

# Protocole

Généré par Doxygen 1.8.6

Lundi 21 Juillet 2014 15 :31 :25



# Table des matières

<b>1</b>	<b>Index hiérarchique</b>	<b>1</b>
1.1	Hiérarchie des classes . . . . .	1
<b>2</b>	<b>Index des classes</b>	<b>3</b>
2.1	Liste des classes . . . . .	3
<b>3</b>	<b>Index des fichiers</b>	<b>5</b>
3.1	Liste des fichiers . . . . .	5
<b>4</b>	<b>Documentation des classes</b>	<b>7</b>
4.1	Référence de la classe CentralWidget . . . . .	7
4.1.1	Description détaillée . . . . .	8
4.1.2	Documentation des constructeurs et destructeur . . . . .	8
4.1.2.1	CentralWidget . . . . .	8
4.1.2.2	~CentralWidget . . . . .	8
4.1.3	Documentation des fonctions membres . . . . .	8
4.1.3.1	cleanButton . . . . .	9
4.1.3.2	endLoop . . . . .	9
4.1.3.3	endTest . . . . .	9
4.1.3.4	endTraining . . . . .	9
4.1.3.5	nextTest . . . . .	9
4.1.3.6	saveChoice . . . . .	9
4.1.3.7	setStyleButton . . . . .	9
4.1.3.8	startTest . . . . .	9
4.1.3.9	startTraining . . . . .	10
4.1.4	Documentation des données membres . . . . .	10
4.1.4.1	m_buttons . . . . .	10
4.1.4.2	m_choices . . . . .	10
4.1.4.3	m_compareTimer . . . . .	10
4.1.4.4	m_debug . . . . .	10
4.1.4.5	m_info . . . . .	10
4.1.4.6	m_layout . . . . .	10

4.1.4.7	<a href="#">m_nextButton</a>	10
4.1.4.8	<a href="#">m_quitButton</a>	10
4.1.4.9	<a href="#">m_respondTime</a>	10
4.1.4.10	<a href="#">m_reStartTrainingButton</a>	10
4.1.4.11	<a href="#">m_signalMapperChoice</a>	11
4.1.4.12	<a href="#">m_startButton</a>	11
4.1.4.13	<a href="#">m_startTestButton</a>	11
4.1.4.14	<a href="#">m_startTrainingButton</a>	11
4.1.4.15	<a href="#">m_state</a>	11
4.1.4.16	<a href="#">m_time</a>	11
4.1.4.17	<a href="#">m_videosWidget</a>	11
4.2	<a href="#">Référence de la classe FullWindow</a>	11
4.2.1	<a href="#">Description détaillée</a>	12
4.2.2	<a href="#">Documentation des constructeurs et destructeur</a>	12
4.2.2.1	<a href="#">FullWindow</a>	12
4.2.2.2	<a href="#">~FullWindow</a>	12
4.2.3	<a href="#">Documentation des fonctions membres</a>	12
4.2.3.1	<a href="#">keyPressEvent</a>	12
4.2.3.2	<a href="#">resizeEvent</a>	12
4.2.4	<a href="#">Documentation des données membres</a>	13
4.2.4.1	<a href="#">m_centralWidget</a>	13
4.2.4.2	<a href="#">m_layout</a>	13
4.2.4.3	<a href="#">m_mainWidget</a>	13
4.3	<a href="#">Référence de la classe Menu</a>	13
4.3.1	<a href="#">Description détaillée</a>	14
4.3.2	<a href="#">Documentation des constructeurs et destructeur</a>	14
4.3.2.1	<a href="#">Menu</a>	14
4.3.2.2	<a href="#">~Menu</a>	14
4.3.3	<a href="#">Documentation des fonctions membres</a>	14
4.3.3.1	<a href="#">startTest</a>	14
4.3.4	<a href="#">Documentation des données membres</a>	14
4.3.4.1	<a href="#">m_fullWindow</a>	14
4.3.4.2	<a href="#">m_label</a>	14
4.3.4.3	<a href="#">m_layout</a>	14
4.3.4.4	<a href="#">m_startButton</a>	14
4.3.4.5	<a href="#">m_widget</a>	14
4.4	<a href="#">Référence de la classe Video</a>	15
4.4.1	<a href="#">Description détaillée</a>	15
4.4.2	<a href="#">Documentation des constructeurs et destructeur</a>	15
4.4.2.1	<a href="#">Video</a>	15

4.4.2.2	~Video	16
4.4.3	Documentation des fonctions membres	16
4.4.3.1	lpl2QImage	16
4.4.3.2	nextFrame	16
4.4.3.3	setVideo	16
4.4.4	Documentation des données membres	16
4.4.4.1	m_capture	16
4.4.4.2	m_image	16
4.4.4.3	m_label	17
4.4.4.4	m_layout	17
4.5	Référence de la classe VideosWidget	17
4.5.1	Description détaillée	18
4.5.2	Documentation des constructeurs et destructeur	18
4.5.2.1	VideosWidget	18
4.5.2.2	~VideosWidget	18
4.5.3	Documentation des fonctions membres	18
4.5.3.1	cleanVideos	18
4.5.3.2	exportResult	19
4.5.3.3	loadNext	20
4.5.3.4	loadSequence	20
4.5.3.5	organizeLayout	20
4.5.3.6	play	20
4.5.3.7	resizeLayout	20
4.5.3.8	runVideos	20
4.5.3.9	setFocus	20
4.5.3.10	showVideos	21
4.5.3.11	start	21
4.5.3.12	stop	21
4.5.4	Documentation des données membres	21
4.5.4.1	m_boucleTimer	21
4.5.4.2	m_cmpSequence	21
4.5.4.3	m_height	21
4.5.4.4	m_labelRef	21
4.5.4.5	m_sequence	21
4.5.4.6	m_sequencePath	21
4.5.4.7	m_timer	21
4.5.4.8	m_videoLayout	22
4.5.4.9	m_videos	22
4.5.4.10	m_width	22

