

Yams

1.0

Généré par Doxygen 1.8.17



<b>1 Yams</b>	<b>1</b>
1.1 Introduction	1
1.2 Captures d'écran	1
<b>2 Index hiérarchique</b>	<b>3</b>
2.1 Hiérarchie des classes	3
<b>3 Index des classes</b>	<b>5</b>
3.1 Liste des classes	5
<b>4 Index des fichiers</b>	<b>7</b>
4.1 Liste des fichiers	7
<b>5 Documentation des classes</b>	<b>9</b>
5.1 Référence de la classe CDice	9
5.1.1 Description détaillée	9
5.1.2 Documentation des constructeurs et destructeur	10
5.1.2.1 CDice()	10
5.1.3 Documentation des fonctions membres	10
5.1.3.1 addDotRect()	10
5.1.3.2 getCount()	10
5.1.3.3 getDiceRect()	11
5.1.3.4 getDotRects()	11
5.1.3.5 setDiceRect()	11
5.1.4 Documentation des données membres	11
5.1.4.1 m_diceRect	11
5.1.4.2 m_IDotRects	12
5.2 Référence de la classe CDiceSet	12
5.2.1 Description détaillée	13
5.2.2 Documentation des constructeurs et destructeur	13
5.2.2.1 CDiceSet()	13
5.2.3 Documentation des fonctions membres	13
5.2.3.1 getAces()	14
5.2.3.2 getFives()	14
5.2.3.3 getFours()	14
5.2.3.4 getSixes()	15
5.2.3.5 getThrees()	15
5.2.3.6 getTotal()	16
5.2.3.7 getTwos()	17
5.2.3.8 operator!=(())	17
5.2.3.9 operator==(())	18
5.2.3.10 operator[]()	18
5.2.4 Documentation des données membres	18
5.2.4.1 m_iAs	19

5.2.4.2 m_iFives	19
5.2.4.3 m_iFours	19
5.2.4.4 m_iSixes	19
5.2.4.5 m_iThrees	19
5.2.4.6 m_iTwos	20
5.3 Référence de la classe QYamsStartFormWidget::CPlayerForm	20
5.3.1 Description détaillée	20
5.3.2 Documentation des constructeurs et destructeur	20
5.3.2.1 CPlayerForm()	20
5.3.3 Documentation des fonctions membres	21
5.3.3.1 getInputName()	21
5.3.3.2 getQWidget()	21
5.3.4 Documentation des données membres	21
5.3.4.1 m_HBoxLayout	21
5.3.4.2 m_iptPlayerName	22
5.3.4.3 m_lblPlayerNum	22
5.4 Référence de la classe QPlayerGridsWidget::CPlayerGridView	22
5.4.1 Description détaillée	23
5.4.2 Documentation des constructeurs et destructeur	23
5.4.2.1 CPlayerGridView()	23
5.4.2.2 ~CPlayerGridView()	23
5.4.3 Documentation des données membres	24
5.4.3.1 lblAs	24
5.4.3.2 lblBonus	24
5.4.3.3 lblBrelan	24
5.4.3.4 lblCarre	24
5.4.3.5 lblChance	25
5.4.3.6 lblFives	25
5.4.3.7 lblFours	25
5.4.3.8 lblFull	25
5.4.3.9 lblLargeStraight	25
5.4.3.10 lblPlayerName	26
5.4.3.11 lblPreTotalSup	26
5.4.3.12 lblSixes	26
5.4.3.13 lblSmallStraight	26
5.4.3.14 lblSuperYams	26
5.4.3.15 lblThrees	27
5.4.3.16 lblTotal	27
5.4.3.17 lblTotalInf	27
5.4.3.18 lblTotalSup	27
5.4.3.19 lblTwos	27
5.4.3.20 lblYams	28

5.5 Référence de la classe QAboutWidget . . . . .	28
5.5.1 Description détaillée . . . . .	29
5.5.2 Documentation des constructeurs et destructeur . . . . .	29
5.5.2.1 QAboutWidget() . . . . .	29
5.5.3 Documentation des fonctions membres . . . . .	30
5.5.3.1 showAboutOpenCVWindow . . . . .	30
5.5.3.2 showAboutQtWindow . . . . .	30
5.5.4 Documentation des données membres . . . . .	31
5.5.4.1 m_ui . . . . .	31
5.6 Référence de la classe QCameraWidget . . . . .	31
5.6.1 Description détaillée . . . . .	33
5.6.2 Documentation des constructeurs et destructeur . . . . .	33
5.6.2.1 QCameraWidget() . . . . .	33
5.6.2.2 ~QCameraWidget() . . . . .	34
5.6.3 Documentation des fonctions membres . . . . .	34
5.6.3.1 _findDices . . . . .	34
5.6.3.2 _findDicesByBlob() . . . . .	35
5.6.3.3 _findDicesByMinArea() . . . . .	35
5.6.3.4 _updateImage() . . . . .	36
5.6.3.5 cameraImgUpdated . . . . .	36
5.6.3.6 dicesUpdated . . . . .	36
5.6.3.7 onWrongDetection . . . . .	37
5.6.4 Documentation des données membres . . . . .	37
5.6.4.1 DEFAULT_KERNEL . . . . .	37
5.6.4.2 m_bIsWrongDetection . . . . .	38
5.6.4.3 m_bThreadsEnabled . . . . .	38
5.6.4.4 m_IDices . . . . .	38
5.6.4.5 m_matImageCaptured . . . . .	38
5.6.4.6 m_pLastDiceSet . . . . .	38
5.6.4.7 m_pLastValidatedDiceSet . . . . .	39
5.6.4.8 m_qTimerFrame . . . . .	39
5.6.4.9 m_ui . . . . .	39
5.6.4.10 m_videoCapture . . . . .	39
5.6.4.11 MAX_BLOB_AREA . . . . .	39
5.6.4.12 MAX_BLOB_THRESHOLD . . . . .	40
5.6.4.13 MAX_CANNY_THRESHOLD . . . . .	40
5.6.4.14 MAX_DICE_DOT_WIDTH . . . . .	40
5.6.4.15 MAX_DICE_WIDTH . . . . .	40
5.6.4.16 MIN_BLOB_AREA . . . . .	40
5.6.4.17 MIN_BLOB_THRESHOLD . . . . .	41
5.6.4.18 MIN_CANNY_THRESHOLD . . . . .	41
5.6.4.19 MIN_CIRCULARITY . . . . .	41

---

5.6.4.20 MIN_DICE_DOT_WIDTH . . . . .	41
5.6.4.21 MIN_DICE_WIDTH . . . . .	41
5.6.4.22 MIN_DISTANCE_FROM_DICE_CENTER . . . . .	42
5.6.4.23 MIN_INERTIA_RATIO . . . . .	42
5.6.4.24 SAME_POS_TOLERANCE . . . . .	42
5.7 Référence de la classe QEndGameWidget . . . . .	42
5.7.1 Description détaillée . . . . .	43
5.7.2 Documentation des constructeurs et destructeur . . . . .	43
5.7.2.1 QEndGameWidget() . . . . .	43
5.7.3 Documentation des fonctions membres . . . . .	44
5.7.3.1 getQuitButton() . . . . .	44
5.7.3.2 getRestartButton() . . . . .	44
5.7.4 Documentation des données membres . . . . .	45
5.7.4.1 m_ui . . . . .	45
5.8 Référence de la classe QPlayerGrid . . . . .	45
5.8.1 Description détaillée . . . . .	48
5.8.2 Documentation des constructeurs et destructeur . . . . .	48
5.8.2.1 QPlayerGrid() . . . . .	48
5.8.3 Documentation des fonctions membres . . . . .	49
5.8.3.1 _checkBonus() . . . . .	49
5.8.3.2 getAces() . . . . .	49
5.8.3.3 getBonus() . . . . .	50
5.8.3.4 getBrelan() . . . . .	50
5.8.3.5 getCarre() . . . . .	50
5.8.3.6 getChance() . . . . .	51
5.8.3.7 getFives() . . . . .	51
5.8.3.8 getFours() . . . . .	51
5.8.3.9 getFull() . . . . .	52
5.8.3.10 getLargeStraight() . . . . .	52
5.8.3.11 getLowerTotal() . . . . .	52
5.8.3.12 getPlayerName() . . . . .	53
5.8.3.13 getPreUpperTotal() . . . . .	53
5.8.3.14 getSixes() . . . . .	53
5.8.3.15 getSmallStraight() . . . . .	54
5.8.3.16 getSuperYams() . . . . .	54
5.8.3.17 getThrees() . . . . .	54
5.8.3.18 getTotal() . . . . .	55
5.8.3.19 getTwos() . . . . .	55
5.8.3.20 getUpperTotal() . . . . .	56
5.8.3.21 getYams() . . . . .	56
5.8.3.22 gridUpdated . . . . .	57
5.8.3.23 isAcesAlreadySet() . . . . .	58

---

5.8.3.24 isBrelanAlreadySet()	58
5.8.3.25 isCarreAlreadySet()	59
5.8.3.26 isChanceAlreadySet()	59
5.8.3.27 isFivesAlreadySet()	60
5.8.3.28 isFoursAlreadySet()	60
5.8.3.29 isFullAlreadySet()	61
5.8.3.30 isGridFinished()	61
5.8.3.31 isLargeStraightAlreadySet()	61
5.8.3.32 isSixesAlreadySet()	62
5.8.3.33 isSmallStraightAlreadySet()	62
5.8.3.34 isSuperYamsAlreadySet()	63
5.8.3.35 isThreesAlreadySet()	63
5.8.3.36 isTwosAlreadySet()	64
5.8.3.37 isYamsAlreadySet()	64
5.8.3.38 setAces()	65
5.8.3.39 setBrelan()	65
5.8.3.40 setCarre()	65
5.8.3.41 setChance()	65
5.8.3.42 setFives()	66
5.8.3.43 setFours()	66
5.8.3.44 setFull()	66
5.8.3.45 setLargeStraight()	66
5.8.3.46 setSixes()	67
5.8.3.47 setSmallStraight()	67
5.8.3.48 setSuperYams()	67
5.8.3.49 setThrees()	67
5.8.3.50 setTwos()	68
5.8.3.51 setYams()	68
5.8.4 Documentation des données membres	68
5.8.4.1 m_blsAcesAlreadySet	68
5.8.4.2 m_blsBonus	68
5.8.4.3 m_blsBrelanAlreadySet	69
5.8.4.4 m_blsCarreAlreadySet	69
5.8.4.5 m_blsChanceAlreadySet	69
5.8.4.6 m_blsFivesAlreadySet	69
5.8.4.7 m_blsFoursAlreadySet	69
5.8.4.8 m_blsFull	70
5.8.4.9 m_blsFullAlreadySet	70
5.8.4.10 m_blsLargeStraight	70
5.8.4.11 m_blsLargeStraightAlreadySet	70
5.8.4.12 m_blsSixesAlreadySet	70
5.8.4.13 m_blsSmallStraight	71

5.8.4.14 m_blsSmallStraightAlreadySet . . . . .	71
5.8.4.15 m_blsSuperYams . . . . .	71
5.8.4.16 m_blsSuperYamsAlreadySet . . . . .	71
5.8.4.17 m_blsThreesAlreadySet . . . . .	71
5.8.4.18 m_blsTwosAlreadySet . . . . .	72
5.8.4.19 m_blsYams . . . . .	72
5.8.4.20 m_blsYamsAlreadySet . . . . .	72
5.8.4.21 m_iAces . . . . .	72
5.8.4.22 m_iBrelan . . . . .	72
5.8.4.23 m_iCarre . . . . .	73
5.8.4.24 m_iChance . . . . .	73
5.8.4.25 m_iFives . . . . .	73
5.8.4.26 m_iFours . . . . .	73
5.8.4.27 m_iSixes . . . . .	73
5.8.4.28 m_iThrees . . . . .	74
5.8.4.29 m_iTwos . . . . .	74
5.8.4.30 m_sPlayerName . . . . .	74
5.9 Référence de la classe QPlayerGridsWidget . . . . .	74
5.9.1 Description détaillée . . . . .	76
5.9.2 Documentation des constructeurs et destructeur . . . . .	76
5.9.2.1 QPlayerGridsWidget() . . . . .	76
5.9.3 Documentation des fonctions membres . . . . .	76
5.9.3.1 actualizeActionButtons . . . . .	77
5.9.3.2 addGrid() . . . . .	79
5.9.3.3 enableActionButtons() . . . . .	80
5.9.3.4 getAcesButton() . . . . .	81
5.9.3.5 getBrelanButton() . . . . .	82
5.9.3.6 getCarreButton() . . . . .	82
5.9.3.7 getChanceButton() . . . . .	83
5.9.3.8 getFivesButton() . . . . .	83
5.9.3.9 getFoursButton() . . . . .	84
5.9.3.10 getFullButton() . . . . .	84
5.9.3.11 getLargeStraightButton() . . . . .	85
5.9.3.12 getSixesButton() . . . . .	85
5.9.3.13 getSmallStraightButton() . . . . .	86
5.9.3.14 getSuperYamsButton() . . . . .	86
5.9.3.15 getThreeButton() . . . . .	87
5.9.3.16 getTwosButton() . . . . .	87
5.9.3.17 getYamsButton() . . . . .	88
5.9.3.18 removeGrid() . . . . .	88
5.9.3.19 updateGrid . . . . .	89
5.9.4 Documentation des données membres . . . . .	90



5.9.4.1 m_mapPlayerGridViews	90
5.9.4.2 m_ui	90
5.10 Référence de la classe QYams	90
5.10.1 Description détaillée	92
5.10.2 Documentation des énumérations membres	93
5.10.2.1 EChoices	93
5.10.2.2 EYamsActions	93
5.10.3 Documentation des constructeurs et destructeur	94
5.10.3.1 QYams()	94
5.10.3.2 ~QYams()	95
5.10.4 Documentation des fonctions membres	95
5.10.4.1 _beforeStart()	96
5.10.4.2 _hideGameBar()	96
5.10.4.3 _nextPlayer()	97
5.10.4.4 _onEndGame()	98
5.10.4.5 _resetChoices()	98
5.10.4.6 _showRules()	99
5.10.4.7 _simulate()	99
5.10.4.8 closeEvent()	100
5.10.4.9 doAction	101
5.10.4.10 launchGame	102
5.10.4.11 playerUpdated	103
5.10.4.12 quit	104
5.10.4.13 restart	104
5.10.4.14 showAboutWindow	105
5.10.4.15 start	106
5.10.4.16 updateTurn	107
5.10.5 Documentation des données membres	108
5.10.5.1 m_choice1	108
5.10.5.2 m_choice2	108
5.10.5.3 m_choice3	108
5.10.5.4 m_iNbrTurn	108
5.10.5.5 m_itPlayerGrids	109
5.10.5.6 m_lpQPlayerGrids	109
5.10.5.7 m_mapLastSimulation	109
5.10.5.8 m_ptrAboutWindow	109
5.10.5.9 m_ptrEndGameWidget	109
5.10.5.10 m_ptrQCameraWidget	110
5.10.5.11 m_ptrQPlayerGridsWidget	110
5.10.5.12 m_ptrQYamsStartFrom	110
5.10.5.13 m_ui	110
5.10.5.14 m_YamsActionsNames	111

5.11 Référence de la classe QYamsStartFormWidget . . . . .	111
5.11.1 Description détaillée . . . . .	113
5.11.2 Documentation des constructeurs et destructeur . . . . .	113
5.11.2.1 QYamsStartFormWidget() . . . . .	113
5.11.3 Documentation des fonctions membres . . . . .	113
5.11.3.1 addPlayerNameInput . . . . .	114
5.11.3.2 onPlayerNameSetUp . . . . .	114
5.11.3.3 playerNameSetUp . . . . .	114
5.11.4 Documentation des données membres . . . . .	115
5.11.4.1 m_IPlayerNameInputs . . . . .	115
5.11.4.2 m_ui . . . . .	115
<b>6 Documentation des fichiers . . . . .</b>	<b>117</b>
6.1 Référence du fichier src/image_recognition/QCameraWidget.cpp . . . . .	117
6.2 Référence du fichier src/image_recognition/QCameraWidget.hpp . . . . .	117
6.3 Référence du fichier src/main.cpp . . . . .	118
6.3.1 Documentation des fonctions . . . . .	118
6.3.1.1 main() . . . . .	119
6.4 Référence du fichier src/models/CDice.cpp . . . . .	119
6.5 Référence du fichier src/models/CDice.hpp . . . . .	119
6.6 Référence du fichier src/models/CDiceSet.hpp . . . . .	121
6.6.1 Documentation des définitions de type . . . . .	121
6.6.1.1 uint . . . . .	121
6.7 Référence du fichier src/models/QPlayerGrid.cpp . . . . .	122
6.8 Référence du fichier src/models/QPlayerGrid.hpp . . . . .	122
6.9 Référence du fichier src/QAboutWidget.cpp . . . . .	123
6.10 Référence du fichier src/QAboutWidget.hpp . . . . .	123
6.11 Référence du fichier src/QEndGameWidget.cpp . . . . .	124
6.12 Référence du fichier src/QEndGameWidget.hpp . . . . .	125
6.13 Référence du fichier src/QPlayerGridsWidget.cpp . . . . .	126
6.14 Référence du fichier src/QPlayerGridsWidget.hpp . . . . .	126
6.15 Référence du fichier src/QYams.cpp . . . . .	127
6.16 Référence du fichier src/QYams.hpp . . . . .	128
6.17 Référence du fichier src/QYamsStartFormWidget.cpp . . . . .	128
6.18 Référence du fichier src/QYamsStartFormWidget.hpp . . . . .	129
<b>Index . . . . .</b>	<b>131</b>

# Chapitre 1

## Yams

Version

1.0

### 1.1 Introduction

Cette application est une adaptation du jeu de société Yams à la reconnaissance d'image.

