

POOW Segmentation

Généré par Doxygen 1.9.1

1 POOW Segmentation	1
1.1 Installation	1
1.1.1 Prérequis	1
1.1.2 Compiler les sources	1
1.2 Features principales	1
1.3 Contacts	1
2 Index hiérarchique	3
2.1 Hiérarchie des classes	3
3 Index des classes	5
3.1 Liste des classes	5
4 Documentation des classes	7
4.1 Référence de la classe CControle	7
4.1.1 Description détaillée	7
4.1.2 Documentation des constructeurs et destructeur	8
4.1.2.1 CControle()	8
4.1.3 Documentation des fonctions membres	8
4.1.3.1 getBoutonPrint()	8
4.1.3.2 getBoutonSave()	8
4.1.3.3 quitApp	9
4.1.4 Documentation des données membres	9
4.1.4.1 m_boutonPrint	9
4.1.4.2 m_boutonQuit	9
4.1.4.3 m_boutonSave	9
4.2 Référence de la classe CFenetre	10
4.2.1 Description détaillée	10
4.2.2 Documentation des fonctions membres	10
4.2.2.1 getImage()	10
4.2.2.2 getPatient()	11
4.2.3 Documentation des données membres	11
4.2.3.1 m_controle	11
4.2.3.2 m_image	11
4.2.3.3 m_patient	11
4.3 Référence de la classe CImage	11
4.3.1 Description détaillée	12
4.3.2 Documentation des constructeurs et destructeur	12
4.3.2.1 CImage()	12
4.3.3 Documentation des fonctions membres	13
4.3.3.1 coefChanged	13
4.3.3.2 formatOrRatioChanged	13
4.3.3.3 getArea()	13

4.3.3.4	getAreaLabel()	14
4.3.3.5	getFormatIntFromText()	14
4.3.3.6	getPaint()	14
4.3.3.7	updateAreaLabel	14
4.3.3.8	updateCoef	15
4.3.3.9	updateFormatHeight	15
4.3.3.10	updateRatioPercent	15
4.3.4	Documentation des données membres	15
4.3.4.1	c_areaLabel1	15
4.3.4.2	c_areaLabel2	16
4.3.4.3	m_areaLabel	16
4.3.4.4	m_boutonEffacerD	16
4.3.4.5	m_boutonEffacerT	16
4.3.4.6	m_boutonFermer	17
4.3.4.7	m_boutonLoad	17
4.3.4.8	m_coef	17
4.3.4.9	m_formatCB	17
4.3.4.10	m_formatHeight	18
4.3.4.11	m_paint	18
4.3.4.12	m_ratioPercent	18
4.3.4.13	m_ratioSB	18
4.4	Référence de la classe CPaint	19
4.4.1	Description détaillée	20
4.4.2	Documentation des constructeurs et destructeur	20
4.4.2.1	CPaint()	20
4.4.3	Documentation des fonctions membres	20
4.4.3.1	areaPxChanged	20
4.4.3.2	clearImage	20
4.4.3.3	closePoly	21
4.4.3.4	computeAreaPxFromPoints()	21
4.4.3.5	draw()	21
4.4.3.6	drawID	22
4.4.3.7	getAreaPx()	22
4.4.3.8	getImageHeight()	22
4.4.3.9	imageHeightChanged	22
4.4.3.10	mousePressEvent()	22
4.4.3.11	openDialog	23
4.4.3.12	openDialogSave	23
4.4.3.13	paintEvent()	23
4.4.3.14	removelast	24
4.4.3.15	updateAreaPx()	24
4.4.3.16	updateImageHeight()	24

4.4.4 Documentation des données membres	24
4.4.4.1 c_paintHeight	24
4.4.4.2 c_paintWidth	24
4.4.4.3 m_areaPx	25
4.4.4.4 m_fileName	25
4.4.4.5 m_image	25
4.4.4.6 m_imageHeight	25
4.4.4.7 m_liPoint	25
4.5 Référence de la classe CPatient	26
4.5.1 Description détaillée	26
4.5.2 Documentation des constructeurs et destructeur	26
4.5.2.1 CPatient()	26
4.5.3 Documentation des fonctions membres	27
4.5.3.1 getDate()	27
4.5.3.2 getNom()	27
4.5.3.3 getPrenom()	27
4.5.4 Documentation des données membres	27
4.5.4.1 m_date	28
4.5.4.2 m_nom	28
4.5.4.3 m_prenom	28
Index	29

Chapitre 1

POOW Segmentation

Application permettant de segmenter manuellement une image médicale.