<b>5</b>	<b>Documentation des fichiers</b>	<b>23</b>
5.1	Référence du fichier centralwidget.cpp	23
5.1.1	Description détaillée	23
5.2	Référence du fichier centralwidget.h	23
5.3	Référence du fichier define.h	24
5.3.1	Description détaillée	24
5.3.2	Documentation des macros	24
5.3.2.1	LAYOUT_HEIGHT	24
5.3.2.2	LAYOUT_R	24
5.3.2.3	LAYOUT_R_T	24
5.3.2.4	LAYOUT_SPACE	25
5.3.2.5	LAYOUT_V_SPACE	25
5.3.2.6	LAYOUT_VIDEOS_HEIGHT	25
5.3.2.7	LAYOUT_VIDEOS_HEIGHT_T	25
5.3.2.8	LAYOUT_VIDEOS_WIDTH	25
5.3.2.9	LAYOUT_VIDEOS_WIDTH_T	25
5.3.2.10	LAYOUT_WIDTH	25
5.3.2.11	NB_TEST	25
5.3.3	Documentation du type de l'énumération	25
5.3.3.1	State	25
5.4	Référence du fichier fullwindow.cpp	25
5.4.1	Description détaillée	26
5.5	Référence du fichier fullwindow.h	26
5.5.1	Description détaillée	26
5.5.2	Documentation des macros	26
5.5.2.1	WIDGET_HEIGHT	26
5.5.2.2	WIDGET_WIDTH	26
5.6	Référence du fichier main.cpp	26
5.6.1	Description détaillée	27
5.6.2	Documentation des fonctions	27
5.6.2.1	main	27
5.7	Référence du fichier menu.cpp	27
5.7.1	Description détaillée	27
5.8	Référence du fichier menu.h	27
5.8.1	Description détaillée	28
5.9	Référence du fichier video.cpp	28
5.9.1	Description détaillée	28
5.10	Référence du fichier video.h	28
5.10.1	Description détaillée	28
5.11	Référence du fichier videoswidget.cpp	29

---

5.12 Référence du fichier videoswidget.h . . . . .	29
<b>Index</b>	<b>30</b>





# Chapitre 1

## Index hiérarchique

### 1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

QMainWindow	
FullWindow . . . . .	11
Menu . . . . .	13
QWidget	
CentralWidget . . . . .	7
Video . . . . .	15
VideosWidget . . . . .	17



## Chapitre 2

# Index des classes

### 2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">CentralWidget</a>	
La classe <a href="#">CentralWidget</a> . . . . .	7
<a href="#">FullWindow</a>	
La classe <a href="#">FullWindow</a> . . . . .	11
<a href="#">Menu</a>	
La classe <a href="#">Menu</a> . . . . .	13
<a href="#">Video</a>	
La classe <a href="#">Video</a> . . . . .	15
<a href="#">VideosWidget</a>	
La classe <a href="#">VideosWidget</a> . . . . .	17



## Chapitre 3

# Index des fichiers

### 3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

<a href="#">centralwidget.cpp</a>		
	Méthodes de la classe <a href="#">CentralWidget</a>	23
<a href="#">centralwidget.h</a>		23
<a href="#">define.h</a>		
	Enumeration pour le déroulement du protocole et grandeurs globales pour le dimensionnement de l'interface	24
<a href="#">fullwindow.cpp</a>		
	Méthodes de la classe <a href="#">FullWindow</a>	25
<a href="#">fullwindow.h</a>		
	Fenêtre plein écran	26
<a href="#">main.cpp</a>		
	Interface de test subjectif	26
<a href="#">menu.cpp</a>		
	Méthodes de la classe <a href="#">Menu</a>	27
<a href="#">menu.h</a>		
	<a href="#">Menu</a> de base	27
<a href="#">video.cpp</a>		
	Méthodes de la classe <a href="#">Video</a>	28
<a href="#">video.h</a>		
	Widget d'affichage d'une video	28
<a href="#">videoswidget.cpp</a>		29
<a href="#">videoswidget.h</a>		29



## Chapitre 4

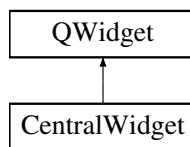
# Documentation des classes

### 4.1 Référence de la classe CentralWidget

La classe [CentralWidget](#).

```
#include <centralwidget.h>
```

Graphe d'héritage de CentralWidget :



#### Connecteurs publics

- void [saveChoice](#) (int choice)  
*Slot appelé lors de la selection d'une vidéo.*
- void [nextTest](#) ()  
*Slot appelé lors du lancement par l'utilisateur des vidéos pour leur comparaison.*
- void [startTraining](#) ()  
*Slot appelé par l'utilisateur pour démarrer la phase d'entraînement.*
- void [startTest](#) ()  
*Slot appelé par l'utilisateur pour démarrer le test reel.*
- void [endLoop](#) ()  
*Slot appelé par un timer pour arrêter les vidéos.*

#### Fonctions membres publiques

- [CentralWidget](#) (QWidget \*parent=0)  
*Constructeur [CentralWidget](#).*
- virtual [~CentralWidget](#) ()  
*Destructeur [CentralWidget](#).*
- void [endTraining](#) ()  
*Termine la phase d'entraînement.*
- void [endTest](#) ()  
*Termine le test.*
- void [setStyleButton](#) (QPushButton \*button, int i=0)  
*Applique un style predefinit à un bouton.*
- void [cleanButton](#) ()  
*Cache tout les boutons à l'écran.*

## Attributs privés

- `QGridLayout * m_layout`  
*Layout général.*
- `QVector< int > m_choices`  
*Tableau récupérant les choix de l'utilisateur.*
- `QVector< QPushButton * > m_buttons`  
*Les boutons de choix des vidéos.*
- `QSignalMapper * m_signalMapperChoice`  
*Pour mapper les bouton de choix.*
- `VideosWidget * m_videosWidget`  
*L'espace dédié au placement des vidéos.*
- `QTimer * m_compareTimer`  
*Timer de temps donné pour la comparaison.*
- `QPushButton * m_nextButton`  
*Un bouton pour passer à la comparaison suivante.*
- `QPushButton * m_startButton`  
*Un bouton pour passer de l'entraînement au test.*
- `QPushButton * m_startTrainingButton`  
*Un bouton pour commencer l'entraînement.*
- `QPushButton * m_reStartTrainingButton`  
*Un bouton pour recommencer l'entraînement.*
- `QPushButton * m_quitButton`  
*Un bouton pour quitter le protocole de test.*
- `QPushButton * m_startTestButton`  
*Un bouton pour commencer le test reel.*
- `QTime * m_time`  
*Indique le temps de réponse.*
- `QVector< int > m_respondTime`  
*Stocke les temps de réponse.*
- `int m_state`  
*Indicateur de déroulement du test.*
- `QLabel * m_info`  
*Label d'informations.*
- `QLabel * m_debug`  
*Label de debug.*

### 4.1.1 Description détaillée

La classe `CentralWidget`.

