	28
4.28	SceneFlorian Class Reference	28
4.29	sceneGlobale Class Reference	29
4.30	Voyant Class Reference	29
4.31	warninghenri Class Reference	29

Chapter 1

README

#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:

- Déclarez votre objet en tant que `objet_virtuel` dans "scene_globale.h". Il sera ensuite défini de la façon qu'il vous plaira dans votre `prénom_scene`.
- Dans le fichier "main.c" du client, dans la partie concernant la commande HELP, ajoutez votre message CANN, ainsi qu'une brève description de ce dernier.
- Dans la fonction `MainWindow::reception()` de "mainwindow.cpp", rajoutez une condition correspondant à votre message CANN, ainsi que l'algorithme fonctionnel. Ce dernier ne doit utiliser que les fonctions `getValue`, `setValue` et `getValueMax` définies dans la classe `objet_virtuel`. N'hésitez pas à rajouter des messages d'erreur en cas de variables non compatibles en provenance du client.
- Dans le fichier "README.md", dans la partie Liste des messages CANN, rajoutez une descriptif de votre fonctionnalité.

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

Bugs répertoriés

Listez ici vos suggestions ou bugs aperçus lors de l'utilisation du programme.

- Lorsque Warning est activé, le message CANN TURN 0 désactive les clignotants mais laisse le voyant warning allumé.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

QGraphicsItem	
cadrantVirtuel	10
jaugeVirtuel	22
objet_virtuel	25
afficheKmHenri	9
CadrantFlorian	9
cadrantHenri	9
clignotant	10
hugo_Compteur	11
hugo_ecran	14
hugo_MyGraphicsitem	14
hugo_voyant_warning	16
hugo_voyants_clignotant	17
hugo_voyants_simples	18
jaugeBatterieHenri	20
jaugeClignotantHenri	20
jaugeEssenceHenri	21
jaugeTemperatureHenri	21
jaugeToursMinuteHenri	22
Jonas_compteur	23
jonas_voyant_simple	24
Voyant	29
warninghenri	29
sceneDeFond	27
sceneDeFondHenri	28
QGraphicsScene	
scene_globale	26
henri_scene	11
hugo_scene	15
Jonas_scene	23
SceneFlorian	28
sceneGlobale	29
QMainWindow	
MainWindow	24
MainWindow	24
MainWindow	24

Chapter 3

Class Index

3.1 Class List

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

afficheKmHenri	9
CadrantFlorian	9
cadrantHenri	9
cadrantVirtuel	10
clignotant	10
henri_scene	11
hugo_Compteur	11
hugo_ecran	14
hugo_MyGraphicsitem	14
hugo_scene	15
hugo_voyant_warning	16
hugo_voyants_clignotant	17
hugo_voyants_simples	18
jaugueBatterieHenri	20
jaugueClignotantHenri	20
jaugueEssenceHenri	21
jaugueTemperatureHenri	21
jaugueToursMinuteHenri	22
jaugueVirtuel	22
Jonas_compteur	23
Jonas_scene	23
jonas_voyant_simple	24
MainWindow	24
objet_virtuel	25
scene_globale	26
sceneDeFond	27
sceneDeFondHenri	28
SceneFlorian	28
sceneGlobale	29
Voyant	29
warninghenri	29

Chapter 4

Class Documentation

4.1 afficheKmHenri Class Reference

Inheritance diagram for afficheKmHenri:

4.2 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:

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

4.3 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:

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

4.4 **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:

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

4.5 **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:

- Florian/clignotant.h
- Florian/clignotant.cpp

4.6 `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:

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

4.7 `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)

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

4.7.1 Constructor & Destructor Documentation

4.7.1.1 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>	

4.7.2 Member Function Documentation

4.7.2.1 paint()

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

<Mise en place Antialiasing//

La première étape est la création de l'arc de cercle servant de support au cadran. L'option critique permet d'ajouter une autre couleur à une partie du cadran

Création des graduations en deux boucles (grandes et petites graduations)