Le polygone défini par l'utilisateur est affiché sur l'image et le programme calcule son aire. Les informations du patient sont imprimées sur l'image.



1.1 Installation

1.1.1 Prérequis

- Un compilateur C++ ;
- la librairie Qt <https://www.qt.io>.

1.1.2 Compiler les sources

1. Cloner le repo

```
git clone https://github.com/esiul/segmentation.git
cd segmentation
```

1. Créer un Makefile avec *qmake*

```
qmake
```

1. Compiler les sources

```
make -j4 && make clean
```

1. Ouvrir l'application

```
open PROJET.app
```

1.2 Features principales

- Segmenter une partie de l'image ;
- calculer l'aire de la partie segmentée ;
- imprimer les informations du patient sur l'image.

1.3 Contacts

Application créée par Louise Kuhl dans le cadre d'un projet informatique.

Signaler les problèmes à louise.kuhl@hotmail.fr.

Chapitre 2

Index hiérarchique

2.1 Hiérarchie des classes

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

QGroupBox	
CImage	11
CPatient	26
QWidget	
CControle	7
CFenetre	10
CPaint	19

Chapitre 3

Index des classes

3.1 Liste des classes

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

CControle	Widget qui gère les commandes de l'application	7
CFenetre	Fenêtre principale de l'application	10
CImage	Widget qui gère la partie "image" de l'application	11
CPaint	Widget qui affiche l'image et permet à l'utilisateur de dessiner dessus	19
CPatient	Widget qui gère les informations du patient	26

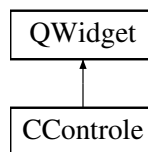
Chapitre 4

Documentation des classes

4.1 Référence de la classe CControle

```
#include <CControle.h>
```

Graphe d'héritage de CControle:



Connecteurs publics

- void `quitApp()`

Fonctions membres publiques

- `CControle` (`QWidget *parent=nullptr`)
- `QPushButton *` `getBoutonSave()`
- `QPushButton *` `getBoutonPrint()`

Attributs privés

- `QPushButton *` `m_boutonPrint`
- `QPushButton *` `m_boutonSave`
- `QPushButton *` `m_boutonQuit`

4.1.1 Description détaillée

Widget qui gère les commandes de l'application.

Les commandes sont :

- imprimer les informations du patient sur l'image ;
- sauvegarder l'image ;
- quitter l'application.

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 CControle()

```
CControle::CControle (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur par défaut.

Paramètres

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

4.1.3 Documentation des fonctions membres

4.1.3.1 getBoutonPrint()

```
QPushButton * CControle::getBoutonPrint ( )
```

Accesseur

Renvoie

[CControle::m_boutonPrint](#)

4.1.3.2 getBoutonSave()

```
QPushButton * CControle::getBoutonSave ( )
```

Accesseur

Renvoie

[CControle::m_boutonSave](#)

4.1.3.3 quitApp

```
void CControle::quitApp ( ) [slot]
```

Slot pour quitter l'application.

Dans un premier temps ouvre une boîte de dialogue pour demander confirmation. Si la réponse est oui, quitte l'application. Sinon, ne fait rien.

Voir également

[CControle::m_boutonQuit](#)

4.1.4 Documentation des données membres

4.1.4.1 m_boutonPrint

```
QPushButton* CControle::m_boutonPrint [private]
```

Bouton pour imprimer les infos sur l'image

Voir également

[CPaint::drawID](#)

4.1.4.2 m_boutonQuit

```
QPushButton* CControle::m_boutonQuit [private]
```

Bouton pour quitter l'application

Voir également

[CControle::quitApp](#)

4.1.4.3 m_boutonSave