La classe qui commande le déroulement du protocole, l'interaction avec l'utilisateur et la récupération de ses choix.

### 4.1.2 Documentation des constructeurs et destructeur

#### 4.1.2.1 `CentralWidget : : CentralWidget ( QWidget * parent = 0 )`

Constructeur `CentralWidget`.

Constructeur de la classe `CentralWidget`

Paramètres

<i>parent</i>	: widget parent
---------------	-----------------

#### 4.1.2.2 `CentralWidget : : ~CentralWidget ( )` [virtual]

Destructeur `CentralWidget`.

Destructeur de la classe `CentralWidget`

### 4.1.3 Documentation des fonctions membres



## 4.1.3.1 void CentralWidget : :cleanButton ( )

Cache tout les boutons à l'écran.

Tout les boutons à l'écran sont masqué et l'utilisateur ne peut plus interagir avec eux. Les boutons nécessaires à la phase suivante doivent être alors réafficher pour l'utilisateur.

## 4.1.3.2 void CentralWidget : :endLoop ( ) [slot]

Slot appelé par un timer pour arrêter les vidéos.

Arrête la lecture des vidéos au bout du timer de 30 sec pour forcer le choix de l'utilisateur.

## 4.1.3.3 void CentralWidget : :endTest ( )

Termine le test.

Est appelé à la fin du test, affiche les remerciements et ne laisse pour evenement possible que la fermeture de l'application de test.

## 4.1.3.4 void CentralWidget : :endTraining ( )

Termine la phase d'entraînement.

Est appelé lors de l'achèvement de la phase d'entraînement, prépare le lancement du test reel ou si besoin un nouvel entraînement.

## 4.1.3.5 void CentralWidget : :nextTest ( ) [slot]

Slot appelé lors du lancement par l'utilisateur des vidéos pour leur comparaison.

Met en place la comparaison suivante, lance le test suivant.

4.1.3.6 void CentralWidget : :saveChoice ( int *choice* ) [slot]

Slot appelé lors de la selection d'une vidéo.

## Paramètres

	choix de l'utilisateur
--	------------------------

Suivant le choix de l'utilisateur, le sauvegarde et met le test en pause.

4.1.3.7 void CentralWidget : :setStyleButton ( QPushButton \* *button*, int *i* = 0 )

Applique un style predefinit à un bouton.

## Paramètres

<i>button</i>	: Un widget bouton
<i>i</i>	: Indique la video associée au bouton, 1 pour gauche, 2 pour droite, 0 si pas d'association.

Modifie le style d'un bouton, ainsi que le texte adéquat lorsqu'il s'agit d'un bouton de choix de vidéo.

## 4.1.3.8 void CentralWidget : :startTest ( ) [slot]

Slot appelé par l'utilisateur pour démarrer le test reel.

Prépare l'interface et lance la phase de test reel.

#### 4.1.3.9 void CentralWidget : :startTraining ( ) [slot]

Slot appelé par l'utilisateur pour démarrer la phase d'entraînement.

Prépare l'interface et lance la phase d'entraînement.

### 4.1.4 Documentation des données membres

#### 4.1.4.1 QVector<QPushButton\*> CentralWidget : :m\_buttons [private]

Les boutons de choix des videos.

#### 4.1.4.2 QVector<int> CentralWidget : :m\_choices [private]

Tableau recuperant les choix de l'utilisateur.

#### 4.1.4.3 QTimer\* CentralWidget : :m\_compareTimer [private]

Timer de temps donné pour la comparaison.

#### 4.1.4.4 QLabel\* CentralWidget : :m\_debug [private]

Label de debug.

#### 4.1.4.5 QLabel\* CentralWidget : :m\_info [private]

Label d'informations.

#### 4.1.4.6 QGridLayout\* CentralWidget : :m\_layout [private]

Layout général.

#### 4.1.4.7 QPushButton\* CentralWidget : :m\_nextButton [private]

Un bouton pour passer à la comparaison suivante.

#### 4.1.4.8 QPushButton\* CentralWidget : :m\_quitButton [private]

Un bouton pour quitter le protocole de test.

#### 4.1.4.9 QVector<int> CentralWidget : :m\_respondTime [private]

Stocke les temps de réponse.

#### 4.1.4.10 QPushButton\* CentralWidget : :m\_reStartTrainingButton [private]

Un bouton pour recommencer l'entraînement.

## 4.1.4.11 QSignalMapper\* CentralWidget : :m\_signalMapperChoice [private]

Pour mapper les bouton de choix.

## 4.1.4.12 QPushButton\* CentralWidget : :m\_startButton [private]

Un bouton pour passer de l'entrainement au test.

## 4.1.4.13 QPushButton\* CentralWidget : :m\_startTestButton [private]

Un bouton pour commencer le test reel.

## 4.1.4.14 QPushButton\* CentralWidget : :m\_startTrainingButton [private]

Un bouton pour commencer l'entrainement.

## 4.1.4.15 int CentralWidget : :m\_state [private]

Indicateur de deroulement du test.

## 4.1.4.16 QTime\* CentralWidget : :m\_time [private]

Indique le temps de réponse.

## 4.1.4.17 VideosWidget\* CentralWidget : :m\_videosWidget [private]

L'espace dédié au placement des vidéos.

La documentation de cette classe a été générée à partir des fichiers suivants :

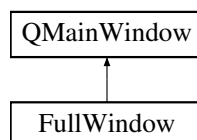
- [centralwidget.h](#)
- [centralwidget.cpp](#)

## 4.2 Référence de la classe FullWindow

La classe [FullWindow](#).

```
#include <fullwindow.h>
```

Graphe d'héritage de FullWindow :



### Fonctions membres publiques

- [FullWindow](#) ()  
Constructeur [FullWindow](#).
- virtual [~FullWindow](#) ()  
Destructeur [FullWindow](#).
- void [keyPressEvent](#) (QKeyEvent \*event)

- *Récupération d'un evenement clavier.*
- void [resizeEvent](#) ()  
*Detection et gestion d'un redimensionnement.*

## Attributs privés

- QWidget \* [m\\_mainWidget](#)  
*Widget principal, qui englobe tout l'écran.*
- QGridLayout \* [m\\_layout](#)  
*Layout generale, pour afficher le widget central dans la fenêtre plein écran.*
- [CentralWidget](#) \* [m\\_centralWidget](#)  
*Widget central, dont les coordonnées et la taille sont fixe.*

### 4.2.1 Description détaillée

La classe [FullWindow](#).