Ajout du texte sur les graduations

Mise en place de l'aiguille

<Cercle au centre de l'aiguille//

<Dégradé à la base de l'aiguille pour donner une impression de relief//

Création d'un polygone pour représenter l'aiguille, et affichage

Affichage d'un effet verre en superposant un cercle transparent avec gradient blanc->noir sur le cadran//

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

- Hugo/hugo_compteur.h
- Hugo/hugo_compteur.cpp

4.8 hugo_ecran Class Reference

Inheritance diagram for hugo_ecran:

Collaboration diagram for hugo_ecran:

Public Member Functions

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

Additional Inherited Members

4.8.1 Member Function Documentation

4.8.1.1 paint()

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

< Conversion pour avoir l'affichage sous forme heure.minutes

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

- Hugo/hugo_ecran.h
- Hugo/hugo_ecran.cpp

4.9 hugo_MyGraphicsitem Class Reference

Inheritance diagram for hugo_MyGraphicsitem:

Collaboration diagram for hugo_MyGraphicsitem:

Public Member Functions

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

Public Attributes

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

Additional Inherited Members

4.9.1 Member Function Documentation

4.9.1.1 paint()

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

<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 if ([getValue\(\)](#)==1){///< 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:

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

4.10 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

4.10.1 Constructor & Destructor Documentation

4.10.1.1 hugo_scene()

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

Création de la scène de fond

Déclaration et paramétrage de tous les compteurs

Déclaration et paramétrages de tous les voyants simples

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

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

4.11 hugo_voyant_warning Class Reference

Inheritance diagram for hugo_voyant_warning:

Collaboration diagram for hugo_voyant_warning:

Public Member Functions

- 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

4.11.1 Member Function Documentation

4.11.1.1 paint()

```
void hugo_voyant_warning::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:

- Hugo/hugo_voyant_warning.h
- Hugo/hugo_voyant_warning.cpp

4.12 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. La variable cligno sert à gérer l'affichage alterné
- 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

4.12.1 Constructor & Destructor Documentation

4.12.1.1 hugo_voyants_clignotant()

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

[hugo_voyants_clignotant::hugo_voyants_clignotant](#). La variable cligno sert à gérer l'affichage alterné

Classe permettant d'afficher les voyants clignotants

4.12.2 Member Function Documentation

4.12.2.1 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:

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

4.13 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

4.13.1 Constructor & Destructor Documentation

4.13.1.1 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

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

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

4.13.2 Member Function Documentation

4.13.2.1 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:

- Hugo/hugo_voyants_simples.h
- Hugo/hugo_voyants_simples.cpp

4.14 `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:

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

4.15 `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:

- `Henri/jaugeclignotanthenri.h`
- `Henri/jaugeclignotanthenri.cpp`

4.16 `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:

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

4.17 `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:

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

4.18 `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:

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

4.19 `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:

- `Florian/jaugevirtuel.h`
- `Florian/jaugevirtuel.cpp`

4.20 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:

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

4.21 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:

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

4.22 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:

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

4.23 MainWindow Class Reference

Inheritance diagram for MainWindow:

Collaboration diagram for MainWindow:

Public Member Functions

- [MainWindow](#) (QWidget *parent=0)
- **MainWindow** (QWidget *parent=0)
- void **cli** ()
- **MainWindow** (QWidget *parent=0)

Public Attributes

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

4.23.1 Constructor & Destructor Documentation

4.23.1.1 MainWindow()

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

La scène par défaut est

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

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

4.24 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**

4.24.1 Constructor & Destructor Documentation

4.24.1.1 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

4.24.2 Member Function Documentation

4.24.2.1 `getValue()`

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

Fonction renvoyant la variable value.

Returns

4.24.2.2 `getValueMax()`

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

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

Returns

4.24.2.3 `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:

- `objet_virtuel.h`
- `objet_virtuel.cpp`

4.25 `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**

4.25.1 Constructor & Destructor Documentation

4.25.1.1 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:

- scene_globale.h
- scene_globale.cpp

4.26 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:

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

4.27 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:

- Henri/scenedefondhenri.h
- Henri/scenedefondhenri.cpp

4.28 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:

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

4.29 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:

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

4.30 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:

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

4.31 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:

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