```
QPushButton* CControle::m_boutonSave [private]
```

Bouton pour sauvegarder l'image

Voir également

[CPaint::openDialogSave](#)

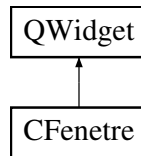
La documentation de cette classe a été générée à partir du fichier suivant :

- CControle.h
- CControle.cpp

4.2 Référence de la classe CFenetre

```
#include <CFenetre.h>
```

Graphe d'héritage de CFenetre:



Fonctions membres publiques

- [CFenetre](#) ()
- [CPatient](#) * [getPatient](#) ()
- [CImage](#) * [getImage](#) ()

Attributs privés

- [CPatient](#) * [m_patient](#)
- [CImage](#) * [m_image](#)
- [CControle](#) * [m_controle](#)

4.2.1 Description détaillée

Fenêtre principale de l'application.

Cette fenêtre contient trois `QGroupBox` pour découper les fonctionnalités de l'application.

4.2.2 Documentation des fonctions membres

4.2.2.1 getImage()

```
CImage * CFenetre::getImage ( )
```

Accesseur

Renvoie

[CFenetre::m_image](#)

4.2.2.2 getPatient()

```
CPatient * CFenetre::getPatient ( )
```

Accesseur

Renvoie

```
CFenetre::m_patient
```

4.2.3 Documentation des données membres

4.2.3.1 m_controle

```
CControle* CFenetre::m_controle [private]
```

Widget qui gère les commandes de l'application

4.2.3.2 m_image

```
CImage* CFenetre::m_image [private]
```

Widget qui gère l'image

4.2.3.3 m_patient

```
CPatient* CFenetre::m_patient [private]
```

Widget qui gère les informations du patient

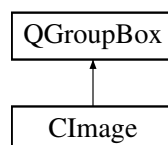
La documentation de cette classe a été générée à partir du fichier suivant :

- CFenetre.h
- CFenetre.cpp

4.3 Référence de la classe CImage

```
#include <CImage.h>
```

Graphe d'héritage de CImage:



Connecteurs publics

- void `updateCoef` ()
- void `updateFormatHeight` (QString text)
- void `updateRatioPercent` (int ratio)
- void `updateAreaLabel` ()

Signaux

- void `formatOrRatioChanged` ()
- void `coefChanged` ()

Fonctions membres publiques

- `CImage` (QWidget *parent=nullptr)
- `CPaint` * `getPaint` ()
- QString `getArea` ()

Fonctions membres privées

- QString `getAreaLabel` ()
- int `getFormatIntFromText` (QString text)

Attributs privés

- QPushButton * `m_boutonLoad`
- QPushButton * `m_boutonFermer`
- QPushButton * `m_boutonEffacerD`
- QPushButton * `m_boutonEffacerT`
- QSpinBox * `m_ratioSB`
- QComboBox * `m_formatCB`
- QLabel * `m_areaLabel`
- `CPaint` * `m_paint`
- float `m_coef`
- int `m_formatHeight`
- int `m_ratioPercent`
- const QString `c_areaLabel1` = "Aire :"
- const QString `c_areaLabel2` = "cm2"

4.3.1 Description détaillée

Widget qui gère la partie "image" de l'application.

On peut découper ce Widget en deux parties :

- une partie "paint" qui contient une image et les dessins que l'utilisateur peut faire, entièrement contenue dans le membre `m_paint`;
- une partie qui contrôle les actions possible sur la partie précédente et un label qui imprime l'aire du polygone.

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 CImage()

```
CImage::CImage (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur par défaut

Paramètres

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

4.3.3 Documentation des fonctions membres

4.3.3.1 coefChanged

```
void CImage::coefChanged ( ) [signal]
```

Signal lorsque le coefficient a changé

Voir également

[CImage::updateCoef](#), [CImage::updateAreaLabel](#)

4.3.3.2 formatOrRatioChanged

```
void CImage::formatOrRatioChanged ( ) [signal]
```

Signal lorsque le format ou bien le ratio a changé

Voir également

[CImage::updateFormatHeight](#), [CImage::updateRatioPercent](#), [CImage::updateCoef](#)