Gère l'affichage en plein écran, pour adapter l'interface de test à l'écran.

### 4.2.2 Documentation des constructeurs et destructeur

#### 4.2.2.1 FullWindow : :FullWindow ( )

Constructeur [FullWindow](#).

Constructeur de la classe [FullWindow](#).

Paramètres

<i>parent</i>	: widget parent
---------------	-----------------

#### 4.2.2.2 FullWindow : :~FullWindow ( ) [virtual]

Destructeur [FullWindow](#).

Destructeur de la classe [FullWindow](#).

### 4.2.3 Documentation des fonctions membres

#### 4.2.3.1 void FullWindow : :keyPressEvent ( QKeyEvent \* event )

Récupération d'un evenement clavier.

Paramètres

<i>event</i>	: La touche sur laquelle l'utilisateur appuie.
--------------	--

Reimplementation de la fonction keyPressEvent pour quitter le test si l'utilisateur appuie sur echap. A desactiver ultérieurement ?

#### 4.2.3.2 void FullWindow : :resizeEvent ( )

Detection et gestion d'un redimensionnement.

Reimplementation de la fonction resizeEvent pour réadapter l'interface en cas de redimensionnement de l'écran.

#### 4.2.4 Documentation des données membres

##### 4.2.4.1 CentralWidget\* FullWindow : :m\_centralWidget [private]

Widget central, dont les coordonnées et la taille sont fixe.

##### 4.2.4.2 QGridLayout\* FullWindow : :m\_layout [private]

Layout generale, pour afficher le widget central dans la fenêtre plein écran.

##### 4.2.4.3 QWidget\* FullWindow : :m\_mainWidget [private]

Widget principal, qui englobe tout l'écran.

La documentation de cette classe a été générée à partir des fichiers suivants :

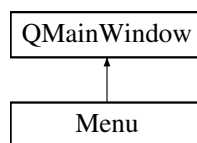
- [fullwindow.h](#)
- [fullwindow.cpp](#)

### 4.3 Référence de la classe Menu

La classe [Menu](#).

```
#include <menu.h>
```

Graphe d'héritage de Menu :



#### Connecteurs publics

- void [startTest](#) ()  
*Lancement du test.*

#### Fonctions membres publiques

- [Menu](#) (QWidget \*parent=0)  
*Constructeur [Menu](#).*
- virtual [~Menu](#) ()  
*Destructeur [Menu](#).*

#### Attributs privés

- QWidget \* [m\\_widget](#)  
*widget central*
- QGridLayout \* [m\\_layout](#)  
*layout central*
- QLabel \* [m\\_label](#)  
*text descriptif*
- QPushButton \* [m\\_startButton](#)  
*bouton de lancement*
- [FullWindow](#) \* [m\\_fullWindow](#)  
*classe d'affichage en plein écran pour le test*

### 4.3.1 Description détaillée

La classe [Menu](#).

La classe gère l'affichage du menu et l'accès au test.

### 4.3.2 Documentation des constructeurs et destructeur

#### 4.3.2.1 `Menu : :Menu ( QWidget * parent = 0 )`

Constructeur [Menu](#).

Constructeur de la classe [Menu](#)

Paramètres

<i>parent</i>	: widget parent
---------------	-----------------

#### 4.3.2.2 `Menu : :~Menu ( ) [virtual]`

Destructeur [Menu](#).

Destructeur de la classe [Menu](#)

### 4.3.3 Documentation des fonctions membres

#### 4.3.3.1 `void Menu : :startTest ( ) [slot]`

Lancement du test.

Instancie la classe [FullWindow](#), pour placer l'utilisateur dans les conditions du test.

### 4.3.4 Documentation des données membres

#### 4.3.4.1 `FullWindow* Menu : :m_fullWindow [private]`

classe d'affichage en plein écran pour le test

#### 4.3.4.2 `QLabel* Menu : :m_label [private]`

text descriptif

#### 4.3.4.3 `QGridLayout* Menu : :m_layout [private]`

layout central

#### 4.3.4.4 `QPushButton* Menu : :m_startButton [private]`

boutton de lancement

#### 4.3.4.5 `QWidget* Menu : :m_widget [private]`

widget central

La documentation de cette classe a été générée à partir des fichiers suivants :

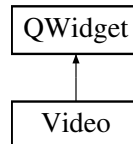
- [menu.h](#)
- [menu.cpp](#)

## 4.4 Référence de la classe Video

La classe [Video](#).

```
#include <video.h>
```

Graphe d'héritage de Video :



### Fonctions membres publiques

- [Video](#) (QWidget \*parent=0)  
*Constructeur [Video](#).*
- virtual [~Video](#) ()  
*Destructeur [~Video](#).*
- double [setVideo](#) (const char \*path)  
*Chargement d'une vidéo.*
- int [nextFrame](#) ()  
*Affichage à l'écran d'une frame.*
- QImage [lpl2QImage](#) (const IplImage \*newImage)  
*Conversion du type IplImage vers QImage.*

### Attributs privés

- QGridLayout \* [m\\_layout](#)  
*Layout pour positionner le label.*
- QLabel \* [m\\_label](#)  
*Label pour y poser les frames.*
- CvCapture \* [m\\_capture](#)  
*capture du flux video.*
- IplImage \* [m\\_image](#)  
*image pour récupérer les frames stocké dans m\_capture une par une.*

#### 4.4.1 Description détaillée

La classe [Video](#).

La classe propose un widget permettant d'afficher un flux video.

#### 4.4.2 Documentation des constructeurs et destructeur

##### 4.4.2.1 Video::Video ( QWidget \* parent = 0 )

Constructeur [Video](#).

## Paramètres

<i>parent</i>	Constructeur de la classe <a href="#">Video</a> .
---------------	---

4.4.2.2 `Video::~~Video ( )` [virtual]

Destructeur `~Video`.

Destructeur de la classe [Video](#).