Elle est réalisée avec le framework [Qt](#) pour la partie graphique et la bibliothèque [OpenCV](#) pour la reconnaissance d'image.

Le jeu prend la forme d'une fenêtre présentant les scores des différents joueurs sous la forme d'un tableau sur la droite, le rendu caméra sur la gauche et les meilleurs choix en bas de la fenêtre.

Si l'utilisateur ne trouve pas les choix proposés judicieux, il peut néanmoins choisir l'action qu'il souhaite faire en cliquant sur le bouton correspondant à l'action qu'il désire réaliser dans la colonne gauche du tableau des scores. Les boutons des actions déjà réalisées seront grisés.

Un rappel des règles du jeu se trouve dans le menu Aide - Règles du jeu.

Le Yams étant un jeu se jouant avec un nombre minimal de deux joueurs sans maximum, il est possible de jouer avec l'application avec autant de contraintes.

La reconnaissance d'image servant à détecter les dés peut ne pas être infallible. C'est pourquoi un bouton *Redé-tecter* a été disposé à droite des boutons représentant les meilleurs choix.

Pour un fonctionnement optimal de la reconnaissance d'image, merci de :

- Jouer sur une surface sombre ;
- Placer une caméra au dessus du jeu à 20 cm de hauteur ;
- Éviter les reflet et ombres ;
- Jouer dans un environnement lumineux ;
- Faire en sorte de mettre hors champ toute zone de mouvement en dehors du plateau de jeu ;
- Bien positionner les dés en dessous de l'objectif de la caméra ;
- Maintenir les dés à environ 1cm de distance les uns des autres.

### 1.2 Captures d'écran

- Menu de démarrage du jeu :
- Pendant le jeu :
- Fin de jeu :



## Chapitre 2

# Index hiérarchique

### 2.1 Hiérarchie des classes

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

CDice . . . . .	9
CDiceSet . . . . .	12
QYamsStartFormWidget::CPlayerForm . . . . .	20
QPlayerGridsWidget::CPlayerGridView . . . . .	22
QMainWindow	
QYams . . . . .	90
QObject	
QPlayerGrid . . . . .	45
QWidget	
QAboutWidget . . . . .	28
QCameraWidget . . . . .	31
QEndGameWidget . . . . .	42
QPlayerGridsWidget . . . . .	74
QYamsStartFormWidget . . . . .	111



## Chapitre 3

# Index des classes

### 3.1 Liste des classes

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

<a href="#">CDice</a>	Encapsule un dé détecté . . . . .	9
<a href="#">CDiceSet</a>	Encapsule un jeu de dés . . . . .	12
<a href="#">QYamsStartFormWidget::CPlayerForm</a>	Encapsule le numéro et le champ de texte pour un joueur . . . . .	20
<a href="#">QPlayerGridsWidget::CPlayerGridView</a>	Encapsule une vue de grille de jeu . . . . .	22
<a href="#">QAboutWidget</a>	Menu À propos de l'application . . . . .	28
<a href="#">QCameraWidget</a>	Encapsule la détection d'image et le rendu vidéo . . . . .	31
<a href="#">QEndGameWidget</a>	Menu de fin de partie . . . . .	42
<a href="#">QPlayerGrid</a>	Encapsule une grille de jeu . . . . .	45
<a href="#">QPlayerGridsWidget</a>	Vue des grilles de jeu . . . . .	74
<a href="#">QYams</a>	Contrôleur du jeu Yams . . . . .	90
<a href="#">QYamsStartFormWidget</a>	Fenêtre de saisie des noms de joueur . . . . .	111





## Chapitre 4

# Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/main.cpp	118
src/QAboutWidget.cpp	123
src/QAboutWidget.hpp	123
src/QEndGameWidget.cpp	124
src/QEndGameWidget.hpp	125
src/QPlayerGridsWidget.cpp	126
src/QPlayerGridsWidget.hpp	126
src/QYams.cpp	127
src/QYams.hpp	128
src/QYamsStartFormWidget.cpp	128
src/QYamsStartFormWidget.hpp	129
src/image_recognition/QCameraWidget.cpp	117
src/image_recognition/QCameraWidget.hpp	117
src/models/CDice.cpp	119
src/models/CDice.hpp	119
src/models/CDiceSet.hpp	121
src/models/QPlayerGrid.cpp	122
src/models/QPlayerGrid.hpp	122



# Chapitre 5

## Documentation des classes

### 5.1 Référence de la classe CDice

Encapsule un dé détecté

```
#include <CDice.hpp>
```

#### Fonctions membres publiques

- `CDice` (const cv::RotatedRect &diceRect)  
*Constructeur.*
- const cv::RotatedRect & `getDiceRect` () const  
*Retourne la position du rectangle du dé*
- void `setDiceRect` (const cv::RotatedRect &diceRect)  
*Définit la position du rectangle du dé*
- const std::list< cv::RotatedRect > & `getDotRects` () const  
*Retourne la liste des positions des points du dé*
- void `addDotRect` (const cv::RotatedRect &dotRect)  
*Définit la liste des positions des points du dé*
- uint `getCount` () const  
*Retourne le nombre de points détectés.*

#### Attributs privés

- cv::RotatedRect `m_diceRect`  
*Position du rectangle du dé*
- std::list< cv::RotatedRect > `m_IDotRects`  
*Liste des positions des points du dé*

#### 5.1.1 Description détaillée

Encapsule un dé détecté

Utile lors de la reconnaissance d'image

Auteur

Corentin VÉROT

Date

2020-05-15

Définition à la ligne 20 du fichier CDice.hpp.

## 5.1.2 Documentation des constructeurs et destructeur

### 5.1.2.1 CDice()

```
CDice::CDice (
    const cv::RotatedRect & diceRect )
```

Constructeur.

#### Paramètres

<i>diceRect</i>	: const cv::RotatedRect& - Position du rectangle du dé
-----------------	--

Définition à la ligne 3 du fichier CDice.cpp.

## 5.1.3 Documentation des fonctions membres

### 5.1.3.1 addDotRect()

```
void CDice::addDotRect (
    const cv::RotatedRect & dotRect ) [inline]
```

Définit la liste des positions des points du dé

#### Paramètres

<i>dotRect</i>	: std::list<cv::RotatedRect>& - Liste des positions des points du dé
----------------	--

Définition à la ligne 69 du fichier CDice.hpp.

### 5.1.3.2 getCount()

```
uint CDice::getCount ( ) const [inline]
```

Retourne le nombre de points détectés.

#### Renvoie

uint - Nombre de dés détectés

Définition à la ligne 76 du fichier CDice.hpp.

### 5.1.3.3 getDiceRect()

```
const cv::RotatedRect& CDice::getDiceRect ( ) const [inline]
```

Retourne la position du rectangle du dé

Renvoie

const cv::RotatedRect& - Position du rectangle du dé

Définition à la ligne 48 du fichier CDice.hpp.

### 5.1.3.4 getDotRects()

```
const std::list<cv::RotatedRect>& CDice::getDotRects ( ) const [inline]
```

Retourne la liste des positions des points du dé

Renvoie

std::list<cv::RotatedRect>& - Liste des positions des points du dé

Définition à la ligne 62 du fichier CDice.hpp.

### 5.1.3.5 setDiceRect()

```
void CDice::setDiceRect (
    const cv::RotatedRect & diceRect ) [inline]
```

Définit la position du rectangle du dé

Paramètres

<i>diceRect</i>	: const cv::RotatedRect& - Position du rectangle du dé
-----------------	--

Définition à la ligne 55 du fichier CDice.hpp.

## 5.1.4 Documentation des données membres

### 5.1.4.1 m\_diceRect

```
cv::RotatedRect CDice::m_diceRect [private]
```

Position du rectangle du dé

Définition à la ligne 28 du fichier CDice.hpp.

#### 5.1.4.2 m\_IDotRects

```
std::list<cv::RotatedRect> CDice::m_IDotRects [private]
```

Liste des positions des points du dé

Définition à la ligne 33 du fichier CDice.hpp.

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

- src/models/[CDice.hpp](#)
- src/models/[CDice.cpp](#)

## 5.2 Référence de la classe CDiceSet

Encapsule un jeu de dés.

```
#include <CDiceSet.hpp>
```

### Fonctions membres publiques

- [CDiceSet](#) ()  
*Constructeur.*
- [uint getAces](#) () const  
*Retourne le score donné par les as.*
- [uint getTwos](#) () const  
*Retourne le score donné par les deux.*
- [uint getThrees](#) () const  
*Retourne le score donné par les trois.*
- [uint getFours](#) () const  
*Retourne le score donné par les quatre.*
- [uint getFives](#) () const  
*Retourne le score donné par les cinq.*
- [uint getSixes](#) () const  
*Retourne le score donné par les six.*
- [uint getTotal](#) () const  
*Retourne la somme totale du jeu.*
- [uint & operator\[\]](#) (uint iDiceNumber)  
*Permet d'accéder au nombre de dés par valeur.*
- const bool & [operator==](#) ([CDiceSet](#) &diceSet)  
*Teste si un set est égal à l'autre.*
- bool [operator!=](#) ([CDiceSet](#) &diceSet)  
*Teste si un set n'est pas égal à l'autre.*

## Attributs privés

- [uint m\\_iAs](#)  
*Nombre d'as dans le jeu de dés.*
- [uint m\\_iTwos](#)  
*Nombre de deux dans le jeu de dés.*
- [uint m\\_iThrees](#)  
*Nombre de trois dans le jeu de dés.*
- [uint m\\_iFours](#)  
*Nombre de quatre dans le jeu de dés.*
- [uint m\\_iFives](#)  
*Nombre de cinq dans le jeu de dés.*
- [uint m\\_iSixes](#)  
*Nombre de six dans le jeu de dés.*

### 5.2.1 Description détaillée

Encapsule un jeu de dés.

#### Auteur

Corentin VÉROT

#### Date

2020-05-15

Définition à la ligne 14 du fichier CDiceSet.hpp.

### 5.2.2 Documentation des constructeurs et destructeur

#### 5.2.2.1 CDiceSet()

```
CDiceSet::CDiceSet ( ) [inline]
```

Constructeur.

Définition à la ligne 131 du fichier CDiceSet.hpp.

### 5.2.3 Documentation des fonctions membres

### 5.2.3.1 getAces()

```
uint CDiceSet::getAces ( ) const [inline]
```

Retourne le score donné par les as.

Renvoie

uint

Définition à la ligne 58 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.2 getFives()

```
uint CDiceSet::getFives ( ) const [inline]
```

Retourne le score donné par les cinq.

Renvoie

uint

Définition à la ligne 86 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.3 getFours()

```
uint CDiceSet::getFours ( ) const [inline]
```

Retourne le score donné par les quatre.

Renvoie

uint

Définition à la ligne 79 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :





### 5.2.3.4 getSixes()

```
uint CDiceSet::getSixes ( ) const [inline]
```

Retourne le score donné par les six.

Renvoie

uint

Définition à la ligne 93 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.5 getThrees()

```
uint CDiceSet::getThrees ( ) const [inline]
```

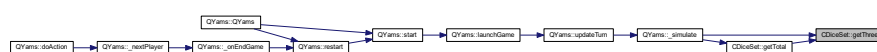
Retourne le score donné par les trois.

Renvoie

uint

Définition à la ligne 72 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.6 getTotal()

```
uint CDiceSet::getTotal ( ) const [inline]
```

Retourne la somme totale du jeu.

Renvoie

uint

Définition à la ligne 100 du fichier CDiceSet.hpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.2.3.7 getTwos()

```
uint CDiceSet::getTwos ( ) const [inline]
```

Retourne le score donné par les deux.

**Renvoie**

uint

Définition à la ligne 65 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.8 operator"!="()

```
bool CDiceSet::operator!= (
    CDiceSet & diceSet ) [inline]
```

Teste si un set n'est pas égal à l'autre.

**Paramètres**

<i>diceSet</i>	: <a href="#">CDiceSet&amp;</a> - Jeu de dés
----------------	--

**Renvoie**

const bool&

Définition à la ligne 127 du fichier CDiceSet.hpp.

Voici le graphe d'appel pour cette fonction :



### 5.2.3.9 operator==( )

```
const bool & CDiceSet::operator==(   
    CDiceSet & diceSet ) [inline]
```

Teste si un set est égal à l'autre.

#### Paramètres

<i>diceSet</i>	: <a href="#">CDiceSet</a> & - Jeu de dés
----------------	---

#### Renvoie

const bool&

Définition à la ligne 162 du fichier CDiceSet.hpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.10 operator[]( )

```
uint & CDiceSet::operator[] (   
    uint iDiceNumber ) [inline]
```

Permet d'accéder au nombre de dés par valeur.

#### Paramètres

<i>iDiceNumber</i>	- Numéro du dé
--------------------	----------------

#### Renvoie

uint& - Nombre de dés de cette valeur

Définition à la ligne 141 du fichier CDiceSet.hpp.

## 5.2.4 Documentation des données membres

#### 5.2.4.1 m\_iAs

```
uint CDiceSet::m_iAs [private]
```

Nombre d'as dans le jeu de dés.

Définition à la ligne 19 du fichier CDiceSet.hpp.

#### 5.2.4.2 m\_iFives

```
uint CDiceSet::m_iFives [private]
```

Nombre de cinq dans le jeu de dés.

Définition à la ligne 39 du fichier CDiceSet.hpp.

#### 5.2.4.3 m\_iFours

```
uint CDiceSet::m_iFours [private]
```

Nombre de quatre dans le jeu de dés.

Définition à la ligne 34 du fichier CDiceSet.hpp.

#### 5.2.4.4 m\_iSixes

```
uint CDiceSet::m_iSixes [private]
```

Nombre de six dans le jeu de dés.

Définition à la ligne 44 du fichier CDiceSet.hpp.

#### 5.2.4.5 m\_iThrees

```
uint CDiceSet::m_iThrees [private]
```

Nombre de trois dans le jeu de dés.

Définition à la ligne 29 du fichier CDiceSet.hpp.

#### 5.2.4.6 m\_iTwos

```
uint CDiceSet::m_iTwos [private]
```

Nombre de deux dans le jeu de dés.

Définition à la ligne 24 du fichier CDiceSet.hpp.

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

— src/models/CDiceSet.hpp

### 5.3 Référence de la classe QYamsStartFormWidget::CPlayerForm

Encapsule le numéro et le champ de texte pour un joueur.