4.3.3.3 getArea()

```
QString CImage::getArea ( )
```

Renvoie

Le texte contenu dans le label [CImage::m_areaLabel](#)

Voir également

[CImage::m_areaLabel](#)

4.3.3.4 getAreaLabel()

```
QString CImage::getAreaLabel ( ) [private]
```

Retourne la chaîne de caractère a inclure dans `m_areaLabel`

Renvoie

Une chaîne de caractère du type `Aire : 1.2 cm2`

Voir également

[CImage::m_areaLabel](#)

4.3.3.5 getFormatIntFromText()

```
int CImage::getFormatIntFromText (
    QString text ) [private]
```

Retourne la hauteur du format, en cm

Paramètres

<i>text</i>	Le format en text, <i>e.g.</i> 18x24
-------------	--------------------------------------

Renvoie

La hauteur en cm, *e.g* 24

4.3.3.6 getPaint()

```
CPaint * CImage::getPaint ( )
```

Accesseur

Renvoie

[CImage::m_paint](#)

4.3.3.7 updateAreaLabel

```
void CImage::updateAreaLabel ( ) [slot]
```

Slot qui actualise le label qui imprime l'aire

Voir également

[CImage::m_areaLabel](#), [CImage::getAreaLabel](#)

4.3.3.8 updateCoef

```
void CImage::updateCoef ( ) [slot]
```

Slot qui actualise le coefficient

Voir également

[CImage::m_coef](#), [CImage::coefChanged](#)

4.3.3.9 updateFormatHeight

```
void CImage::updateFormatHeight (
    QString text ) [slot]
```

Slot qui actualise la hauteur du format

Voir également

[CImage::m_formatHeight](#), [CImage::getFormatIntFromText](#), [CImage::formatOrRatioChanged](#)

4.3.3.10 updateRatioPercent

```
void CImage::updateRatioPercent (
    int ratio ) [slot]
```

Slot qui actualise le ratio `m_ratio`

Paramètres

<i>ratio</i>	La valeur à actualiser, en pourcent
--------------	-------------------------------------

Voir également

[CImage::m_ratioPercent](#), [CImage::formatOrRatioChanged](#)

4.3.4 Documentation des données membres

4.3.4.1 c_areaLabel1

```
const QString CImage::c_areaLabel1 = "Aire : " [private]
```

Variable constante qui définit une partie du label qui affiche l'aire

Voir également

[c_areaLabel2](#)

4.3.4.2 c_areaLabel2

```
const QString CImage::c_areaLabel2 = "cm2" [private]
```

Variable constante qui définit une partie du label qui affiche l'aire

Voir également

[c_areaLabel1](#)

4.3.4.3 m_areaLabel

```
QLabel* CImage::m_areaLabel [private]
```

Label qui imprime l'aire du polygone

Voir également

[CImage::updateAreaLabel](#), [CImage::getArea](#)

4.3.4.4 m_boutonEffacerD

```
QPushButton* CImage::m_boutonEffacerD [private]
```

Bouton pour effacer le dernier point tracé

Voir également

[CPaint::removeLast](#)

4.3.4.5 m_boutonEffacerT

```
QPushButton* CImage::m_boutonEffacerT [private]
```

Bouton pour effacer tout le tracé

Voir également

[CPaint::clearImage](#)

4.3.4.6 m_boutonFermer

```
QPushButton* CImage::m_boutonFermer [private]
```

Bouton pour fermer le polygone

Voir également

[CPaint::closePoly](#)

4.3.4.7 m_boutonLoad

```
QPushButton* CImage::m_boutonLoad [private]
```

Bouton pour charger une image

Voir également

[CPaint::openDialog](#)

4.3.4.8 m_coef

```
float CImage::m_coef [private]
```

Coefficient qui permet de traduire des pixels en cm dans le monde réel

Le coefficient c est calculé suivant

$$c := \frac{h_F}{h_I} \cdot \frac{100}{r}, \quad (4.1)$$

avec h_F la hauteur du format (en cm), h_I la hauteur de l'image (en pixels), r le ratio (en pourcent).