## 4.4.3 Documentation des fonctions membres

4.4.3.1 `QImage Video::lpl2QImage ( const QImage * newImage )`

Conversion du type `lplImage` vers `QImage`.

## Paramètres

<i>newImage</i>	: Une image au format <code>lplImage</code> .
-----------------	---

Les frames des vidéos étant capturées au format `lpl2Qimage`, une conversion est necessaire pour l'affichage de l'image dans un widget.

## Renvoie

Image d'origine au format `QImage`.

4.4.3.2 `int Video::nextFrame ( )`

Affichage à l'écran d'une frame.

Affiche la frame suivante de la vidéo.

## Renvoie

0 si la frame est récupérée, -1 sinon, par exemple si la video est terminée.

4.4.3.3 `double Video::setVideo ( const char * path )`

Chargement d'une vidéo.

## Paramètres

<i>path</i>	: chemin vers la video à charger.
-------------	-----------------------------------

Charge une video située à un chemin donné, pour un affichage ultérieur.

## Renvoie

0 si la video est trouvée et chargée, -1 sinon.

## 4.4.4 Documentation des données membres

4.4.4.1 `CvCapture* Video::m_capture` [private]

capture du flux video.

4.4.4.2 `lplImage* Video::m_image` [private]

image pour récupérer les frames stocké dans `m_capture` une par une.



## 4.4.4.3 QLabel\* Video : :m\_label [private]

Label pour y poser les frames.

## 4.4.4.4 QGridLayout\* Video : :m\_layout [private]

Layout pour positionner le label.

La documentation de cette classe a été générée à partir des fichiers suivants :

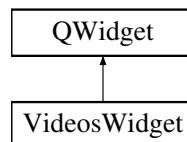
- [video.h](#)
- [video.cpp](#)

## 4.5 Référence de la classe VideosWidget

La classe [VideosWidget](#).

```
#include <videoswidget.h>
```

Graphe d'héritage de VideosWidget :



### Connecteurs publics

- void [runVideos](#) ()  
*Affiche les frames suivantes des trois videos.*
- void [start](#) ()  
*recharge les vidéos*

### Fonctions membres publiques

- [VideosWidget](#) (QWidget \*parent=0, int w=800, int h=600)  
*Constructeur [VideosWidget](#).*
- virtual [~VideosWidget](#) ()  
*Destructeur [VideosWidget](#).*
- void [loadSequence](#) (std : :string path)  
*loadSequence*
- bool [exportResult](#) (QVector< int > choices, QVector< int > respondTime)  
*exportation des résultats du test*
- void [organizeLayout](#) (double resolution)  
*Organisation du layout vidéos.*
- void [stop](#) ()  
*stop*
- void [play](#) ()  
*play*
- bool [loadNext](#) ()  
*Chargement des vidéos suivantes.*
- void [setFocus](#) (int i)  
*Met en avant une vidéo.*
- void [resizeLayout](#) (int w, int h)  
*redimensionnement du layout*
- void [cleanVideos](#) ()  
*Cache les vidéos.*
- void [showVideos](#) ()  
*Affiche les vidéos.*

## Attributs privés

- int [m\\_height](#)  
*hauteur du layout*
- int [m\\_width](#)  
*largeur du layout*
- int [m\\_cmpSequence](#)  
*compteur de comparaison effectué par l'utilisateur*
- std::string [m\\_sequencePath](#)  
*Le nom du fichier contenant la sequence.*
- QVector< std::string > [m\\_sequence](#)  
*La sequence de test.*
- QVector< [Video](#) \* > [m\\_videos](#)  
*les trois videos à afficher*
- QLabel \* [m\\_labelRef](#)  
*texte "référence" pour indiquer la référence*
- QGridLayout \* [m\\_videoLayout](#)  
*Layout des vidéos.*
- QTimer \* [m\\_timer](#)  
*timer de synchronisation des videos*
- QTimer \* [m\\_boucleTimer](#)  
*timer de boucle des videos*

### 4.5.1 Description détaillée

La classe [VideosWidget](#).

Classe d'affichage et de gestion de trois flux vidéos en parallèle, ainsi que de gestion de flux d'entrée/sortie des séquences et des résultats

### 4.5.2 Documentation des constructeurs et destructeur

#### 4.5.2.1 [VideosWidget](#) : [VideosWidget](#) ( [QWidget](#) \* *parent* = 0, int *w* = 800, int *h* = 600 )

Constructeur [VideosWidget](#).

Constructeur de la classe [VideosWidget](#)

Paramètres

<i>parent</i>	: widget parent
<i>h</i>	: hauteur du widget
<i>w</i>	: largeur du widget

#### 4.5.2.2 [VideosWidget](#) : [~VideosWidget](#) ( ) [virtual]

Destructeur [VideosWidget](#).

Destructeur de la classe [VideosWidget](#)

### 4.5.3 Documentation des fonctions membres

#### 4.5.3.1 void [VideosWidget](#) : [cleanVideos](#) ( )

Cache les vidéos.

Les trois vidéos sont cachées, pour assurer une transition correct entre les différentes phase du test.

4.5.3.2 bool VideosWidget :exportResult ( QVector< int > *choices*, QVector< int > *respondTime* )

exportation des résultats du test

**Paramètres**

<i>choices</i>	: tableau des choix de l'utilisateur
<i>respondTime</i>	: tableau des temps de réponse associés

Exporte les résultats du test, avec les temps de réponse.

**4.5.3.3 bool VideosWidget :loadNext ( )**

Chargement des vidéos suivantes.

Charge les trois vidéos suivantes de la sequence

**4.5.3.4 void VideosWidget :loadSequence ( std :string path )**

loadSequence

Charge une séquence de test

**4.5.3.5 void VideosWidget :organizeLayout ( double resolution )**

Organisation du layout vidéos.

**Paramètres**

<i>resolution</i>	: Les resolutions respectives des trois videos
-------------------	--

Permet de modifier le layout pour qu'il s'adapte au format des videos (hauteur, placement vis à vis de chacune)

**4.5.3.6 void VideosWidget :play ( )**

play

Met en route les vidéos

**4.5.3.7 void VideosWidget :resizeLayout ( int w, int h )**

redimensionnement du layout

**Paramètres**

<i>w</i>	: une largeur
<i>h</i>	: une hauteur

Redimensionne le layout, notamment entre la phase d'entraînement et le test réel.