#### Fonctions membres publiques

- [CPlayerForm](#) ([QYamsStartFormWidget](#) \*parent, [uint](#) iPlayerNumber)  
*Constructeur.*
- [QString](#) [getInputName](#) () const  
*Retourne le nom saisi par l'utilisateur.*
- [QHBoxLayout](#) \* [getQWidget](#) () const  
*Retourne le layout.*

#### Attributs privés

- [QHBoxLayout](#) \* [m\\_HBoxLayout](#)  
*Layout horizontal.*
- [QLabel](#) \* [m\\_lblPlayerNum](#)  
*Label visant affichant le numéro du joueur.*
- [QLineEdit](#) \* [m\\_iptPlayerName](#)  
*Champ de texte visant à accueillir le nom du joueur.*

#### 5.3.1 Description détaillée

Encapsule le numéro et le champ de texte pour un joueur.

Définition à la ligne 32 du fichier QYamsStartFormWidget.hpp.

#### 5.3.2 Documentation des constructeurs et destructeur

##### 5.3.2.1 CPlayerForm()

```
QYamsStartFormWidget::CPlayerForm::CPlayerForm (  
    QYamsStartFormWidget * parent,  
    uint iPlayerNumber )
```

Constructeur.

## Paramètres

<i>parent</i>	: QYamsStartFormWidget*
<i>iPlayerNumber</i>	: uint - Numéro du joueur

Définition à la ligne 5 du fichier QYamsStartFormWidget.cpp.

### 5.3.3 Documentation des fonctions membres

#### 5.3.3.1 getInputName()

```
QString QYamsStartFormWidget::CPlayerForm::getInputName ( ) const [inline]
```

Retourne le nom saisi par l'utilisateur.

## Renvoie

QString

Définition à la ligne 65 du fichier QYamsStartFormWidget.hpp.

#### 5.3.3.2 getQWidget()

```
QHBoxLayout* QYamsStartFormWidget::CPlayerForm::getQWidget ( ) const [inline]
```

Retourne le layout.

## Renvoie

QHBoxLayout\*

Définition à la ligne 72 du fichier QYamsStartFormWidget.hpp.

### 5.3.4 Documentation des données membres

#### 5.3.4.1 m\_HBoxLayout

```
QHBoxLayout* QYamsStartFormWidget::CPlayerForm::m_HBoxLayout [private]
```

Layout horizontal.

Contient un label à gauche et un champ de texte à droite

Définition à la ligne 39 du fichier QYamsStartFormWidget.hpp.

### 5.3.4.2 m\_iptPlayerName

`QLineEdit* QYamsStartFormWidget::CPlayerForm::m_iptPlayerName [private]`

Champ de texte visant à accueillir le nom du joueur.

Définition à la ligne 49 du fichier `QYamsStartFormWidget.hpp`.

### 5.3.4.3 m\_lblPlayerNum

`QLabel* QYamsStartFormWidget::CPlayerForm::m_lblPlayerNum [private]`

Label visant affichant le numéro du joueur.

Définition à la ligne 44 du fichier `QYamsStartFormWidget.hpp`.

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

- [src/QYamsStartFormWidget.hpp](#)
- [src/QYamsStartFormWidget.cpp](#)

## 5.4 Référence de la classe `QPlayerGridsWidget::CPlayerGridView`

Encapsule une vue de grille de jeu.

### Fonctions membres publiques

- [CPlayerGridView](#) ([QPlayerGridsWidget](#) \*parent, const `QString` &sPlayerName)  
*Constructeur.*
- [~CPlayerGridView](#) ()

### Attributs publics

- `QLabel` \* [lblPlayerName](#)  
*Affichage nom du joueur.*
- `QLabel` \* [lblAs](#)  
*Affichage de As.*
- `QLabel` \* [lblTwos](#)  
*Affichage des deux.*
- `QLabel` \* [lblThrees](#)  
*Affichage des trois.*
- `QLabel` \* [lblFours](#)  
*Affichage des quatre.*
- `QLabel` \* [lblFives](#)  
*Affichage cinq.*
- `QLabel` \* [lblSixes](#)  
*Affichage six.*
- `QLabel` \* [lblPreTotalSup](#)  
*Affichage du total avant ajout du bonus.*
- `QLabel` \* [lblBonus](#)  
*Affichage du bonus obtenu.*
- `QLabel` \* [lblTotalSup](#)  
*Affichage du total de la partie supérieure.*
- `QLabel` \* [lblBrelan](#)



- QLabel \* [lblCarre](#)  
*Affichage du Brelan.*
- QLabel \* [lblFull](#)  
*Affichage du Carré*
- QLabel \* [lblSmallStraight](#)  
*Affichage du Full.*
- QLabel \* [lblLargeStraight](#)  
*Affichage de la petite suite.*
- QLabel \* [lblYams](#)  
*Affichage de la grande suite.*
- QLabel \* [lblSuperYams](#)  
*Affichage du Yams.*
- QLabel \* [lblChance](#)  
*Affichage du super Yams.*
- QLabel \* [lblTotalInf](#)  
*Affichage de la chance.*
- QLabel \* [lblTotal](#)  
*Affichage du total de la partie inférieure.*
- QLabel \* [lblTotal](#)  
*Affichage du total global.*

### 5.4.1 Description détaillée

Encapsule une vue de grille de jeu.

Structure de données visant à simplifier la construction de l'affichage des grilles de jeu

Définition à la ligne 34 du fichier QPlayerGridsWidget.hpp.

### 5.4.2 Documentation des constructeurs et destructeur

#### 5.4.2.1 CPlayerGridView()

```
QPlayerGridsWidget::CPlayerGridView::CPlayerGridView (
    QPlayerGridsWidget * parent,
    const QString & sPlayerName )
```

Constructeur.

Instancie tous les pointeurs des données membres

Paramètres

<i>parent</i>	: QPlayerGridsWidget* - Widget parent
<i>sPlayerName</i>	: const QString& - Nom du joueur

Définition à la ligne 150 du fichier QPlayerGridsWidget.cpp.

#### 5.4.2.2 ~CPlayerGridView()

```
QPlayerGridsWidget::CPlayerGridView::~CPlayerGridView ( )
```

Définition à la ligne 197 du fichier QPlayerGridsWidget.cpp.

### 5.4.3 Documentation des données membres

#### 5.4.3.1 lblAs

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblAs
```

Affichage de As.

Définition à la ligne 45 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.2 lblBonus

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblBonus
```

Affichage du bonus obtenu.

Définition à la ligne 73 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.3 lblBrelan

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblBrelan
```

Affichage du Brelan.

Définition à la ligne 82 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.4 lblCarre

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblCarre
```

Affichage du Carré

Définition à la ligne 86 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.5 lblChance

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblChance
```

Affichage de la chance.

Définition à la ligne 110 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.6 lblFives

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblFives
```

Affichage cinq.

Définition à la ligne 61 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.7 lblFours

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblFours
```

Affichage des quatre.

Définition à la ligne 57 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.8 lblFull

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblFull
```

Affichage du Full.

Définition à la ligne 90 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.9 lblLargeStraight

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblLargeStraight
```

Affichage de la grande suite.

Définition à la ligne 98 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.10 lblPlayerName

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblPlayerName
```

Affichage nom du joueur.

Définition à la ligne 40 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.11 lblPreTotalSup

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblPreTotalSup
```

Affichage du total avant ajout du bonus.

Définition à la ligne 69 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.12 lblSixes

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblSixes
```

Affichage six.

Définition à la ligne 65 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.13 lblSmallStraight

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblSmallStraight
```

Affichage de la petite suite.

Définition à la ligne 94 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.14 lblSuperYams

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblSuperYams
```

Affichage du super Yams.

Définition à la ligne 106 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.15 lblThrees

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblThrees
```

Affichage des trois.

Définition à la ligne 53 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.16 lblTotal

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblTotal
```

Affichage du total global.

Définition à la ligne 119 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.17 lblTotalInf

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblTotalInf
```

Affichage du total de la partie inférieure.

Définition à la ligne 114 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.18 lblTotalSup

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblTotalSup
```

Affichage du total de la partie supérieure.

Définition à la ligne 77 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.19 lblTwos

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblTwos
```

Affichage des deux.

Définition à la ligne 49 du fichier QPlayerGridsWidget.hpp.

#### 5.4.3.20 lblYams

```
QLabel* QPlayerGridsWidget::CPlayerGridView::lblYams
```

Affichage du Yams.

Définition à la ligne 102 du fichier QPlayerGridsWidget.hpp.

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

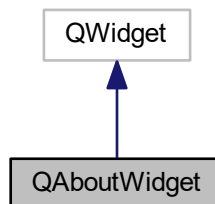
- [src/QPlayerGridsWidget.hpp](#)
- [src/QPlayerGridsWidget.cpp](#)

## 5.5 Référence de la classe QAboutWidget

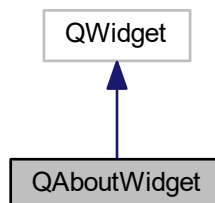
Menu À propos de l'application.

```
#include <QAboutWidget.hpp>
```

Graphe d'héritage de QAboutWidget:



Graphe de collaboration de QAboutWidget:



## Connecteurs publics

- void [showAboutQtWindow](#) ()  
*Affiche la fenêtre "About Qt".*
- void [showAboutOpenCVWindow](#) ()  
*Crée et affiche une fenêtre "About OpenCV".*

## Fonctions membres publiques

- [QAboutWidget](#) (QWidget \*parent=Q\_NULLPTR)  
*Constructeur.*

## Attributs privés

- Ui::QAboutWidget [m\\_ui](#)  
*Interface utilisateur / vue.*

### 5.5.1 Description détaillée

Menu À propos de l'application.

#### Auteur

Corentin VÉROT

#### Date

2020-05-15

Définition à la ligne 24 du fichier QAboutWidget.hpp.

### 5.5.2 Documentation des constructeurs et destructeur

#### 5.5.2.1 QAboutWidget()

```
QAboutWidget::QAboutWidget (
    QWidget * parent = Q_NULLPTR )
```

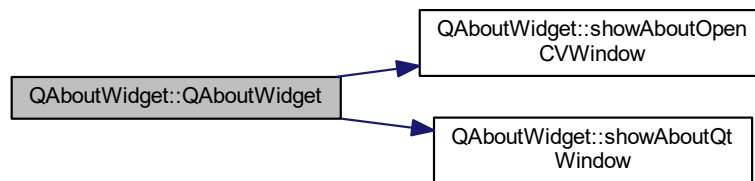
Constructeur.

#### Paramètres

<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR
---------------	--

Définition à la ligne 3 du fichier QAboutWidget.cpp.

Voici le graphe d'appel pour cette fonction :



### 5.5.3 Documentation des fonctions membres

#### 5.5.3.1 showAboutOpenCVWindow

```
void QAboutWidget::showAboutOpenCVWindow ( ) [slot]
```

Crée et affiche une fenêtre "About OpenCV".

Cette fenêtre est inspirée par la fenêtre "About Qt"

Définition à la ligne 22 du fichier `QAboutWidget.cpp`.

Voici le graphe des appelants de cette fonction :



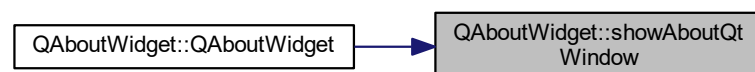
#### 5.5.3.2 showAboutQtWindow

```
void QAboutWidget::showAboutQtWindow ( ) [slot]
```

Affiche la fenêtre "About Qt".

Définition à la ligne 17 du fichier `QAboutWidget.cpp`.

Voici le graphe des appelants de cette fonction :





## 5.5.4 Documentation des données membres

### 5.5.4.1 m\_ui

```
Ui::QAboutWidget QAboutWidget::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 40 du fichier QAboutWidget.hpp.

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

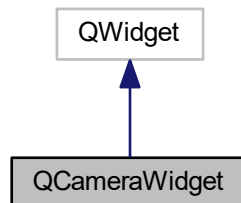
- [src/QAboutWidget.hpp](#)
- [src/QAboutWidget.cpp](#)

## 5.6 Référence de la classe QCameraWidget

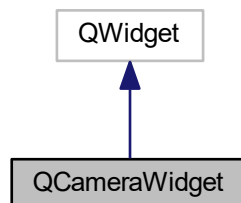
Encapsule la détection d'image et le rendu vidéo.

```
#include <QCameraWidget.hpp>
```

Graphe d'héritage de QCameraWidget:



Graphe de collaboration de QCameraWidget:



## Connecteurs publics

- void `onWrongDetection` ()  
*Callback lorsque l'utilisateur signale une erreur de détection.*

## Signaux

- void `cameraImgUpdated` ()  
*Signale que l'image de la caméra a changé*
- void `dicesUpdated` (`CDiceSet` &diceSet, bool isDetectionCorrection)  
*Signale que le jeu de dés a été mis à jour.*

## Fonctions membres publiques

- `QCameraWidget` (`QWidget *parent=Q_NULLPTR`)  
*Constructeur.*
- `~QCameraWidget` ()  
*Destructeur.*

## Connecteurs privés

- void `_findDices` ()  
*Lance l'analyse de l'image et emmet un jeu de dés si l'analyse est probante.*

## Fonctions membres privées

- void `_updateImage` ()  
*Met à jour l'image.*
- void `_findDicesByMinArea` (`std::list< std::shared_ptr< CDice >> &IDetectedDices`, `size_t &iNDetectedDices`) const  
*Traitement d'image par aire minimum.*
- void `_findDicesByBlob` (`size_t &iNDetectedDices`) const  
*Traitement d'image par simple blob.*

## Attributs privés

- const `uint MIN_DICE_WIDTH` = 1000  
*Largeur minimum d'un dé*
- const `uint MAX_DICE_WIDTH` = 4000  
*Largeur maximum d'un dé*
- const `uint MIN_DICE_DOT_WIDTH` = 10  
*Largeur minimum d'un point d'un dé*
- const `uint MAX_DICE_DOT_WIDTH` = 200  
*Largeur maximum d'un point d'un dé*
- const `cv::Size DEFAULT_KERNEL` = `cv::Size(3, 3)`  
*Noyau par défaut utilisé pour les opérations sur image.*
- const `uint MIN_CANNY_THRESHOLD` = 210  
*Seuil minimum pour le seuillage de Canny.*
- const `uint MAX_CANNY_THRESHOLD` = 255  
*Seuil maximum pour le seuillage de Canny.*
- const `uint MIN_DISTANCE_FROM_DICE_CENTER` = 7  
*Distance minimale depuis le centre du dé*
- const `uint MIN_BLOB_THRESHOLD` = 100  
*Seuil minimum pour le seuillage.*
- const `uint MAX_BLOB_THRESHOLD` = 255  
*Seuil maximum pour le seuillage.*
- const `uint MIN_BLOB_AREA` = 5

- Aire minimum.*
- const uint MAX\_BLOB\_AREA = 200
- Aire maximum.*
- const float MIN\_CIRCULARITY = 0.3
- TODO.*
- const float MIN\_INERTIA\_RATIO = 0.5
- TODO.*
- const float SAME\_POS\_TOLERANCE = 10
- Tolérance d'écart entre la position d'un même dé sur deux images différentes.*
- Ui::QCameraWidget m\_ui
- Interface utilisateur / vue.*
- cv::VideoCapture m\_videoCapture
- Capture vidéo.*
- std::unique\_ptr< cv::Mat > m\_matImageCaptured = nullptr
- Dernière image capturée.*
- QTimer m\_qTimerFrame
- Chrono.*
- std::list< std::shared\_ptr< CDice > > m\_IDices
- Liste des dés détectés.*
- std::shared\_ptr< CDiceSet > m\_pLastDiceSet = nullptr
- Jeu de dés nécessitant d'être confirmé par une seconde analyse.*
- std::shared\_ptr< CDiceSet > m\_pLastValidatedDiceSet = nullptr
- Dernier jeu de dés validé*
- bool m\_blsWrongDetection = false
- Est-ce que la dernière détection contenait une erreur ?*
- bool m\_bThreadsEnabled = false
- Active ou désactive le traitement d'image sur deux cœurs.*

### 5.6.1 Description détaillée

Encapsule la détection d'image et le rendu vidéo.