Voir également

[CImage::updateCoef](#), [CImage::coefChanged](#)

4.3.4.9 m_formatCB

```
QComboBox* CImage::m_formatCB [private]
```

ComboBox qui définit le format de notre image radiologique

Les formats possibles sont (en cm) :

- 18x24
- 24x30
- 36x43

Voir également

[m_formatHeight](#)

4.3.4.10 m_formatHeight

```
int CImage::m_formatHeight [private]
```

Hauteur de l'image radiologique, en cm

Voir également

[CImage::m_formatCB](#)

4.3.4.11 m_paint

```
CPaint* CImage::m_paint [private]
```

Objet qui gère la partie image et dessine

Voir également

[CImage::getPaint](#)

4.3.4.12 m_ratioPercent

```
int CImage::m_ratioPercent [private]
```

Ratio de l'image radiologique, en pourcent

Voir également

[CImage::m_ratioSB](#)

4.3.4.13 m_ratioSB

```
QSpinBox* CImage::m_ratioSB [private]
```

SpinBox qui définit le ratio de notre image radiologique

Le ratio d'une image radiologique est le coefficient appliqué pour traduire une longueur sur l'image en longueur dans le monde réel.

Voir également

[m_ratioPercent](#)

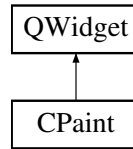
La documentation de cette classe a été générée à partir du fichier suivant :

- CImage.h
- CImage.cpp

4.4 Référence de la classe CPaint

```
#include <CPaint.h>
```

Graphe d'héritage de CPaint:



Connecteurs publics

- void [openDialog](#) ()
- void [clearImage](#) ()
- void [closePoly](#) ()
- void [removelast](#) ()
- void [openDialogSave](#) ()
- void [drawID](#) ()

Signaux

- void [areaPxChanged](#) ()
- void [imageHeightChanged](#) ()

Fonctions membres publiques

- [CPaint](#) (QWidget *parent=nullptr)
- int [getAreaPx](#) ()
- int [getImageHeight](#) ()

Fonctions membres protégées

- void [paintEvent](#) (QPaintEvent *event) override
- void [mousePressEvent](#) (QMouseEvent *event) override

Fonctions membres privées

- void [draw](#) ()
- void [updateAreaPx](#) ()
- int [computeAreaPxFromPoints](#) (QVector< QPoint > *vec)
- void [updateImageHeight](#) ()

Attributs privés

- QImage * [m_image](#)
- QString [m_fileName](#)
- QVector< QPoint > * [m_liPoint](#)
- int [m_areaPx](#)
- int [m_imageHeight](#)
- const int [c_paintWidth](#) = 400
- const int [c_paintHeight](#) = 400

4.4.1 Description détaillée

Widget qui affiche l'image et permet à l'utilisateur de dessiner dessus.

4.4.2 Documentation des constructeurs et destructeur

4.4.2.1 CPaint()

```
CPaint::CPaint (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur par défaut

Paramètres

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

4.4.3 Documentation des fonctions membres

4.4.3.1 areaPxChanged

```
void CPaint::areaPxChanged ( ) [signal]
```

Signal émit lorsque [CPaint::m_areaPx](#) a changé

Voir également

[CPaint::m_areaPx](#), [CPaint::updateAreaPx](#)

4.4.3.2 clearImage

```
void CPaint::clearImage ( ) [slot]
```

Slot qui efface les dessins sur l'image

Voir également

[CImage::m_boutonEffacerT](#).

4.4.3.3 closePoly

```
void CPaint::closePoly ( ) [slot]
```

Slot qui ferme le polygone

Voir également

[CImage::m_boutonFermer](#), [CPaint::updateAreaPx](#).

4.4.3.4 computeAreaPxFromPoints()

```
int CPaint::computeAreaPxFromPoints (
    QVector< QPoint > * vec ) [private]
```

Calcule l'aire d'un polygone, en pixels

On note les coordonnées des sommets du polygone (dans l'ordre) $(x_0, y_0), (x_1, y_1) \dots (x_{n-1}, y_{n-1})$. L'aire du polygone est alors

$$A = \frac{1}{2} \sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i, \quad (4.2)$$

où $x_0 = x_n$ et $y_0 = y_n$.

Paramètres

vec	Un vecteur de QPoint, qui définit le polygone
-----	---

Renvoie

L'aire du polygone en pixels

4.4.3.5 draw()