**4.5.3.8 void VideosWidget :runVideos ( ) [slot]**

Affiche les frames suivantes des trois videos.

Appel la méthode d'affichage de frame suivante des trois vidéos, pour une lecture synchronisée.

**4.5.3.9 void VideosWidget :setFocus ( int i )**

Met en avant une vidéo.

## Paramètres

<i>i</i>	: numero de la video
----------	----------------------

encadre en vert la vidéo indiquée

## 4.5.3.10 void VideosWidget : :showVideos ( )

Affiche les vidéos.

Affiche les vidéos à l'écran.

## 4.5.3.11 void VideosWidget : :start ( ) [slot]

recharge les vidéos

recharge les vidéos de la séquence de test, i.e. remet au début la lecture des vidéos.

## 4.5.3.12 void VideosWidget : :stop ( )

stop

Met en pause les vidéos

## 4.5.4 Documentation des données membres

## 4.5.4.1 QTimer\* VideosWidget : :m\_boucleTimer [private]

timer de boucle des videos

## 4.5.4.2 int VideosWidget : :m\_cmpSequence [private]

compteur de comparaison effectué par l'utilisateur

## 4.5.4.3 int VideosWidget : :m\_height [private]

hauteur du layout

## 4.5.4.4 QLabel\* VideosWidget : :m\_labelRef [private]

texte "référence" pour indiquer la référence

## 4.5.4.5 QVector&lt;std : :string&gt; VideosWidget : :m\_sequence [private]

La sequence de test.

## 4.5.4.6 std : :string VideosWidget : :m\_sequencePath [private]

Le nom du fichier contenant la sequence.

## 4.5.4.7 QTimer\* VideosWidget : :m\_timer [private]

timer de synchronisation des videos

**4.5.4.8** `QGridLayout* VideosWidget : :m_videoLayout` [private]

Layout des vidéos.

**4.5.4.9** `QVector<Video*> VideosWidget : :m_videos` [private]

les trois videos à afficher

**4.5.4.10** `int VideosWidget : :m_width` [private]

largeur du layout

La documentation de cette classe a été générée à partir des fichiers suivants :

- [videoswidget.h](#)
- [videoswidget.cpp](#)

## Chapitre 5

# Documentation des fichiers

### 5.1 Référence du fichier centralwidget.cpp

méthodes de la classe [CentralWidget](#)

```
#include "centralwidget.h"  
#include <QApplication>  
#include <QDesktopWidget>
```

#### 5.1.1 Description détaillée

méthodes de la classe [CentralWidget](#) Gestion du déroulement du test, gestion de l'interface utilisateur

Auteur

Pierre Lespingal

### 5.2 Référence du fichier centralwidget.h

```
#include "define.h"  
#include <QGridLayout>  
#include <QMouseEvent>  
#include <QKeyEvent>  
#include <QVector>  
#include <QString>  
#include <QMessageBox>  
#include "video.h"  
#include "videoswidget.h"  
#include <QTimer>  
#include <QTime>  
#include <QDateTime>  
#include <fstream>  
#include <string>  
#include <QSignalMapper>  
#include <QPushButton>
```

#### Classes

— class [CentralWidget](#)

La classe [CentralWidget](#).

## 5.3 Référence du fichier define.h

Enumeration pour le déroulement du protocole et grandeurs globales pour le dimensionnement de l'interface.

### Macros

- `#define NB_TEST 100`  
*Nombre de comparaison maximum lors d'un test.*
- `#define LAYOUT_HEIGHT 900`  
*Hauteur de l'espace de travail en temps normal.*
- `#define LAYOUT_VIDEOS_HEIGHT 600`  
*Hauteur de l'espace d'affichage des vidéos en temps normal.*
- `#define LAYOUT_VIDEOS_HEIGHT_T 800`  
*Hauteur de l'espace d'affichage des vidéos pendant la phase d'entraînement.*
- `#define LAYOUT_WIDTH 1066`  
*Largeur de l'espace de travail en temps normal.*
- `#define LAYOUT_VIDEOS_WIDTH 800`  
*Hauteur de l'espace d'affichage des vidéos en temps normal.*
- `#define LAYOUT_VIDEOS_WIDTH_T 1066`  
*Hauteur de l'espace d'affichage des vidéos pendant la phase d'entraînement.*
- `#define LAYOUT_SPACE 20`  
*espacement horizontal entre deux objets*
- `#define LAYOUT_V_SPACE 4`  
*espacement vertical entre deux objets*
- `#define LAYOUT_R LAYOUT_VIDEOS_HEIGHT/LAYOUT_VIDEOS_WIDTH`  
*Resolution de l'espace vidéo en temps normal.*
- `#define LAYOUT_R_T LAYOUT_VIDEOS_HEIGHT_T/LAYOUT_VIDEOS_WIDTH_T`  
*Resolution de l'espace vidéo pendant la pahse d'entraînement.*

### Énumérations

- `enum State { Training, Test, End }`  
*Enumeration state.*

#### 5.3.1 Description détaillée

Enumeration pour le déroulement du protocole et grandeurs globales pour le dimensionnement de l'interface.

##### Auteur

Pierre Lespingal

#### 5.3.2 Documentation des macros

##### 5.3.2.1 `#define LAYOUT_HEIGHT 900`

Hauteur de l'espace de travail en temps normal.

##### 5.3.2.2 `#define LAYOUT_R LAYOUT_VIDEOS_HEIGHT/LAYOUT_VIDEOS_WIDTH`

Resolution de l'espace vidéo en temps normal.

##### 5.3.2.3 `#define LAYOUT_R_T LAYOUT_VIDEOS_HEIGHT_T/LAYOUT_VIDEOS_WIDTH_T`

Resolution de l'espace vidéo pendant la pahse d'entraînement.



#### 5.3.2.4 #define LAYOUT\_SPACE 20

espacement horizontal entre deux objets

#### 5.3.2.5 #define LAYOUT\_V\_SPACE 4

espacement vertical entre deux objets

#### 5.3.2.6 #define LAYOUT\_VIDEOS\_HEIGHT 600

Hauteur de l'espace d'affichage des vidéos en temps normal.

#### 5.3.2.7 #define LAYOUT\_VIDEOS\_HEIGHT\_T 800

Hauteur de l'espace d'affichage des vidéos pendant la phase d'entraînement.