#### Auteur

Corentin VÉROT

#### Date

2020-05-15

Définition à la ligne 36 du fichier QCameraWidget.hpp.

### 5.6.2 Documentation des constructeurs et destructeur

#### 5.6.2.1 QCameraWidget()

```
QCameraWidget::QCameraWidget (
    QWidget * parent = Q_NULLPTR )
```

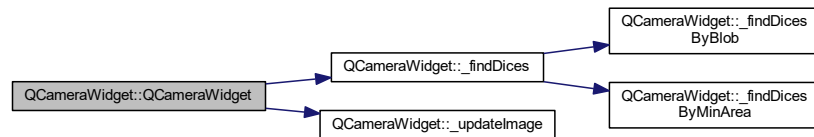
Constructeur.

#### Paramètres

<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR
---------------	--

Définition à la ligne 6 du fichier QCameraWidget.cpp.

Voici le graphe d'appel pour cette fonction :



### 5.6.2.2 ~QCameraWidget()

```
QCameraWidget::~QCameraWidget ( )
```

Destructeur.

Définition à la ligne 25 du fichier QCameraWidget.cpp.

## 5.6.3 Documentation des fonctions membres

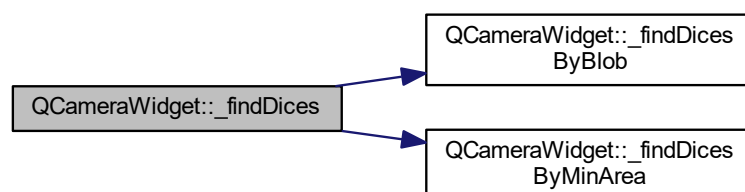
### 5.6.3.1 \_findDices

```
void QCameraWidget::_findDices ( ) [private], [slot]
```

Lance l'analyse de l'image et emmet un jeu de dés si l'analyse est probante.

Définition à la ligne 78 du fichier QCameraWidget.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.6.3.2 \_findDicesByBlob()

```
void QCameraWidget::_findDicesByBlob (
    size_t & iNDetectedDices ) const [private]
```

Traitement d'image par simple blob.

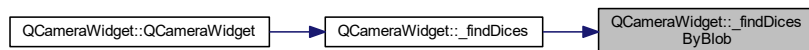
Un blob / goutte, est un groupe de pixels connectés partageant une même propriété au sein d'une image (ici les points des dés)

#### Paramètres

<i>iNDetectedDices</i>	: size_t& - Nombre de dés détectés
------------------------	------------------------------------

Définition à la ligne 270 du fichier QCameraWidget.cpp.

Voici le graphe des appelants de cette fonction :



### 5.6.3.3 \_findDicesByMinArea()

```
void QCameraWidget::_findDicesByMinArea (
    std::list< std::shared_ptr< CDice >> & lDetectedDices,
    size_t & iNDetectedDices ) const [private]
```

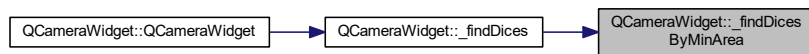
Traitement d'image par aire minimum.

#### Paramètres

<i>lDetectedDices</i>	: std::list<std::shared_ptr<CDice>>& - Liste des dés détectés
<i>iNDetectedDices</i>	: size_t& - Nombre de dés détectés

Définition à la ligne 182 du fichier QCameraWidget.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.6.3.4 \_updateImage()

```
void QCameraWidget::_updateImage ( ) [private]
```

Met à jour l'image.

Définition à la ligne 37 du fichier QCameraWidget.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.6.3.5 cameraImgUpdated

```
void QCameraWidget::cameraImgUpdated ( ) [signal]
```

Signale que l'image de la caméra a changé

Voici le graphe des appelants de cette fonction :



#### 5.6.3.6 dicesUpdated

```
void QCameraWidget::dicesUpdated (
    CDiceSet & diceSet,
    bool isDetectionCorrection ) [signal]
```

Signale que le jeu de dés a été mis à jour.

## Paramètres

<i>diceSet</i>	: <a href="#">CDiceSet</a> & - Nouveau jeu de dés
<i>isDetectionCorrection</i>	: bool - Vrai s'il s'agit d'une correction de détection

Voici le graphe des appelants de cette fonction :



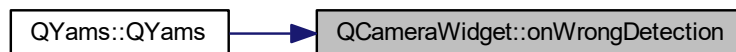
## 5.6.3.7 onWrongDetection

```
void QCameraWidget::onWrongDetection ( ) [slot]
```

Callback lorsque l'utilisateur signale une erreur de détection.

Définition à la ligne 30 du fichier QCameraWidget.cpp.

Voici le graphe des appelants de cette fonction :



## 5.6.4 Documentation des données membres

## 5.6.4.1 DEFAULT\_KERNEL

```
const cv::Size QCameraWidget::DEFAULT_KERNEL = cv::Size(3, 3) [private]
```

Noyau par défaut utilisé pour les opérations sur image.

Définition à la ligne 64 du fichier QCameraWidget.hpp.

#### 5.6.4.2 m\_bIsWrongDetection

```
bool QCameraWidget::m_bIsWrongDetection = false [private]
```

Est-ce que la dernière détection contenait une erreur ?

Définition à la ligne 159 du fichier QCameraWidget.hpp.

#### 5.6.4.3 m\_bThreadsEnabled

```
bool QCameraWidget::m_bThreadsEnabled = false [private]
```

Active ou désactive le traitement d'image sur deux cœurs.

L'analyse par aire minimum sera faite sur un cœurs et celle par Blob sur l'autre.

Définition à la ligne 166 du fichier QCameraWidget.hpp.

#### 5.6.4.4 m\_lDices

```
std::list<std::shared_ptr<CDice> > QCameraWidget::m_lDices [private]
```

Liste des dés détectés.

Définition à la ligne 142 du fichier QCameraWidget.hpp.

#### 5.6.4.5 m\_matImageCaptured

```
std::unique_ptr<cv::Mat> QCameraWidget::m_matImageCaptured = nullptr [private]
```

Dernière image capturée.

Définition à la ligne 130 du fichier QCameraWidget.hpp.

#### 5.6.4.6 m\_pLastDiceSet

```
std::shared_ptr<CDiceSet> QCameraWidget::m_pLastDiceSet = nullptr [private]
```

Jeu de dés nécessitant d'être confirmé par une seconde analyse.

La seconde analyse doit donner un set identique pour valider

Définition à la ligne 149 du fichier QCameraWidget.hpp.



#### 5.6.4.7 m\_pLastValidatedDiceSet

```
std::shared_ptr<CDiceSet> QCameraWidget::m_pLastValidatedDiceSet = nullptr [private]
```

Dernier jeu de dés validé

Définition à la ligne 154 du fichier QCameraWidget.hpp.

#### 5.6.4.8 m\_qTimerFrame

```
QTimer QCameraWidget::m_qTimerFrame [private]
```

Chrono.

Sert à actualiser l'image toutes les 20ms et à lancer l'analyse

Définition à la ligne 137 du fichier QCameraWidget.hpp.

#### 5.6.4.9 m\_ui

```
Ui::QCameraWidget QCameraWidget::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 120 du fichier QCameraWidget.hpp.

#### 5.6.4.10 m\_videoCapture

```
cv::VideoCapture QCameraWidget::m_videoCapture [private]
```

Capture vidéo.

Définition à la ligne 125 du fichier QCameraWidget.hpp.

#### 5.6.4.11 MAX\_BLOB\_AREA

```
const uint QCameraWidget::MAX_BLOB_AREA = 200 [private]
```

Aire maximum.

Définition à la ligne 99 du fichier QCameraWidget.hpp.

#### 5.6.4.12 MAX\_BLOB\_THRESHOLD

```
const uint QCameraWidget::MAX_BLOB_THRESHOLD = 255 [private]
```

Seuil maximum pour le seuillage.

Définition à la ligne 89 du fichier QCameraWidget.hpp.

#### 5.6.4.13 MAX\_CANNY\_THRESHOLD

```
const uint QCameraWidget::MAX_CANNY_THRESHOLD = 255 [private]
```

Seuil maximum pour le seuillage de Canny.

Définition à la ligne 74 du fichier QCameraWidget.hpp.

#### 5.6.4.14 MAX\_DICE\_DOT\_WIDTH

```
const uint QCameraWidget::MAX_DICE_DOT_WIDTH = 200 [private]
```

Largeur maximum d'un point d'un dé

Définition à la ligne 59 du fichier QCameraWidget.hpp.

#### 5.6.4.15 MAX\_DICE\_WIDTH

```
const uint QCameraWidget::MAX_DICE_WIDTH = 4000 [private]
```

Largeur maximum d'un dé

Définition à la ligne 49 du fichier QCameraWidget.hpp.

#### 5.6.4.16 MIN\_BLOB\_AREA

```
const uint QCameraWidget::MIN_BLOB_AREA = 5 [private]
```

Aire minimum.

Définition à la ligne 94 du fichier QCameraWidget.hpp.

#### 5.6.4.17 MIN\_BLOB\_THRESHOLD

```
const uint QCameraWidget::MIN_BLOB_THRESHOLD = 100 [private]
```

Seuil minimum pour le seuillage.

Définition à la ligne 84 du fichier QCameraWidget.hpp.

#### 5.6.4.18 MIN\_CANNY\_THRESHOLD

```
const uint QCameraWidget::MIN_CANNY_THRESHOLD = 210 [private]
```

Seuil minimum pour le seuillage de Canny.

Définition à la ligne 69 du fichier QCameraWidget.hpp.

#### 5.6.4.19 MIN\_CIRCULARITY

```
const float QCameraWidget::MIN_CIRCULARITY = 0.3 [private]
```

TODO.

Définition à la ligne 104 du fichier QCameraWidget.hpp.

#### 5.6.4.20 MIN\_DICE\_DOT\_WIDTH

```
const uint QCameraWidget::MIN_DICE_DOT_WIDTH = 10 [private]
```

Largeur minimum d'un point d'un dé

Définition à la ligne 54 du fichier QCameraWidget.hpp.

#### 5.6.4.21 MIN\_DICE\_WIDTH

```
const uint QCameraWidget::MIN_DICE_WIDTH = 1000 [private]
```

Largeur minimum d'un dé

Définition à la ligne 44 du fichier QCameraWidget.hpp.

#### 5.6.4.22 MIN\_DISTANCE\_FROM\_DICE\_CENTER

```
const uint QCameraWidget::MIN_DISTANCE_FROM_DICE_CENTER = 7 [private]
```

Distance minimale depuis le centre du dé

Définition à la ligne 79 du fichier QCameraWidget.hpp.

#### 5.6.4.23 MIN\_INERTIA\_RATIO

```
const float QCameraWidget::MIN_INERTIA_RATIO = 0.5 [private]
```

TODO.

Définition à la ligne 109 du fichier QCameraWidget.hpp.

#### 5.6.4.24 SAME\_POS\_TOLERANCE

```
const float QCameraWidget::SAME_POS_TOLERANCE = 10 [private]
```

Tolérance d'écart entre la position d'un même dé sur deux images différentes.

Définition à la ligne 114 du fichier QCameraWidget.hpp.

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

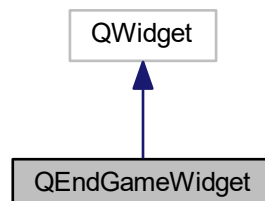
- [src/image\\_recognition/QCameraWidget.hpp](#)
- [src/image\\_recognition/QCameraWidget.cpp](#)

## 5.7 Référence de la classe QEndGameWidget

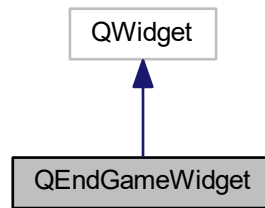
Menu de fin de partie.

```
#include <QEndGameWidget.hpp>
```

Graphe d'héritage de QEndGameWidget:



Graphe de collaboration de QEndGameWidget:



### Fonctions membres publiques

- [QEndGameWidget](#) (const QString &sWinnerName, QWidget \*parent=Q\_NULLPTR)  
*Constructeur.*
- const QPushButton \* [getQuitButton](#) () const  
*Retourne le bouton quitter.*
- const QPushButton \* [getRestartButton](#) () const  
*Retourne le bouton redémarrer.*

### Attributs privés

- Ui::QEndGameWidget [m\\_ui](#)  
*Interface utilisateur / vue.*

#### 5.7.1 Description détaillée

Menu de fin de partie.

Auteur

Corentin VÉROT

Date

2020-05-15

Définition à la ligne 19 du fichier QEndGameWidget.hpp.

#### 5.7.2 Documentation des constructeurs et destructeur

##### 5.7.2.1 QEndGameWidget()

```
QEndGameWidget::QEndGameWidget (  
    const QString & sWinnerName,  
    QWidget * parent = Q_NULLPTR )
```

Constructeur.

## Paramètres

<i>sWinnerName</i>	: QString& - Nom du joueur
<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR

Définition à la ligne 3 du fichier QEndGameWidget.cpp.

### 5.7.3 Documentation des fonctions membres

#### 5.7.3.1 getQuitButton()

```
const QPushButton* QEndGameWidget::getQuitButton ( ) const [inline]
```

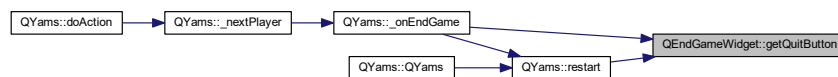
Retourne le bouton quitter.

## Renvoie

const QPushButton\*

Définition à la ligne 37 du fichier QEndGameWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.7.3.2 getRestartButton()

```
const QPushButton* QEndGameWidget::getRestartButton ( ) const [inline]
```

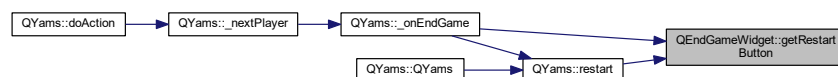
Retourne le bouton redémarrer.

## Renvoie

const QPushButton\*

Définition à la ligne 44 du fichier QEndGameWidget.hpp.

Voici le graphe des appelants de cette fonction :



## 5.7.4 Documentation des données membres

### 5.7.4.1 m\_ui

```
Ui::QEndGameWidget QEndGameWidget::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 50 du fichier QEndGameWidget.hpp.