```
void CPaint::draw ( ) [private]
```

Dessine les points et lignes sur l'image

Généralement appelée lorsque l'utilisateur clique sur l'image. Dessine le dernier point de [CPaint::m_liPoint](#) (celui qui vient d'être défini par le clique) et une ligne entre le dernier et avant-dernier point.

Voir également

[CPaint::mousePressEvent](#)

4.4.3.6 drawID

```
void CPaint::drawID ( ) [slot]
```

Slot pour imprimer les informations du patient sur l'image

Voir également

[CControle::m_boutonPrint](#)

4.4.3.7 getAreaPx()

```
int CPaint::getAreaPx ( )
```

Accesseur

Renvoie

[CPaint::m_areaPx](#)

4.4.3.8 getImageHeight()

```
int CPaint::getImageHeight ( )
```

Accesseur

Renvoie

[CPaint::m_imageHeight](#)

4.4.3.9 imageHeightChanged

```
void CPaint::imageHeightChanged ( ) [signal]
```

Signal émit lorsque [CPaint::m_imageHeight](#) a changé

Voir également

[CPaint::m_imageHeight](#), [CPaint::updateImageHeight](#)

4.4.3.10 mousePressEvent()

```
void CPaint::mousePressEvent (
    QMouseEvent * event ) [override], [protected]
```

Evènement clique gauche

Si clique gauche, ajoute la position du clique au vecteur [CPaint::m_liPoint](#) puis dessine sur l'image avec [CPaint::draw](#)

Paramètres

<i>event</i>	Un mouse event
--------------	----------------

Voir également

[CPaint::m_liPoint](#), [CPaint::draw](#)

4.4.3.11 openDialog

```
void CPaint::openDialog ( ) [slot]
```

Slot qui charge une image dans [CPaint::m_image](#)

Voir également

[CImage::m_boutonLoad](#).

4.4.3.12 openDialogSave

```
void CPaint::openDialogSave ( ) [slot]
```

Slot pour sauvegarder l'image

Voir également

[CControle::m_boutonSave](#)

4.4.3.13 paintEvent()

```
void CPaint::paintEvent (
    QPaintEvent * event ) [override], [protected]
```

Re-dessine le Widget lorsqu'un paint event est reçu

Paramètres

<i>event</i>	Un paint event
--------------	----------------

Voir également

[CPaint::m_image](#)

4.4.3.14 removelast

```
void CPaint::removelast ( ) [slot]
```

Slot qui efface le dernier point et la ligne qui y est reliée.

Voir également

[CImage::m_boutonEffacerD](#).

4.4.3.15 updateAreaPx()

```
void CPaint::updateAreaPx ( ) [private]
```

Actualise l'attribut [CPaint::m_areaPx](#)

Si le polygone est assez grand (plus que deux points), calcule l'aire avec [CPaint::computeAreaPxFromPoints\(\)](#). Puis emet le signal [CPaint::areaPxChanged\(\)](#).

4.4.3.16 updateImageHeight()

```
void CPaint::updateImageHeight ( ) [private]
```

Actualise l'attribut [CPaint::m_imageHeight](#)

Voir également

[CPaint::imageHeightChanged](#)

4.4.4 Documentation des données membres

4.4.4.1 c_paintHeight

```
const int CPaint::c_paintHeight = 400 [private]
```

Hauteur du widget, fixée, en pixels

4.4.4.2 c_paintWidth

```
const int CPaint::c_paintWidth = 400 [private]
```

Largeur du widget, fixée, en pixels

4.4.4.3 m_areaPx

```
int CPaint::m_areaPx [private]
```

Aire du polygone, en pixels

Voir également

[CPaint::updateAreaPx](#), [CPaint::getAreaPx](#), [CPaint::areaPxChanged](#)

4.4.4.4 m_fileName