#### 5.3.2.8 #define LAYOUT\_VIDEOS\_WIDTH 800

Hauteur de l'espace d'affichage des vidéos en temps normal.

#### 5.3.2.9 #define LAYOUT\_VIDEOS\_WIDTH\_T 1066

Hauteur de l'espace d'affichage des vidéos pendant la phase d'entraînement.

#### 5.3.2.10 #define LAYOUT\_WIDTH 1066

Largeur de l'espace de travail en temps normal.

#### 5.3.2.11 #define NB\_TEST 100

Nombre de comparaison maximum lors d'un test.

### 5.3.3 Documentation du type de l'énumération

#### 5.3.3.1 enum State

Enumeration state.

Enumeration permettant de se reperer dans les différentes phases du protocole.

Valeurs énumérées

***Training***

***Test***

***End***

## 5.4 Référence du fichier fullwindow.cpp

méthodes de la classe [FullWindow](#)

```
#include "fullwindow.h"
#include <QMessageBox>
#include <QApplication>
#include <QDesktopWidget>
```

### 5.4.1 Description détaillée

méthodes de la classe [FullWindow](#) Gestion du plein écran, placement de la zone d'interaction avec l'utilisateur

#### Auteur

Pierre Lespingal

## 5.5 Référence du fichier fullwindow.h

Fenêtre plein écran.

```
#include "define.h"
#include <QWidget>
#include <QMainWindow>
#include "centralwidget.h"
```

### Classes

- class [FullWindow](#)  
*La classe [FullWindow](#).*

### Macros

- #define [WIDGET\\_WIDTH](#) 1000
- #define [WIDGET\\_HEIGHT](#) 600

### 5.5.1 Description détaillée

Fenêtre plein écran.

#### Auteur

Pierre Lespingal

### 5.5.2 Documentation des macros

5.5.2.1 #define [WIDGET\\_HEIGHT](#) 600

5.5.2.2 #define [WIDGET\\_WIDTH](#) 1000

## 5.6 Référence du fichier main.cpp

Interface de test subjectif.

```
#include <QApplication>
#include <QTextCodec>
#include "menu.h"
```

## Fonctions

— int [main](#) (int argc, char \*argv[])

### 5.6.1 Description détaillée

Interface de test subjectif.

#### Auteur

Pierre Lespingal

#### Date

10 Juin 2014

Propose un protocole de test utilisateur pour l'évaluation de la qualité d'objet 3D

### 5.6.2 Documentation des fonctions

5.6.2.1 int [main](#) ( int *argc*, char \* *argv*[] )

## 5.7 Référence du fichier menu.cpp

méthodes de la classe [Menu](#)

```
#include "menu.h"
```

### 5.7.1 Description détaillée

méthodes de la classe [Menu](#) Gestion du menu de base

#### Auteur

Pierre Lespingal

## 5.8 Référence du fichier menu.h

[Menu](#) de base.

```
#include <QMainWindow>
#include <QPushButton>
#include <QLabel>
#include <QGridLayout>
#include "fullwindow.h"
```

## Classes

- class [Menu](#)  
*La classe [Menu](#).*

### 5.8.1 Description détaillée

[Menu](#) de base.

Auteur

Pierre Lespingal

## 5.9 Référence du fichier video.cpp

méthodes de la classe [Video](#)

```
#include "video.h"
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
#include <QMessageBox>
#include <QDir>
```

### 5.9.1 Description détaillée

méthodes de la classe [Video](#) Gestion d'un flux vidéo

Auteur

Pierre Lespingal

## 5.10 Référence du fichier video.h

widget d'affichage d'une video

```
#include <define.h>
#include <QWidget>
#include <QGridLayout>
#include <QLabel>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
```

## Classes

- class [Video](#)  
*La classe [Video](#).*

### 5.10.1 Description détaillée

widget d'affichage d'une video

Auteur

Pierre Lespingal

## 5.11 Référence du fichier videoswidget.cpp

```
#include "videoswidget.h"
```

## 5.12 Référence du fichier videoswidget.h

```
#include <QWidget>
#include <vector>
#include <QGridLayout>
#include <QDateTime>
#include <QVector>
#include "video.h"
#include <QMessageBox>
#include <QTimer>
#include <fstream>
```

### Classes

- class [VideosWidget](#)  
*La classe [VideosWidget](#).*

# Index

- ~CentralWidget
  - CentralWidget, [8](#)
- ~FullWindow
  - FullWindow, [12](#)
- ~Menu
  - Menu, [14](#)
- ~Video
  - Video, [16](#)
- ~VideosWidget
  - VideosWidget, [18](#)
- CentralWidget, [7](#)
  - ~CentralWidget, [8](#)
  - CentralWidget, [8](#)
  - CentralWidget, [8](#)
  - cleanButton, [8](#)
  - endLoop, [9](#)
  - endTest, [9](#)
  - endTraining, [9](#)
  - m\_buttons, [10](#)
  - m\_choices, [10](#)
  - m\_compareTimer, [10](#)
  - m\_debug, [10](#)
  - m\_info, [10](#)
  - m\_layout, [10](#)
  - m\_nextButton, [10](#)
  - m\_quitButton, [10](#)
  - m\_reStartTrainingButton, [10](#)
  - m\_respondTime, [10](#)
  - m\_signalMapperChoice, [10](#)
  - m\_startButton, [11](#)
  - m\_startTestButton, [11](#)
  - m\_startTrainingButton, [11](#)
  - m\_state, [11](#)
  - m\_time, [11](#)
  - m\_videosWidget, [11](#)
  - nextTest, [9](#)
  - saveChoice, [9](#)
  - setStyleButton, [9](#)
  - startTest, [9](#)
  - startTraining, [9](#)
- centralwidget.cpp, [23](#)
- centralwidget.h, [23](#)
- cleanButton
  - CentralWidget, [8](#)
- cleanVideos
  - VideosWidget, [18](#)
- define.h
  - End, [25](#)
- Test, [25](#)
- Training, [25](#)
- define.h, [24](#)
  - LAYOUT\_HEIGHT, [24](#)
  - LAYOUT\_R, [24](#)
  - LAYOUT\_R\_T, [24](#)
  - LAYOUT\_SPACE, [24](#)
  - LAYOUT\_V\_SPACE, [25](#)
  - LAYOUT\_VIDEOS\_WIDTH, [25](#)
  - LAYOUT\_WIDTH, [25](#)
  - NB\_TEST, [25](#)
  - State, [25](#)
- End
  - define.h, [25](#)
- endLoop
  - CentralWidget, [9](#)
- endTest
  - CentralWidget, [9](#)
- endTraining
  - CentralWidget, [9](#)
- exportResult
  - VideosWidget, [18](#)
- FullWindow, [11](#)
  - ~FullWindow, [12](#)
  - FullWindow, [12](#)
  - FullWindow, [12](#)
  - keyPressEvent, [12](#)
  - m\_centralWidget, [13](#)
  - m\_layout, [13](#)
  - m\_mainWidget, [13](#)
  - resizeEvent, [12](#)
- fullwindow.cpp, [25](#)
- fullwindow.h, [26](#)
  - WIDGET\_HEIGHT, [26](#)
  - WIDGET\_WIDTH, [26](#)
- Ipl2QImage
  - Video, [16](#)
- keyPressEvent
  - FullWindow, [12](#)
- LAYOUT\_HEIGHT
  - define.h, [24](#)
- LAYOUT\_R
  - define.h, [24](#)
- LAYOUT\_R\_T
  - define.h, [24](#)
- LAYOUT\_SPACE