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

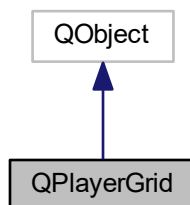
- [src/QEndGameWidget.hpp](#)
- [src/QEndGameWidget.cpp](#)

## 5.8 Référence de la classe QPlayerGrid

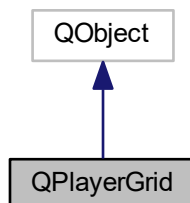
Encapsule une grille de jeu.

```
#include <QPlayerGrid.hpp>
```

Graphe d'héritage de QPlayerGrid:



Graphe de collaboration de QPlayerGrid:



## Signaux

- void `gridUpdated` (QPlayerGrid \*playerGrid)  
*Signale que la grille a été mise à jour.*

## Fonctions membres publiques

- QPlayerGrid (const QString &sPlayerName)  
*Constructeur.*
- QString `getPlayerName` () const  
*Retourne le nom du joueur.*
- const uint & `getAces` () const  
*Retourne la valeur des As.*
- const bool & `isAcesAlreadySet` () const  
*Retourne vrai si les as ont été défini.*
- void `setAces` (uint iValue)  
*Défini la valeur des As.*
- const uint & `getTwos` () const  
*Retourne la valeur des deux.*
- const bool & `isTwosAlreadySet` () const  
*Retourne vrai si les deux ont été défini.*
- void `setTwos` (uint iValue)  
*Défini la valeur des deux.*
- const uint & `getThrees` () const  
*Retourne la valeur des trois.*
- const bool & `isThreesAlreadySet` () const  
*Retourne vrai si les trois ont été défini.*
- void `setThrees` (uint iValue)  
*Défini la valeur des trois.*
- const uint & `getFours` () const  
*Retourne la valeur des quatre.*
- const bool & `isFoursAlreadySet` () const  
*Retourne vrai si les quatre ont été défini.*
- void `setFours` (uint iValue)  
*Défini la valeur des quatre.*
- const uint & `getFives` () const  
*Retourne la valeur des cinq.*
- const bool & `isFivesAlreadySet` () const  
*Retourne vrai si les cinq ont été défini.*
- void `setFives` (uint iValue)  
*Défini la valeur des cinq.*
- const uint & `getSixes` () const  
*Retourne la valeur des six.*
- const bool & `isSixesAlreadySet` () const  
*Retourne vrai si les six ont été défini.*
- void `setSixes` (uint iValue)  
*Défini la valeur des six.*
- const uint & `getBonus` () const  
*Retourne le bonus s'il y a.*
- const uint & `getBrelan` () const  
*Retourne la valeur du Brelan.*
- const bool & `isBrelanAlreadySet` () const  
*Retourne vrai si le Brelan a été défini.*
- void `setBrelan` (uint iValue)  
*Défini la valeur du Brelan.*
- const uint & `getCarre` () const  
*Retourne la valeur du Carré*
- const bool & `isCarreAlreadySet` () const  
*Retourne vrai si le Carré a été défini.*
- void `setCarre` (uint iValue)  
*Défini la valeur du Carré*
- const uint & `getFull` () const  
*Retourne la valeur du Full.*
- const bool & `isFullAlreadySet` () const  
*Retourne vrai si le Full a été défini.*
- void `setFull` (bool blsFull)  
*Défini s'il y a Full.*



- const [uint](#) & [getSmallStraight](#) () const  
*Retourne la valeur de la petite suite.*
- const bool & [isSmallStraightAlreadySet](#) () const  
*Retourne vrai si la petite suite a été défini.*
- void [setSmallStraight](#) (bool blsSmallStraight)  
*Défini s'il y a petite suite.*
- const [uint](#) & [getLargeStraight](#) () const  
*Retourne la valeur de la grande suite.*
- const bool & [isLargeStraightAlreadySet](#) () const  
*Retourne vrai si la grande suite a été défini.*
- void [setLargeStraight](#) (bool blsLargeStraight)  
*Défini s'il y a grande suite.*
- const [uint](#) & [getYams](#) () const  
*Retourne la valeur du Yams.*
- const bool & [isYamsAlreadySet](#) () const  
*Retourne vrai si le Yams a été défini.*
- void [setYams](#) (bool blsYams)  
*Défini s'il y a Yams.*
- const [uint](#) & [getSuperYams](#) () const  
*Retourne la valeur du super Yams.*
- const bool & [isSuperYamsAlreadySet](#) () const  
*Retourne vrai si le super Yams a été défini.*
- void [setSuperYams](#) (bool blsSuperYams)  
*Défini s'il y a super Yams.*
- const [uint](#) & [getChance](#) () const  
*Retourne la valeur de la chance.*
- const bool & [isChanceAlreadySet](#) () const  
*Retourne vrai si la chance a été défini.*
- void [setChance](#) (uint iChance)  
*Défini la valeur de la chance.*
- [uint](#) [getPreUpperTotal](#) () const
- [uint](#) [getUpperTotal](#) () const
- [uint](#) [getLowerTotal](#) () const
- [uint](#) [getTotal](#) () const
- bool [isGridFinished](#) () const

## Fonctions membres privées

- void [\\_checkBonus](#) ()  
*Vérifie pour la partie supérieure si le joueur a obtenu le bonus ou non.*

## Attributs privés

- QString [m\\_sPlayerName](#)  
*Nom du joueur.*
- [uint](#) [m\\_iAces](#)  
*Valeur des As.*
- bool [m\\_blsAcesAlreadySet](#)  
*Vrai si les as ont déjà été définis.*
- [uint](#) [m\\_iTwos](#)  
*Valeur des deux.*
- bool [m\\_blsTwosAlreadySet](#)  
*Vrai si les deux ont déjà été définis.*
- [uint](#) [m\\_iThrees](#)  
*Valeur des trois.*
- bool [m\\_blsThreesAlreadySet](#)  
*Vrai si les trois ont déjà été définis.*
- [uint](#) [m\\_iFours](#)  
*Valeur des quatre.*
- bool [m\\_blsFoursAlreadySet](#)  
*Vrai si les quatre ont déjà été définis.*
- [uint](#) [m\\_iFives](#)  
*Valeur des cinq.*
- bool [m\\_blsFivesAlreadySet](#)  
*Vrai si les cinq ont déjà été définis.*

- [uint m\\_iSixes](#)  
*Valeur des six.*
- [bool m\\_blsSixesAlreadySet](#)  
*Vrai si les six ont déjà été définis.*
- [bool m\\_blsBonus](#)  
*Vrai si le bonus est acquis.*
- [uint m\\_iBrelan](#)  
*Valeur du Brelan.*
- [bool m\\_blsBrelanAlreadySet](#)  
*Vrai si le Brelan a déjà été défini.*
- [uint m\\_iCarre](#)  
*Valeur du Carré*
- [bool m\\_blsCarreAlreadySet](#)  
*Vrai si le Carré a déjà été défini.*
- [bool m\\_blsFull](#)  
*Vrai s'il y a Full.*
- [bool m\\_blsFullAlreadySet](#)  
*Vrai si le Full a déjà été défini.*
- [bool m\\_blsSmallStraight](#)  
*Vrai s'il y a la petite suite.*
- [bool m\\_blsSmallStraightAlreadySet](#)  
*Vrai si la petite suite a déjà été défini.*
- [bool m\\_blsLargeStraight](#)  
*Vrai s'il y a la grande suite.*
- [bool m\\_blsLargeStraightAlreadySet](#)  
*Vrai si la grande suite a déjà été défini.*
- [bool m\\_blsYams](#)  
*Vrai s'il y a le Yams.*
- [bool m\\_blsYamsAlreadySet](#)  
*Vrai si le Yams a déjà été défini.*
- [bool m\\_blsSuperYams](#)  
*Vrai s'il y a le super Yams.*
- [bool m\\_blsSuperYamsAlreadySet](#)  
*Vrai si le super Yams a déjà été défini.*
- [uint m\\_iChance](#)  
*Valeur de la chance.*
- [bool m\\_blsChanceAlreadySet](#)  
*Vrai si la chance a déjà été défini.*

### 5.8.1 Description détaillée

Encapsule une grille de jeu.

Auteur

Corentin VÉROT

Date

2020-05-15

Définition à la ligne 18 du fichier QPlayerGrid.hpp.

### 5.8.2 Documentation des constructeurs et destructeur

#### 5.8.2.1 QPlayerGrid()

```
QPlayerGrid::QPlayerGrid (
    const QString & sPlayerName )
```

Constructeur.

## Paramètres

<i>sPlayerName</i>	: QString& - Nom du joueur
--------------------	----------------------------

Définition à la ligne 5 du fichier QPlayerGrid.cpp.

### 5.8.3 Documentation des fonctions membres

#### 5.8.3.1 `_checkBonus()`

```
void QPlayerGrid::_checkBonus ( ) [private]
```

Vérifie pour la partie supérieure si le joueur a obtenu le bonus ou non.

Définition à la ligne 293 du fichier QPlayerGrid.cpp.

#### 5.8.3.2 `getAces()`

```
const uint& QPlayerGrid::getAces ( ) const [inline]
```

Retourne la valeur des As.

## Renvoie

const uint&

Définition à la ligne 196 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.3 getBonus()

```
const uint& QPlayerGrid::getBonus ( ) const [inline]
```

Retourne le bonus s'il y a.

Renvoie

const uint&

Définition à la ligne 310 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.4 getBrelan()

```
const uint& QPlayerGrid::getBrelan ( ) const [inline]
```

Retourne la valeur du Bralan.

Renvoie

const uint&

Définition à la ligne 318 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.5 getCarre()

```
const uint& QPlayerGrid::getCarre ( ) const [inline]
```

Retourne la valeur du Carré

Renvoie

const uint&

Définition à la ligne 337 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.6 getChance()

```
const uint& QPlayerGrid::getChance ( ) const [inline]
```

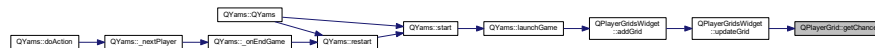
Retourne la valeur de la chance.

Renvoie

const uint&

Définition à la ligne 451 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.7 getFives()

```
const uint& QPlayerGrid::getFives ( ) const [inline]
```

Retourne la valeur des cinq.

Renvoie

const uint&

Définition à la ligne 272 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.8 getFours()

```
const uint& QPlayerGrid::getFours ( ) const [inline]
```

Retourne la valeur des quatre.

Renvoie

const uint&

Définition à la ligne 253 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.9 getFull()

```
const uint& QPlayerGrid::getFull ( ) const [inline]
```

Retourne la valeur du Full.

Renvoie

const uint&

Définition à la ligne 356 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.10 getLargeStraight()

```
const uint& QPlayerGrid::getLargeStraight ( ) const [inline]
```

Retourne la valeur de la grande suite.

Renvoie

const uint&

Définition à la ligne 394 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.11 getLowerTotal()

```
uint QPlayerGrid::getLowerTotal ( ) const
```

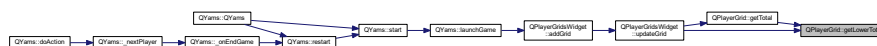
Retourne le total de la partie inférieure

Renvoie

uint

Définition à la ligne 258 du fichier QPlayerGrid.cpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.12 getPlayerName()

```
QString QPlayerGrid::getPlayerName ( ) const [inline]
```

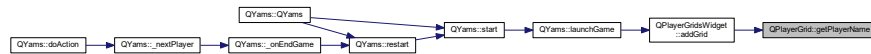
Retourne le nom du joueur.

Renvoie

QString - Nom du joueur

Définition à la ligne 188 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.13 getPreUpperTotal()

```
uint QPlayerGrid::getPreUpperTotal ( ) const
```

Retourne le pré-total de la partie supérieure (sans le bonus)

Renvoie

uint

Définition à la ligne 246 du fichier QPlayerGrid.cpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.14 getSixes()

```
const uint& QPlayerGrid::getSixes ( ) const [inline]
```

Retourne la valeur des six.

Renvoie

const uint&

Définition à la ligne 291 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.15 getSmallStraight()

```
const uint& QPlayerGrid::getSmallStraight ( ) const [inline]
```

Retourne la valeur de la petite suite.

Renvoie

const uint&

Définition à la ligne 375 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.16 getSuperYams()

```
const uint& QPlayerGrid::getSuperYams ( ) const [inline]
```

Retourne la valeur du super Yams.

Renvoie

const uint&

Définition à la ligne 432 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.17 getThrees()

```
const uint& QPlayerGrid::getThrees ( ) const [inline]
```

Retourne la valeur des trois.

Renvoie

const uint&

Définition à la ligne 234 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :





### 5.8.3.18 getTotal()

```
uint QPlayerGrid::getTotal ( ) const [inline]
```

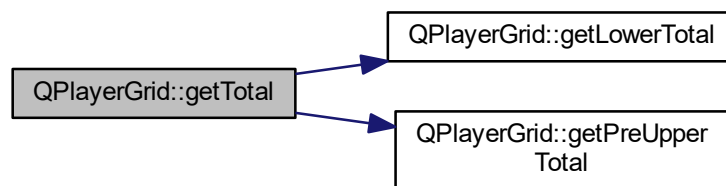
Retourne le total de la partie

Renvoie

uint

Définition à la ligne 493 du fichier QPlayerGrid.hpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.8.3.19 getTwos()

```
const uint& QPlayerGrid::getTwos ( ) const [inline]
```

Retourne la valeur des deux.

Renvoie

const uint&

Définition à la ligne 215 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.20 getUpperTotal()

```
uint QPlayerGrid::getUpperTotal ( ) const [inline]
```

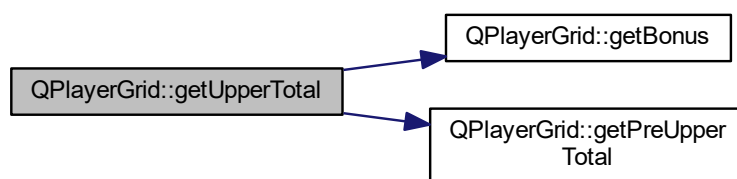
Retourne le total de la partie supérieure en contant le bonus si besoin est

Renvoie

uint

Définition à la ligne 479 du fichier QPlayerGrid.hpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.8.3.21 getYams()

```
const uint& QPlayerGrid::getYams ( ) const [inline]
```

Retourne la valeur du Yams.

Renvoie

const uint&

Définition à la ligne 413 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.22 gridUpdated

```
void QPlayerGrid::gridUpdated (  
    QPlayerGrid * playerGrid ) [signal]
```

Signale que la grille a été mise à jour.

## Paramètres

<i>playerGrid</i>	: QPlayerGrid*
-------------------	----------------

Voici le graphe des appelants de cette fonction :



### 5.8.3.23 isAcesAlreadySet()

```
const bool& QPlayerGrid::isAcesAlreadySet ( ) const [inline]
```

Retourne vrai si les as ont été défini.

#### Renvoie

const bool&

Définition à la ligne 202 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.24 isBrelanAlreadySet()

```
const bool& QPlayerGrid::isBrelanAlreadySet ( ) const [inline]
```

Retourne vrai si le Brelan a été défini.

#### Renvoie

const bool&

Définition à la ligne 324 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.25 isCarreAlreadySet()

```
const bool& QPlayerGrid::isCarreAlreadySet ( ) const [inline]
```

Retourne vrai si le Carré a été défini.

Renvoie

const bool&

Définition à la ligne 343 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.26 isChanceAlreadySet()

```
const bool& QPlayerGrid::isChanceAlreadySet ( ) const [inline]
```

Retourne vrai si la chance a été défini.

Renvoie

const bool&

Définition à la ligne 457 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.27 isFivesAlreadySet()

```
const bool& QPlayerGrid::isFivesAlreadySet ( ) const [inline]
```

Retourne vrai si les cinq ont été défini.

Renvoie

const bool&

Définition à la ligne 278 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.28 isFoursAlreadySet()

```
const bool& QPlayerGrid::isFoursAlreadySet ( ) const [inline]
```

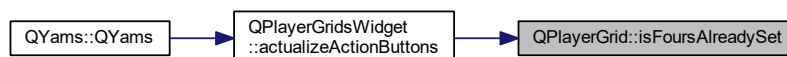
Retourne vrai si les quatre ont été défini.

Renvoie

const bool&

Définition à la ligne 259 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.29 isFullAlreadySet()

```
const bool& QPlayerGrid::isFullAlreadySet ( ) const [inline]
```

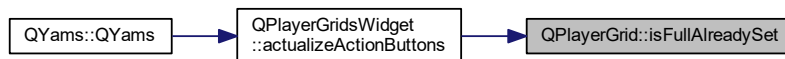
Retourne vrai si le Full a été défini.

Renvoie

const bool&

Définition à la ligne 362 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.30 isGridFinished()

```
bool QPlayerGrid::isGridFinished ( ) const
```

Retourne vrai si la grille est complétée

Renvoie

bool

Définition à la ligne 272 du fichier QPlayerGrid.cpp.

### 5.8.3.31 isLargeStraightAlreadySet()

```
const bool& QPlayerGrid::isLargeStraightAlreadySet ( ) const [inline]
```

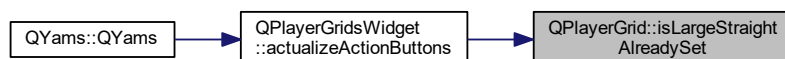
Retourne vrai si la grande suite a été défini.

Renvoie

const bool&

Définition à la ligne 400 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.32 isSixesAlreadySet()

```
const bool& QPlayerGrid::isSixesAlreadySet ( ) const [inline]
```

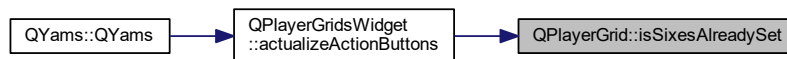
Retourne vrai si les six ont été défini.