```
QString CPaint::m_fileName [private]
```

Le chemin vers l'image

4.4.4.5 m_image

```
QImage* CPaint::m_image [private]
```

L'image sur laquelle on dessine

4.4.4.6 m_imageHeight

```
int CPaint::m_imageHeight [private]
```

Hauteur de l'image, en pixels

Voir également

[CPaint::updateImageHeight](#), [CPaint::getImageHeight](#), [CPaint::imageHeightChanged](#)

4.4.4.7 m_liPoint

```
QVector<QPoint>* CPaint::m_liPoint [private]
```

Le vecteur de QPoint qui définit le polygone (sommets)

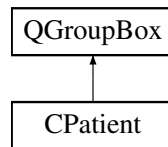
La documentation de cette classe a été générée à partir du fichier suivant :

- CPaint.h
- CPaint.cpp

4.5 Référence de la classe CPatient

```
#include <CPatient.h>
```

Graphe d'héritage de CPatient:



Fonctions membres publiques

- [CPatient](#) (QWidget *parent=nullptr)
- QString [getNom](#) ()
- QString [getPrenom](#) ()
- QString [getDate](#) ()

Attributs privés

- QLineEdit * [m_nom](#)
- QLineEdit * [m_prenom](#)
- QDateEdit * [m_date](#)

4.5.1 Description détaillée

Widget qui gère les informations du patient.

Les informations d'un patient sont son nom, son prénom et sa date de naissance.

4.5.2 Documentation des constructeurs et destructeur

4.5.2.1 CPatient()

```
CPatient::CPatient (
    QWidget * parent = nullptr ) [explicit]
```

Construit un widget enfant de parent.

Paramètres

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

4.5.3 Documentation des fonctions membres

4.5.3.1 getDate()

```
QString CPatient::getDate ( )
```

Renvoie

La date de naissance du patient

Voir également

[CPatient::m_date](#)

4.5.3.2 getNom()

```
QString CPatient::getNom ( )
```

Renvoie

Le nom du patient

Voir également

[CPatient::m_nom](#)

4.5.3.3 getPrenom()

```
QString CPatient::getPrenom ( )
```

Renvoie

Le prénom du patient

Voir également

[CPatient::m_prenom](#)

4.5.4 Documentation des données membres

4.5.4.1 m_date

```
QDateEdit* CPatient::m_date [private]
```

La date de naissance du patient

4.5.4.2 m_nom

```
QLineEdit* CPatient::m_nom [private]
```

Le nom du patient

4.5.4.3 m_prenom