- define.h, [24](#)
- LAYOUT\_V\_SPACE
  - define.h, [25](#)
- LAYOUT\_VIDEOS\_HEIGHT
  - define.h, [25](#)
- LAYOUT\_VIDEOS\_WIDTH
  - define.h, [25](#)
- LAYOUT\_WIDTH
  - define.h, [25](#)
- loadNext
  - VideosWidget, [20](#)
- loadSequence
  - VideosWidget, [20](#)
- m\_boucleTimer
  - VideosWidget, [21](#)
- m\_buttons
  - CentralWidget, [10](#)
- m\_capture
  - Video, [16](#)
- m\_centralWidget
  - FullWindow, [13](#)
- m\_choices
  - CentralWidget, [10](#)
- m\_cmpSequence
  - VideosWidget, [21](#)
- m\_compareTimer
  - CentralWidget, [10](#)
- m\_debug
  - CentralWidget, [10](#)
- m\_fullWindow
  - Menu, [14](#)
- m\_height
  - VideosWidget, [21](#)
- m\_image
  - Video, [16](#)
- m\_info
  - CentralWidget, [10](#)
- m\_label
  - Menu, [14](#)
  - Video, [16](#)
- m\_labelRef
  - VideosWidget, [21](#)
- m\_layout
  - CentralWidget, [10](#)
  - FullWindow, [13](#)
  - Menu, [14](#)
  - Video, [17](#)
- m\_mainWidget
  - FullWindow, [13](#)
- m\_nextButton
  - CentralWidget, [10](#)
- m\_quitButton
  - CentralWidget, [10](#)
- m\_reStartTrainingButton
  - CentralWidget, [10](#)
- m\_respondTime
  - CentralWidget, [10](#)
- m\_sequence
  - VideosWidget, [21](#)
- m\_sequencePath
  - VideosWidget, [21](#)
- m\_signalMapperChoice
  - CentralWidget, [10](#)
- m\_startButton
  - CentralWidget, [11](#)
  - Menu, [14](#)
- m\_startTestButton
  - CentralWidget, [11](#)
- m\_startTrainingButton
  - CentralWidget, [11](#)
- m\_state
  - CentralWidget, [11](#)
- m\_time
  - CentralWidget, [11](#)
- m\_timer
  - VideosWidget, [21](#)
- m\_videoLayout
  - VideosWidget, [21](#)
- m\_videos
  - VideosWidget, [22](#)
- m\_videosWidget
  - CentralWidget, [11](#)
- m\_widget
  - Menu, [14](#)
- m\_width
  - VideosWidget, [22](#)
- main
  - main.cpp, [27](#)
- main.cpp, [26](#)
  - main, [27](#)
- Menu, [13](#)
  - ~Menu, [14](#)
  - m\_fullWindow, [14](#)
  - m\_label, [14](#)
  - m\_layout, [14](#)
  - m\_startButton, [14](#)
  - m\_widget, [14](#)
  - Menu, [14](#)
  - startTest, [14](#)
- menu.cpp, [27](#)
- menu.h, [27](#)
- NB\_TEST
  - define.h, [25](#)
- nextFrame
  - Video, [16](#)
- nextTest
  - CentralWidget, [9](#)
- organizeLayout
  - VideosWidget, [20](#)
- play
  - VideosWidget, [20](#)
- resizeEvent
  - FullWindow, [12](#)

- resizeLayout
  - VideosWidget, 20
- runVideos
  - VideosWidget, 20
- saveChoice
  - CentralWidget, 9
- setFocus
  - VideosWidget, 20
- setStyleButton
  - CentralWidget, 9
- setVideo
  - Video, 16
- showVideos
  - VideosWidget, 21
- start
  - VideosWidget, 21
- startTest
  - CentralWidget, 9
  - Menu, 14
- startTraining
  - CentralWidget, 9
- State
  - define.h, 25
- stop
  - VideosWidget, 21
- Test
  - define.h, 25
- Training
  - define.h, 25
- Video, 15
  - ~Video, 16
  - Ipl2QImage, 16
  - m\_capture, 16
  - m\_image, 16
  - m\_label, 16
  - m\_layout, 17
  - nextFrame, 16
  - setVideo, 16
  - Video, 15
- video.cpp, 28
- video.h, 28
- VideosWidget, 17
  - ~VideosWidget, 18
  - cleanVideos, 18
  - exportResult, 18
  - loadNext, 20
  - loadSequence, 20
  - m\_boucleTimer, 21
  - m\_cmpSequence, 21
  - m\_height, 21
  - m\_labelRef, 21
  - m\_sequence, 21
  - m\_sequencePath, 21
  - m\_timer, 21
  - m\_videoLayout, 21
  - m\_videos, 22
  - m\_width, 22
  - organizeLayout, 20
  - play, 20
  - resizeLayout, 20
  - runVideos, 20
  - setFocus, 20
  - showVideos, 21
  - start, 21
  - stop, 21
  - VideosWidget, 18
  - VideosWidget, 18
- videoswidget.cpp, 29
- videoswidget.h, 29
- WIDGET\_HEIGHT
  - fullwindow.h, 26
- WIDGET\_WIDTH
  - fullwindow.h, 26