#### Renvoie

const bool&

Définition à la ligne 297 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



### 5.8.3.33 isSmallStraightAlreadySet()

```
const bool& QPlayerGrid::isSmallStraightAlreadySet ( ) const [inline]
```

Retourne vrai si la petite suite a été défini.

#### Renvoie

const bool&

Définition à la ligne 381 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :





#### 5.8.3.34 isSuperYamsAlreadySet()

```
const bool& QPlayerGrid::isSuperYamsAlreadySet ( ) const [inline]
```

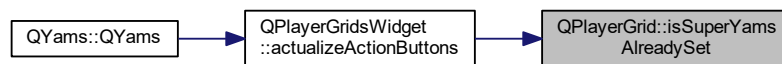
Retourne vrai si le super Yams a été défini.

Renvoie

const bool&

Définition à la ligne 438 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.35 isThreesAlreadySet()

```
const bool& QPlayerGrid::isThreesAlreadySet ( ) const [inline]
```

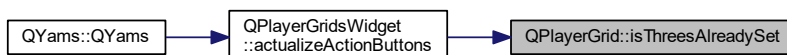
Retourne vrai si les trois ont été défini.

Renvoie

const bool&

Définition à la ligne 240 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.36 isTwosAlreadySet()

```
const bool& QPlayerGrid::isTwosAlreadySet ( ) const [inline]
```

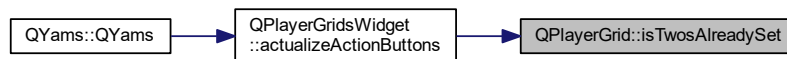
Retourne vrai si les deux ont été défini.

##### Renvoie

const bool&

Définition à la ligne 221 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.37 isYamsAlreadySet()

```
const bool& QPlayerGrid::isYamsAlreadySet ( ) const [inline]
```

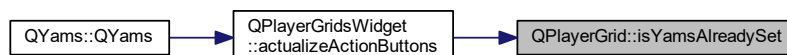
Retourne vrai si le Yams a été défini.

##### Renvoie

const bool&

Définition à la ligne 419 du fichier QPlayerGrid.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.38 setAces()

```
void QPlayerGrid::setAces (
    uint iValue )
```

Défini la valeur des As.

iValue : uint

Définition à la ligne 57 du fichier QPlayerGrid.cpp.

#### 5.8.3.39 setBrelan()

```
void QPlayerGrid::setBrelan (
    uint iValue )
```

Défini la valeur du Brelan.

iValue : uint

Définition à la ligne 141 du fichier QPlayerGrid.cpp.

#### 5.8.3.40 setCarre()

```
void QPlayerGrid::setCarre (
    uint iValue )
```

Défini la valeur du Carré

iValue : uint

Définition à la ligne 154 du fichier QPlayerGrid.cpp.

#### 5.8.3.41 setChance()

```
void QPlayerGrid::setChance (
    uint iChance )
```

Défini la valeur de la chance.

iChance : uint

Définition à la ligne 232 du fichier QPlayerGrid.cpp.

#### 5.8.3.42 setFives()

```
void QPlayerGrid::setFives (
    uint iValue )
```

Défini la valeur des cinq.

iValue : uint

Définition à la ligne 113 du fichier QPlayerGrid.cpp.

#### 5.8.3.43 setFours()

```
void QPlayerGrid::setFours (
    uint iValue )
```

Défini la valeur des quatre.

iValue : uint

Définition à la ligne 99 du fichier QPlayerGrid.cpp.

#### 5.8.3.44 setFull()

```
void QPlayerGrid::setFull (
    bool bIsFull )
```

Défini s'il y a Full.

bIsFull : bool

Définition à la ligne 167 du fichier QPlayerGrid.cpp.

#### 5.8.3.45 setLargeStraight()

```
void QPlayerGrid::setLargeStraight (
    bool bIsLargeStraight )
```

Défini s'il y a grande suite.

bIsLargeStraight : bool

Définition à la ligne 193 du fichier QPlayerGrid.cpp.

#### 5.8.3.46 setSixes()

```
void QPlayerGrid::setSixes (
    uint iValue )
```

Défini la valeur des six.

iValue : uint

Définition à la ligne 127 du fichier QPlayerGrid.cpp.

#### 5.8.3.47 setSmallStraight()

```
void QPlayerGrid::setSmallStraight (
    bool bIsSmallStraight )
```

Défini s'il y a petite suite.

bIsSmallStraight : bool

Définition à la ligne 180 du fichier QPlayerGrid.cpp.

#### 5.8.3.48 setSuperYams()

```
void QPlayerGrid::setSuperYams (
    bool bIsSuperYams )
```

Défini s'il y a super Yams.

bIsSuperYams : bool

Définition à la ligne 219 du fichier QPlayerGrid.cpp.

#### 5.8.3.49 setThrees()

```
void QPlayerGrid::setThrees (
    uint iValue )
```

Défini la valeur des trois.

iValue : uint

Définition à la ligne 85 du fichier QPlayerGrid.cpp.

#### 5.8.3.50 setTwos()

```
void QPlayerGrid::setTwos (
    uint iValue )
```

Défini la valeur des deux.

iValue : uint

Définition à la ligne 71 du fichier QPlayerGrid.cpp.

#### 5.8.3.51 setYams()

```
void QPlayerGrid::setYams (
    bool bIsYams )
```

Défini s'il y a Yams.

blsYams : bool

Définition à la ligne 206 du fichier QPlayerGrid.cpp.

### 5.8.4 Documentation des données membres

#### 5.8.4.1 m\_bIsAcesAlreadySet

```
bool QPlayerGrid::m_bIsAcesAlreadySet [private]
```

Vrai si les as ont déjà été définis.

Définition à la ligne 37 du fichier QPlayerGrid.hpp.

#### 5.8.4.2 m\_bIsBonus

```
bool QPlayerGrid::m_bIsBonus [private]
```

Vrai si le bonus est acquis.

Définition à la ligne 92 du fichier QPlayerGrid.hpp.

#### 5.8.4.3 m\_bIsBrelanAlreadySet

```
bool QPlayerGrid::m_bIsBrelanAlreadySet [private]
```

Vrai si le Brelan a déjà été défini.

Définition à la ligne 103 du fichier QPlayerGrid.hpp.

#### 5.8.4.4 m\_bIsCarreAlreadySet

```
bool QPlayerGrid::m_bIsCarreAlreadySet [private]
```

Vrai si le Carré a déjà été défini.

Définition à la ligne 113 du fichier QPlayerGrid.hpp.

#### 5.8.4.5 m\_bIsChanceAlreadySet

```
bool QPlayerGrid::m_bIsChanceAlreadySet [private]
```

Vrai si la chance a déjà été défini.

Définition à la ligne 173 du fichier QPlayerGrid.hpp.

#### 5.8.4.6 m\_bIsFivesAlreadySet

```
bool QPlayerGrid::m_bIsFivesAlreadySet [private]
```

Vrai si les cinq ont déjà été définis.

Définition à la ligne 77 du fichier QPlayerGrid.hpp.

#### 5.8.4.7 m\_bIsFoursAlreadySet

```
bool QPlayerGrid::m_bIsFoursAlreadySet [private]
```

Vrai si les quatre ont déjà été définis.

Définition à la ligne 67 du fichier QPlayerGrid.hpp.

#### 5.8.4.8 m\_bIsFull

```
bool QPlayerGrid::m_bIsFull [private]
```

Vrai s'il y a Full.

Définition à la ligne 118 du fichier QPlayerGrid.hpp.

#### 5.8.4.9 m\_bIsFullAlreadySet

```
bool QPlayerGrid::m_bIsFullAlreadySet [private]
```

Vrai si le Full a déjà été défini.

Définition à la ligne 123 du fichier QPlayerGrid.hpp.

#### 5.8.4.10 m\_bIsLargeStraight

```
bool QPlayerGrid::m_bIsLargeStraight [private]
```

Vrai s'il y a la grande suite.

Définition à la ligne 138 du fichier QPlayerGrid.hpp.

#### 5.8.4.11 m\_bIsLargeStraightAlreadySet

```
bool QPlayerGrid::m_bIsLargeStraightAlreadySet [private]
```

Vrai si la grande suite a déjà été défini.

Définition à la ligne 143 du fichier QPlayerGrid.hpp.

#### 5.8.4.12 m\_bIsSixesAlreadySet

```
bool QPlayerGrid::m_bIsSixesAlreadySet [private]
```

Vrai si les six ont déjà été définis.

Définition à la ligne 87 du fichier QPlayerGrid.hpp.



#### 5.8.4.13 m\_bIsSmallStraight

```
bool QPlayerGrid::m_bIsSmallStraight [private]
```

Vrai s'il y a la petite suite.

Définition à la ligne 128 du fichier QPlayerGrid.hpp.

#### 5.8.4.14 m\_bIsSmallStraightAlreadySet

```
bool QPlayerGrid::m_bIsSmallStraightAlreadySet [private]
```

Vrai si la petite suite a déjà été défini.

Définition à la ligne 133 du fichier QPlayerGrid.hpp.

#### 5.8.4.15 m\_bIsSuperYams

```
bool QPlayerGrid::m_bIsSuperYams [private]
```

Vrai s'il y a le super Yams.

Définition à la ligne 158 du fichier QPlayerGrid.hpp.

#### 5.8.4.16 m\_bIsSuperYamsAlreadySet

```
bool QPlayerGrid::m_bIsSuperYamsAlreadySet [private]
```

Vrai si le super Yams a déjà été défini.

Définition à la ligne 163 du fichier QPlayerGrid.hpp.

#### 5.8.4.17 m\_bIsThreesAlreadySet

```
bool QPlayerGrid::m_bIsThreesAlreadySet [private]
```

Vrai si les trois ont déjà été définis.

Définition à la ligne 57 du fichier QPlayerGrid.hpp.

#### 5.8.4.18 m\_bIsTwosAlreadySet

```
bool QPlayerGrid::m_bIsTwosAlreadySet [private]
```

Vrai si les deux ont déjà été définis.

Définition à la ligne 47 du fichier QPlayerGrid.hpp.

#### 5.8.4.19 m\_bIsYams

```
bool QPlayerGrid::m_bIsYams [private]
```

Vrai s'il y a le Yams.

Définition à la ligne 148 du fichier QPlayerGrid.hpp.

#### 5.8.4.20 m\_bIsYamsAlreadySet

```
bool QPlayerGrid::m_bIsYamsAlreadySet [private]
```

Vrai si le Yams a déjà été défini.

Définition à la ligne 153 du fichier QPlayerGrid.hpp.

#### 5.8.4.21 m\_iAces

```
uint QPlayerGrid::m_iAces [private]
```

Valeur des As.

Définition à la ligne 32 du fichier QPlayerGrid.hpp.

#### 5.8.4.22 m\_iBrelan

```
uint QPlayerGrid::m_iBrelan [private]
```

Valeur du Brelan.

Définition à la ligne 98 du fichier QPlayerGrid.hpp.

#### 5.8.4.23 m\_iCarre

```
uint QPlayerGrid::m_iCarre [private]
```

Valeur du Carré

Définition à la ligne 108 du fichier QPlayerGrid.hpp.

#### 5.8.4.24 m\_iChance

```
uint QPlayerGrid::m_iChance [private]
```

Valeur de la chance.

Définition à la ligne 168 du fichier QPlayerGrid.hpp.

#### 5.8.4.25 m\_iFives

```
uint QPlayerGrid::m_iFives [private]
```

Valeur des cinq.

Définition à la ligne 72 du fichier QPlayerGrid.hpp.

#### 5.8.4.26 m\_iFours

```
uint QPlayerGrid::m_iFours [private]
```

Valeur des quatre.

Définition à la ligne 62 du fichier QPlayerGrid.hpp.

#### 5.8.4.27 m\_iSixes

```
uint QPlayerGrid::m_iSixes [private]
```

Valeur des six.

Définition à la ligne 82 du fichier QPlayerGrid.hpp.

#### 5.8.4.28 m\_iThrees

```
uint QPlayerGrid::m_iThrees [private]
```

Valeur des trois.

Définition à la ligne 52 du fichier QPlayerGrid.hpp.

#### 5.8.4.29 m\_iTwos

```
uint QPlayerGrid::m_iTwos [private]
```

Valeur des deux.

Définition à la ligne 42 du fichier QPlayerGrid.hpp.

#### 5.8.4.30 m\_sPlayerName

```
QString QPlayerGrid::m_sPlayerName [private]
```

Nom du joueur.

Définition à la ligne 26 du fichier QPlayerGrid.hpp.

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

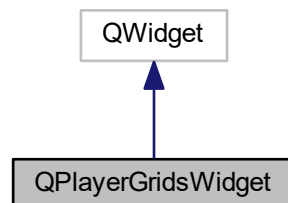
- [src/models/QPlayerGrid.hpp](#)
- [src/models/QPlayerGrid.cpp](#)

## 5.9 Référence de la classe QPlayerGridsWidget

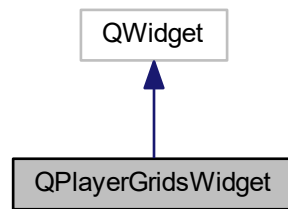
Vue des grilles de jeu.

```
#include <QPlayerGridsWidget.hpp>
```

Graphe d'héritage de QPlayerGridsWidget:



Graphe de collaboration de QPlayerGridsWidget:



## Classes

- class `CPlayerGridView`  
*Encapsule une vue de grille de jeu.*

## Connecteurs publics

- void `actualizeActionButtons` (`QPlayerGrid` &playerGrid) const  
*Active et désactive les boutons des actions en fonction des coups déjà réalisés d'un joueur.*
- void `updateGrid` (`QPlayerGrid` \*ptrQPlayerGrid)  
*Met à jour la vue d'une grille.*

## Fonctions membres publiques

- `QPlayerGridsWidget` (`QWidget` \*parent=Q\_NULLPTR)  
*Constructeur.*
- const `QPushButton` \* `getAcesButton` () const  
*Retourne le bouton des As.*
- const `QPushButton` \* `getTwosButton` () const  
*Retourne le bouton des deux.*
- const `QPushButton` \* `getThreeButton` () const  
*Retourne le bouton des trois.*
- const `QPushButton` \* `getFoursButton` () const  
*Retourne le bouton des quatre.*
- const `QPushButton` \* `getFivesButton` () const  
*Retourne le bouton des cinq.*
- const `QPushButton` \* `getSixesButton` () const  
*Retourne le bouton des six.*
- const `QPushButton` \* `getBrelanButton` () const  
*Retourne le bouton du Brelan.*
- const `QPushButton` \* `getCarreButton` () const  
*Retourne le bouton du Carré*
- const `QPushButton` \* `getFullButton` () const  
*Retourne le bouton du Full.*
- const `QPushButton` \* `getSmallStraightButton` () const  
*Retourne le bouton de la petite suite.*
- const `QPushButton` \* `getLargeStraightButton` () const  
*Retourne le bouton de la grande suite.*
- const `QPushButton` \* `getYamsButton` () const  
*Retourne le bouton du Yams.*
- const `QPushButton` \* `getChanceButton` () const  
*Retourne le bouton de la chance.*

- const QPushButton \* [getSuperYamsButton](#) () const  
*Retourne le bouton du super Yams.*
- void [addGrid](#) (QPlayerGrid \*ptrQPlayerGrid)  
*Ajoute une grille à la liste d'affichage.*
- void [removeGrid](#) (QPlayerGrid \*ptrQPlayerGrid)  
*Supprime une grille de la liste d'affichage.*
- void [enableActionButtons](#) (bool bEnable) const  
*Active / désactive les boutons des actions.*

## Attributs privés

- Ui::QPlayerGridsWidget [m\\_ui](#)  
*Interface utilisateur / vue.*
- std::map< [QPlayerGrid](#) \*, std::unique\_ptr< [CPlayerGridView](#) > > [m\\_mapPlayerGridViews](#)  
*Associe chaque modèle de grille à sa vue respective.*

### 5.9.1 Description détaillée

Vue des grilles de jeu.