```
QLineEdit* CPatient::m_prenom [private]
```

Le prénom du patient

La documentation de cette classe a été générée à partir du fichier suivant :

- CPatient.h
- CPatient.cpp

Index

- areaPxChanged
 - CPaint, [20](#)
- c_areaLabel1
 - CImage, [15](#)
- c_areaLabel2
 - CImage, [16](#)
- c_paintHeight
 - CPaint, [24](#)
- c_paintWidth
 - CPaint, [24](#)
- CControle, [7](#)
 - CControle, [8](#)
 - getBoutonPrint, [8](#)
 - getBoutonSave, [8](#)
 - m_boutonPrint, [9](#)
 - m_boutonQuit, [9](#)
 - m_boutonSave, [9](#)
 - quitApp, [8](#)
- CFenetre, [10](#)
 - getImage, [10](#)
 - getPatient, [10](#)
 - m_controle, [11](#)
 - m_image, [11](#)
 - m_patient, [11](#)
- CImage, [11](#)
 - c_areaLabel1, [15](#)
 - c_areaLabel2, [16](#)
 - CImage, [12](#)
 - coefChanged, [13](#)
 - formatOrRatioChanged, [13](#)
 - getArea, [13](#)
 - getAreaLabel, [13](#)
 - getFormatIntFromText, [14](#)
 - getPaint, [14](#)
 - m_areaLabel, [16](#)
 - m_boutonEffacerD, [16](#)
 - m_boutonEffacerT, [16](#)
 - m_boutonFermer, [16](#)
 - m_boutonLoad, [17](#)
 - m_coef, [17](#)
 - m_formatCB, [17](#)
 - m_formatHeight, [17](#)
 - m_paint, [18](#)
 - m_ratioPercent, [18](#)
 - m_ratioSB, [18](#)
 - updateAreaLabel, [14](#)
 - updateCoef, [14](#)
 - updateFormatHeight, [15](#)
 - updateRatioPercent, [15](#)
- clearImage
 - CPaint, [20](#)
- closePoly
 - CPaint, [20](#)
- coefChanged
 - CImage, [13](#)
- computeAreaPxFromPoints
 - CPaint, [21](#)
- CPaint, [19](#)
 - areaPxChanged, [20](#)
 - c_paintHeight, [24](#)
 - c_paintWidth, [24](#)
 - clearImage, [20](#)
 - closePoly, [20](#)
 - computeAreaPxFromPoints, [21](#)
 - CPaint, [20](#)
 - draw, [21](#)
 - drawID, [21](#)
 - getAreaPx, [22](#)
 - getImageHeight, [22](#)
 - imageHeightChanged, [22](#)
 - m_areaPx, [24](#)
 - m_fileName, [25](#)
 - m_image, [25](#)
 - m_imageHeight, [25](#)
 - m_liPoint, [25](#)
 - mousePressEvent, [22](#)
 - openDialog, [23](#)
 - openDialogSave, [23](#)
 - paintEvent, [23](#)
 - removelast, [23](#)
 - updateAreaPx, [24](#)
 - updateImageHeight, [24](#)
- CPatient, [26](#)
 - CPatient, [26](#)
 - getDate, [27](#)
 - getNom, [27](#)
 - getPrenom, [27](#)
 - m_date, [27](#)
 - m_nom, [28](#)
 - m_prenom, [28](#)
- draw
 - CPaint, [21](#)
- drawID
 - CPaint, [21](#)
- formatOrRatioChanged
 - CImage, [13](#)

- getArea
 - CIImage, 13
- getAreaLabel
 - CIImage, 13
- getAreaPx
 - CPaint, 22
- getBoutonPrint
 - CControle, 8
- getBoutonSave
 - CControle, 8
- getDate
 - CPatient, 27
- getFormatIntFromText
 - CIImage, 14
- getImage
 - CFenetre, 10
- getImageHeight
 - CPaint, 22
- getNom
 - CPatient, 27
- getPaint
 - CIImage, 14
- getPatient
 - CFenetre, 10
- getPrenom
 - CPatient, 27
- imageHeightChanged
 - CPaint, 22
- m_areaLabel
 - CIImage, 16
- m_areaPx
 - CPaint, 24
- m_boutonEffacerD
 - CIImage, 16
- m_boutonEffacerT
 - CIImage, 16
- m_boutonFermer
 - CIImage, 16
- m_boutonLoad
 - CIImage, 17
- m_boutonPrint
 - CControle, 9
- m_boutonQuit
 - CControle, 9
- m_boutonSave
 - CControle, 9
- m_coef
 - CIImage, 17
- m_controle
 - CFenetre, 11
- m_date
 - CPatient, 27
- m_fileName
 - CPaint, 25
- m_formatCB
 - CIImage, 17
- m_formatHeight
 - CIImage, 17
- m_image
 - CFenetre, 11
- m_imageHeight
 - CPaint, 25
- m_liPoint
 - CPaint, 25
- m_nom
 - CPatient, 28
- m_paint
 - CIImage, 18
- m_patient
 - CFenetre, 11
- m_prenom
 - CPatient, 28
- m_ratioPercent
 - CIImage, 18
- m_ratioSB
 - CIImage, 18
- mousePressEvent
 - CPaint, 22
- openDialog
 - CPaint, 23
- openDialogSave
 - CPaint, 23
- paintEvent
 - CPaint, 23
- quitApp
 - CControle, 8
- removelast
 - CPaint, 23
- updateAreaLabel
 - CIImage, 14
- updateAreaPx
 - CPaint, 24
- updateCoef
 - CIImage, 14
- updateFormatHeight
 - CIImage, 15
- updateImageHeight
 - CPaint, 24
- updateRatioPercent
 - CIImage, 15