#### Auteur

Corentin VÉROT

#### Date

2020-05-15

Définition à la ligne 25 du fichier QPlayerGridsWidget.hpp.

### 5.9.2 Documentation des constructeurs et destructeur

#### 5.9.2.1 QPlayerGridsWidget()

```
QPlayerGridsWidget::QPlayerGridsWidget (
    QWidget * parent = Q_NULLPTR )
```

Constructeur.

#### Paramètres

<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR
---------------	--

Définition à la ligne 5 du fichier QPlayerGridsWidget.cpp.

### 5.9.3 Documentation des fonctions membres

### 5.9.3.1 actualizeActionButtons

```
void QPlayerGridsWidget::actualizeActionButtons (
    QPlayerGrid & playerGrid ) const [slot]
```

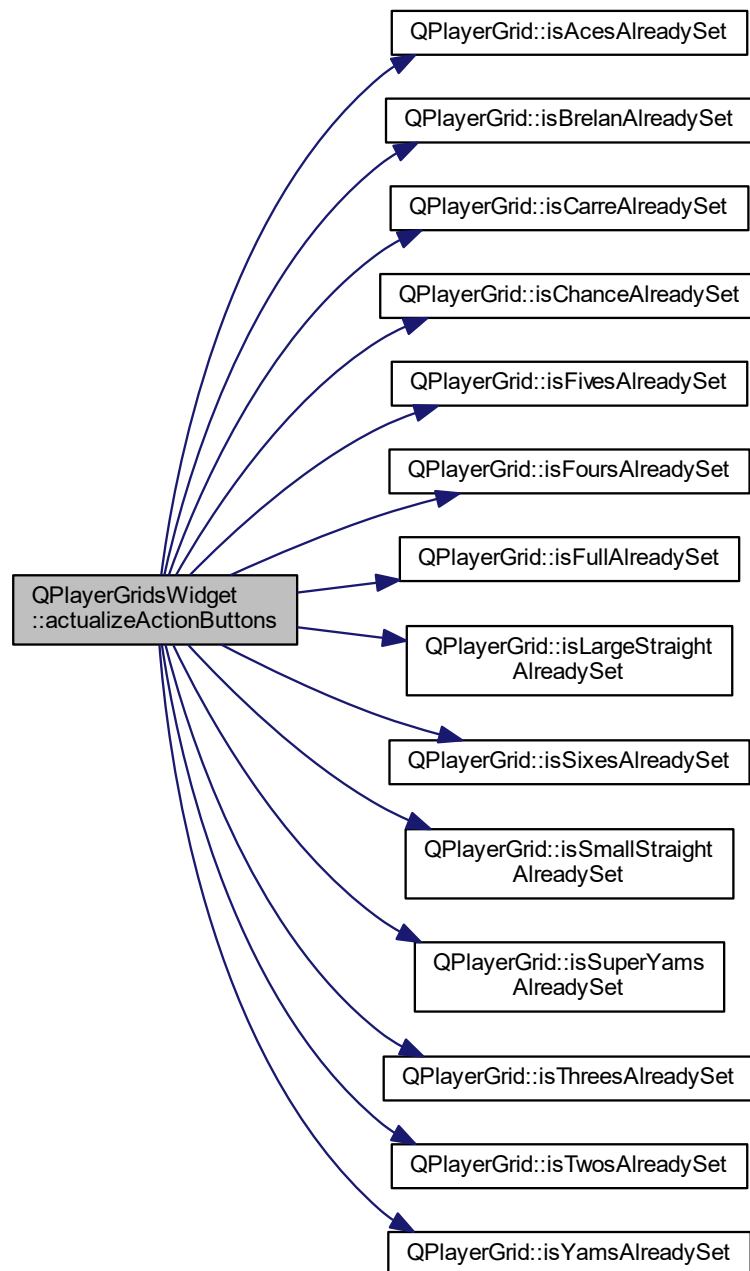
Active et désactive les boutons des actions en fonction des coups déjà réalisés d'un joueur.

#### Paramètres

<i>playerGrid</i>	: QPlayerGrid& - Grille du joueur
-------------------	-----------------------------------

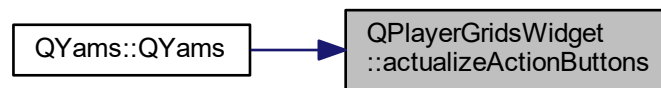
Définition à la ligne 84 du fichier QPlayerGridsWidget.cpp.

Voici le graphe d'appel pour cette fonction :





Voici le graphe des appelants de cette fonction :



### 5.9.3.2 addGrid()

```
void QPlayerGridsWidget::addGrid (
    QPlayerGrid * ptrQPlayerGrid )
```

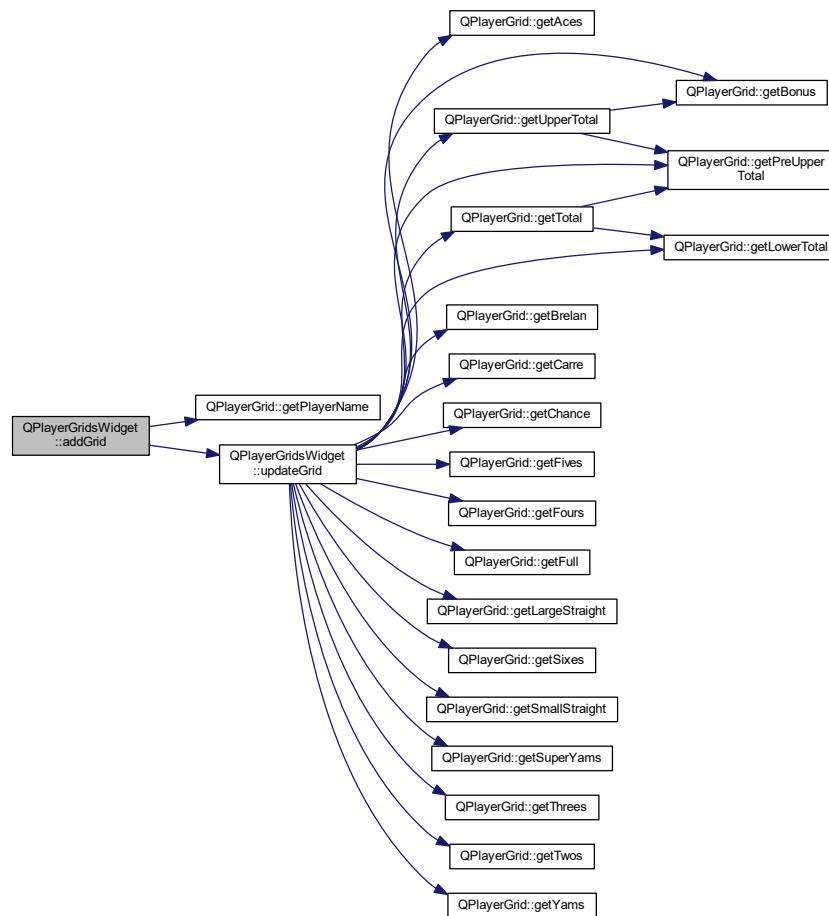
Ajoute une grille à la liste d'affichage.

#### Paramètres

<i>ptrQPlayerGrid</i>	: QPlayerGrid* - Grille à ajouter
-----------------------	-----------------------------------

Définition à la ligne 11 du fichier QPlayerGridsWidget.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.9.3.3 enableActionButtons()

```
void QPlayerGridsWidget::enableActionButtons (
    bool bEnable ) const
```

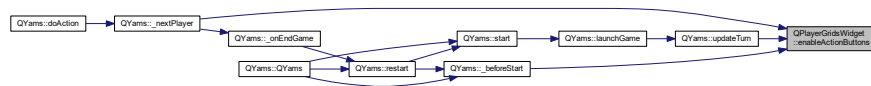
Active / désactive les boutons des actions.

## Paramètres

<i>bEnable</i>	: bool - Si vrai : active, si faux : désactive
----------------	--

Définition à la ligne 131 du fichier QPlayerGridsWidget.cpp.

Voici le graphe des appelants de cette fonction :



## 5.9.3.4 getAcesButton()

```
const QPushButton* QPlayerGridsWidget::getAcesButton ( ) const [inline]
```

Retourne le bouton des As.

## Renvoie

QPushButton\*

Définition à la ligne 158 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.5 getBrelanButton()

```
const QPushButton* QPlayerGridsWidget::getBrelanButton ( ) const [inline]
```

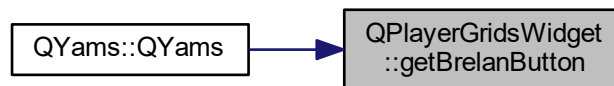
Retourne le bouton du Brelan.

Renvoie

QPushButton\*

Définition à la ligne 200 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.6 getCarreButton()

```
const QPushButton* QPlayerGridsWidget::getCarreButton ( ) const [inline]
```

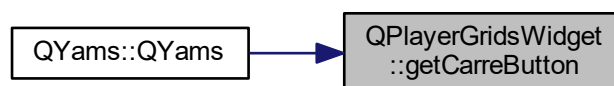
Retourne le bouton du Carré

Renvoie

QPushButton\*

Définition à la ligne 207 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.7 getChanceButton()

```
const QPushButton* QPlayerGridsWidget::getChanceButton ( ) const [inline]
```

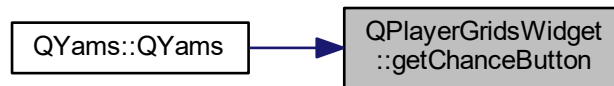
Retourne le bouton de la chance.

Renvoie

QPushButton\*

Définition à la ligne 242 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.8 getFivesButton()

```
const QPushButton* QPlayerGridsWidget::getFivesButton ( ) const [inline]
```

Retourne le bouton des cinq.

Renvoie

QPushButton\*

Définition à la ligne 186 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.9 getFoursButton()

```
const QPushButton* QPlayerGridsWidget::getFoursButton ( ) const [inline]
```

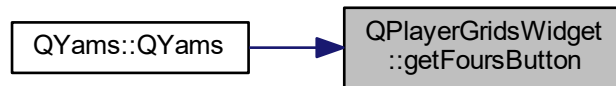
Retourne le bouton des quatre.

Renvoie

QPushButton\*

Définition à la ligne 179 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.10 getFullButton()

```
const QPushButton* QPlayerGridsWidget::getFullButton ( ) const [inline]
```

Retourne le bouton du Full.

Renvoie

QPushButton\*

Définition à la ligne 214 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.11 getLargeStraightButton()

```
const QPushButton* QPlayerGridsWidget::getLargeStraightButton ( ) const [inline]
```

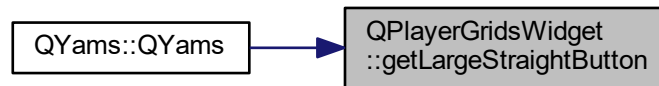
Retourne le bouton de la grande suite.

Renvoie

QPushButton\*

Définition à la ligne 228 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.12 getSixesButton()

```
const QPushButton* QPlayerGridsWidget::getSixesButton ( ) const [inline]
```

Retourne le bouton des six.

Renvoie

QPushButton\*

Définition à la ligne 193 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.13 getSmallStraightButton()

```
const QPushButton* QPlayerGridsWidget::getSmallStraightButton ( ) const [inline]
```

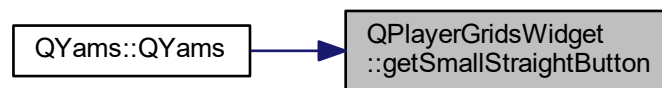
Retourne le bouton de la petite suite.

Renvoie

QPushButton\*

Définition à la ligne 221 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.14 getSuperYamsButton()

```
const QPushButton* QPlayerGridsWidget::getSuperYamsButton ( ) const [inline]
```

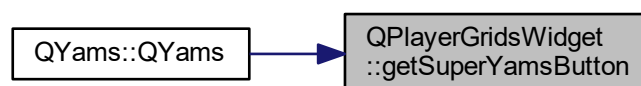
Retourne le bouton du super Yams.

Renvoie

QPushButton\*

Définition à la ligne 249 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :





#### 5.9.3.15 getThreeButton()

```
const QPushButton* QPlayerGridsWidget::getThreeButton ( ) const [inline]
```

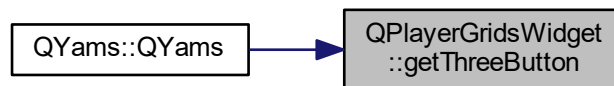
Retourne le bouton des trois.

Renvoie

QPushButton\*

Définition à la ligne 172 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.16 getTwosButton()

```
const QPushButton* QPlayerGridsWidget::getTwosButton ( ) const [inline]
```

Retourne le bouton des deux.

Renvoie

QPushButton\*

Définition à la ligne 165 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.17 getYamsButton()

```
const QPushButton* QPlayerGridsWidget::getYamsButton ( ) const [inline]
```

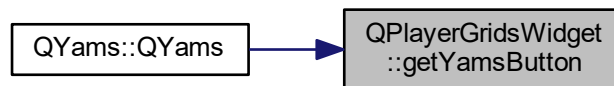
Retourne le bouton du Yams.

Renvoie

QPushButton\*

Définition à la ligne 235 du fichier QPlayerGridsWidget.hpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.18 removeGrid()

```
void QPlayerGridsWidget::removeGrid (
    QPlayerGrid * ptrQPlayerGrid )
```

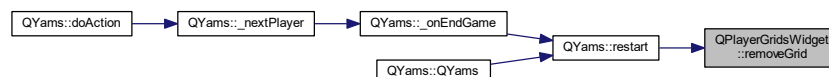
Supprime une grille de la liste d'affichage.

Paramètres

<i>ptrQPlayerGrid</i>	: QPlayerGrid* - Grille à supprimer
-----------------------	-------------------------------------

Définition à la ligne 79 du fichier QPlayerGridsWidget.cpp.

Voici le graphe des appelants de cette fonction :



### 5.9.3.19 updateGrid

```
void QPlayerGridsWidget::updateGrid (  
    QPlayerGrid * ptrQPlayerGrid ) [slot]
```

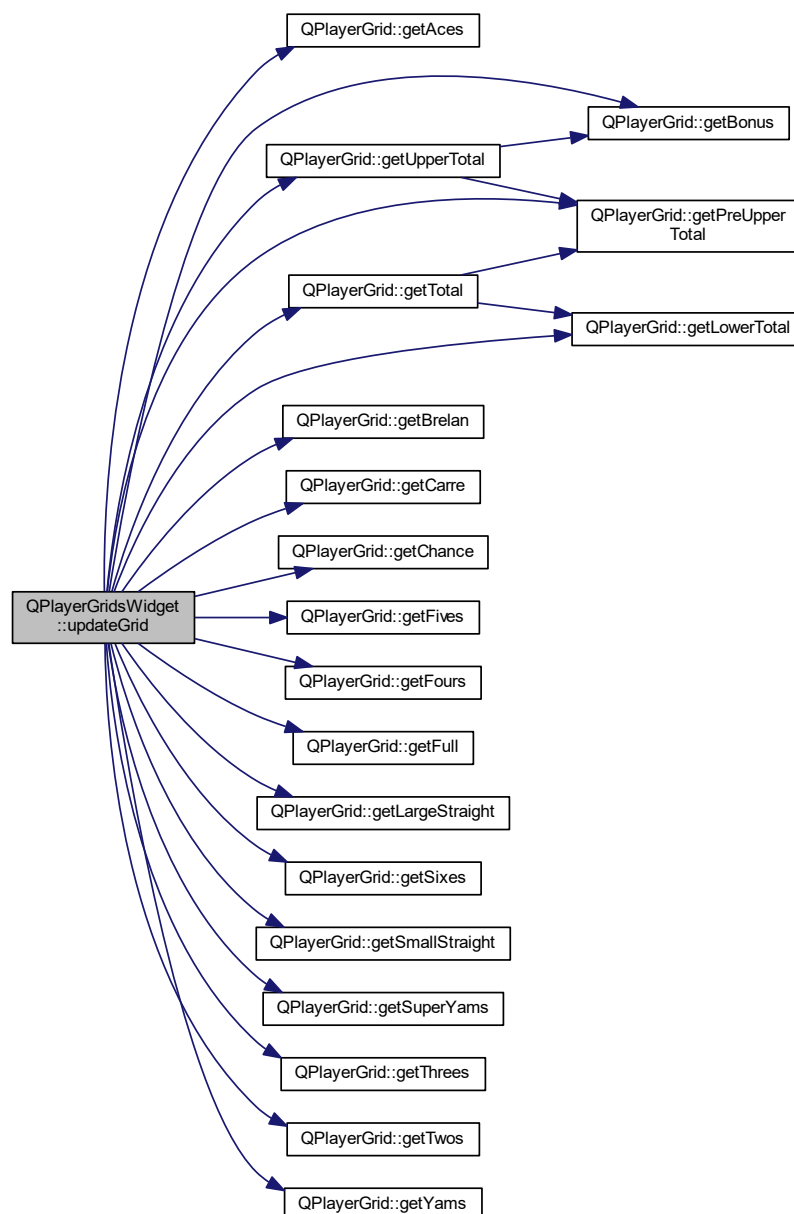
Met à jour la vue d'une grille.

#### Paramètres

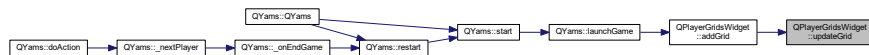
<i>ptrQPlayerGrid</i>	: <a href="#">QPlayerGrid</a> & - Grille du joueur
-----------------------	--

Définition à la ligne 103 du fichier QPlayerGridsWidget.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



## 5.9.4 Documentation des données membres

### 5.9.4.1 m\_mapPlayerGridViews

```
std::map<QPlayerGrid*, std::unique_ptr<CPlayerGridView> > QPlayerGridsWidget::m_mapPlayerGridViews [private]
```

Associe chaque modèle de grille à sa vue respective.

Définition à la ligne 142 du fichier QPlayerGridsWidget.hpp.

### 5.9.4.2 m\_ui

```
Ui::QPlayerGridsWidget QPlayerGridsWidget::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 137 du fichier QPlayerGridsWidget.hpp.

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

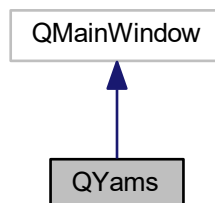
- src/[QPlayerGridsWidget.hpp](#)
- src/[QPlayerGridsWidget.cpp](#)

## 5.10 Référence de la classe QYams

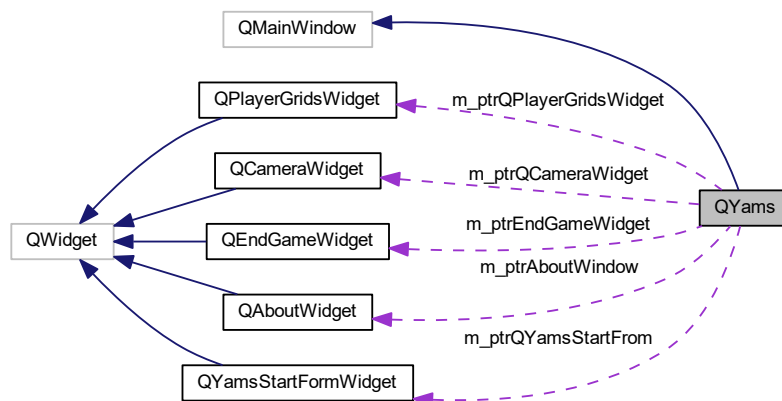
Contôleur du jeu Yams.

```
#include <QYams.hpp>
```

Graphe d'héritage de QYams:



Graphe de collaboration de QYams:



## Types publics

- enum `EYamsActions` {  
`NOTHING` = 0, `ACES` = 1, `TWOS` = 2, `THREES` = 3,  
`FOURS` = 4, `FIVES` = 5, `SIXES` = 6, `BRELAN` = 7,  
`CARRE` = 8, `FULL` = 9, `SMALL_STRAIGHT` = 10, `LARGE_STRAIGHT` = 11,  
`YAMS` = 12, `SUPER_YAMS` = 13, `CHANCE` = 14 }  
*Toutes les actions réalisables par le joueur.*
- enum `EChoices` { `ANYTHING` = 0, `ONE` = 1, `TWO` = 2, `THREE` = 3 }  
*Les choix possibles (1 choix = 1 bouton)*

## Connecteurs publics

- void `start` ()  
*Démarre une partie.*
- void `restart` ()  
*Redémarre le jeu.*
- void `quit` ()  
*Quitte l'application.*
- void `doAction` (`EYamsActions` selectedAction)  
*Effectue l'action choisie par le joueur.*
- void `updateTurn` (`CDiceSet` &diceSet, bool isDetectionCorrection)  
*Réagit lorsqu'un nouveau jeu de dés est détecté*
- void `launchGame` (std::list< QString > \*ptrLPlayerNames)  
*Lance le jeu après la saisie des noms de joueurs.*
- void `showAboutWindow` ()  
*Affiche la fenêtre "À propos".*

## Signaux

- void `playerUpdated` (`QPlayerGrid` &playerGrid)  
*Signal émit lorsque le tour du joueur a évolué*

## Fonctions membres publiques

- [QYams](#) (QWidget \*parent=Q\_NULLPTR)  
*Constructeur.*
- [~QYams](#) ()  
*Destructeur.*

## Fonctions membres privées

- void [closeEvent](#) (QCloseEvent \*event) override  
*Action effectuée lors de la fermeture de la fenêtre.*
- void [\\_simulate](#) (CDiceSet &diceSet, std::vector< std::pair< [QYams::EYamsActions](#), uint >> &vSorted, SimulationResult)  
*Simule les score pouvant être obtenus avec la combinaison de dés détectée.*
- void [\\_nextPlayer](#) ()  
*Passe au joueur suivant.*
- void [\\_onEndGame](#) ()  
*Action à effectuer à la fin de la partie.*
- void [\\_resetChoices](#) (EChoices choice=ANYTHING)  
*Remet le ou les bouton(s) de choix à leur état d'origine.*
- void [\\_hideGameBar](#) (bool bHide) const  
*Cache le layout contenant les boutons de choix et le compteur de tour.*
- void [\\_beforeStart](#) ()  
*Action à effectuer avant de démarrer la partie.*
- void [\\_showRules](#) ()  
*Affiche le menu des règles.*

## Attributs privés

- std::map< [EYamsActions](#), QString > [m\\_YamsActionsNames](#)  
*Associe une action à un texte pour l'affichage.*
- Ui::QYams [m\\_ui](#)  
*Interface utilisateur / vue.*
- [QPlayerGridsWidget](#) \* [m\\_ptrQPlayerGridsWidget](#)  
*Vue des grilles des joueurs.*
- [QYamsStartFormWidget](#) \* [m\\_ptrQYamsStartFrom](#)  
*Menu de début de jeu.*
- [QCameraWidget](#) \* [m\\_ptrQCameraWidget](#)  
*Widget réalisant la reconnaissance d'image.*
- [QAboutWidget](#) \* [m\\_ptrAboutWindow](#) = nullptr  
*Fenêtre "À propos".*
- [QEndGameWidget](#) \* [m\\_ptrEndGameWidget](#) = nullptr  
*Menu de fin de jeu.*
- std::map< [QYams::EYamsActions](#), uint > [m\\_mapLastSimulation](#)  
*Contient la dernière simulation des scores.*
- std::list< [QPlayerGrid](#) \* > [m\\_lpQPlayerGrids](#)  
*Liste contenant toutes les grilles des joueurs.*
- int [m\\_iNbrTurn](#) = 0  
*Nombre de lancer effectués durant le tour présent.*
- std::list< [QPlayerGrid](#) \* >::iterator [m\\_itPlayerGrids](#)  
*Joueur actuel.*
- [EYamsActions](#) [m\\_choice1](#)  
*Action du bouton "Choix 1".*
- [EYamsActions](#) [m\\_choice2](#)  
*Action du bouton "Choix 2".*
- [EYamsActions](#) [m\\_choice3](#)  
*Action du bouton "Choix 3".*

### 5.10.1 Description détaillée

Contrôleur du jeu Yams.

**Auteur**

Corentin VÉROT

**Date**

2020-05-15

Définition à la ligne 35 du fichier QYams.hpp.

**5.10.2 Documentation des énumérations membres****5.10.2.1 EChoices**

```
enum QYams::EChoices
```

Les choix possibles (1 choix = 1 bouton)

**Valeurs énumérées**

ANYTHING	
ONE	
TWO	
THREE	

Définition à la ligne 68 du fichier QYams.hpp.

**5.10.2.2 EYamsActions**

```
enum QYams::EYamsActions
```

Toutes les actions réalisables par le joueur.

**Valeurs énumérées**

NOTHING	
ACES	
TWOS	
THREES	
FOURS	
FIVES	
SIXES	
BRELAN	
CARRE	
FULL	
SMALL_STRAIGHT	

## Valeurs énumérées

LARGE_STRAIGHT	
YAMS	
SUPER_YAMS	
CHANCE	

Définition à la ligne 44 du fichier QYams.hpp.

### 5.10.3 Documentation des constructeurs et destructeur

#### 5.10.3.1 QYams()

```
QYams::QYams (
    QWidget * parent = Q_NULLPTR )
```

Constructeur.

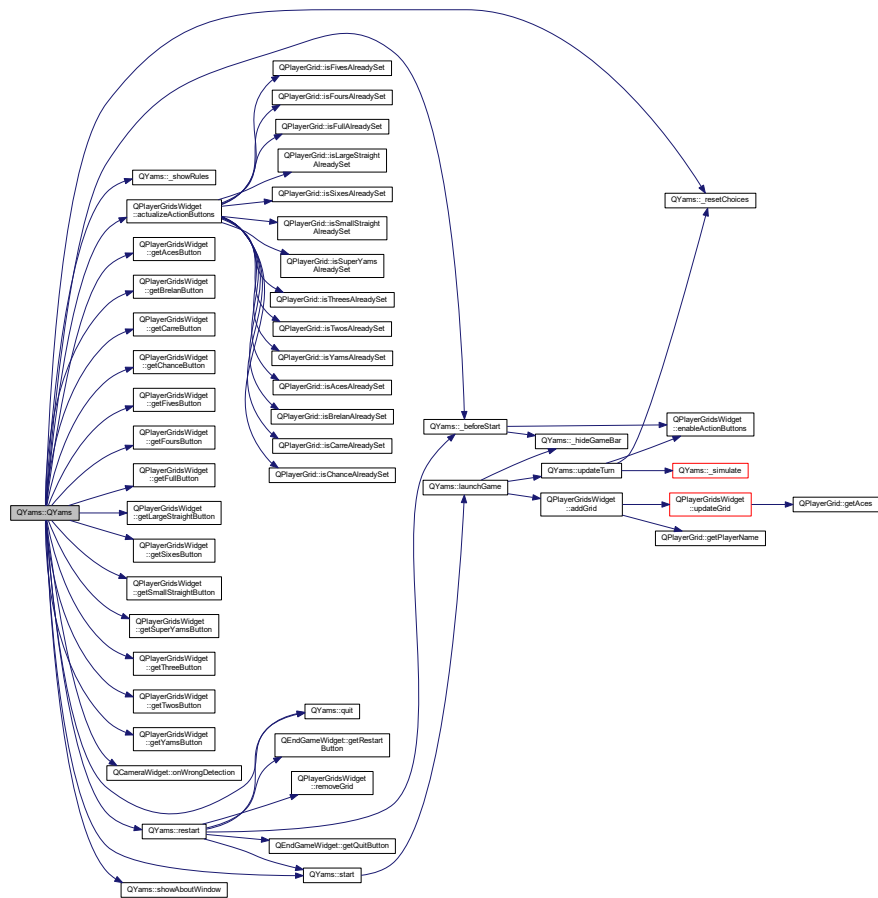
## Paramètres

<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR
---------------	--

Définition à la ligne 5 du fichier QYams.cpp.



Voici le graphe d'appel pour cette fonction :



### 5.10.3.2 ~QYams()

QYams::~~QYams ( )

Destructeur.

Définition à la ligne 52 du fichier QYams.cpp.

### 5.10.4 Documentation des fonctions membres

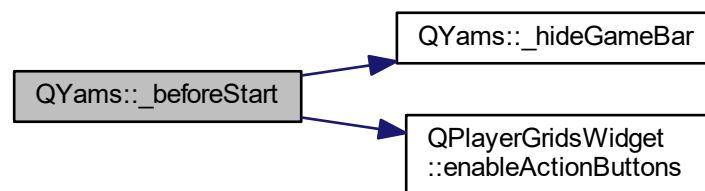
#### 5.10.4.1 `_beforeStart()`

```
void QYams::_beforeStart ( ) [private]
```

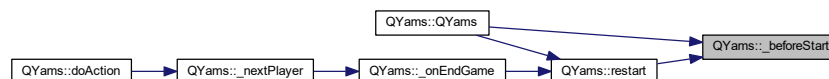
Action à effectuer avant de démarrer la partie.

Définition à la ligne 404 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.10.4.2 `_hideGameBar()`

```
void QYams::_hideGameBar (
    bool bHide ) const [private]
```

Cache le layout contenant les boutons de choix et le compteur de tour.

##### Paramètres

<code>bHide</code>	: bool - Si vrai : masque, si faux : affiche
--------------------	--

Définition à la ligne 392 du fichier QYams.cpp.



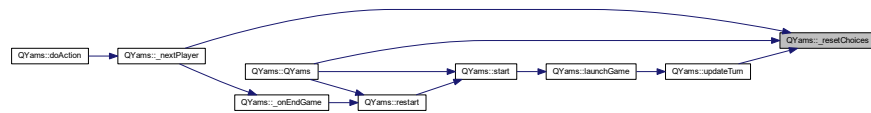


## Paramètres

<i>choice</i>	: EChoice - Valeur par défaut EChoices::ANYTHING
---------------	--

Définition à la ligne 365 du fichier QYams.cpp.

Voici le graphe des appelants de cette fonction :

5.10.4.6 `_showRules()`

```
void QYams::_showRules ( ) [private]
```

Affiche le menu des règles.

Définition à la ligne 417 du fichier QYams.cpp.

Voici le graphe des appelants de cette fonction :

5.10.4.7 `_simulate()`

```
void QYams::_simulate (
    CDiceSet & diceSet,
    std::vector< std::pair< QYams::EYamsActions, uint >> & vSortedSimulationResult )
[private]
```

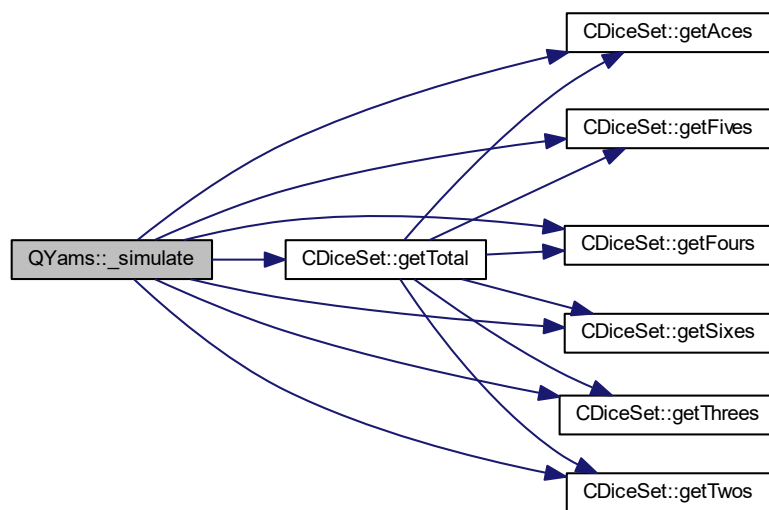
Simule les score pouvant être obtenus avec la combinaison de dés détectée.

## Paramètres

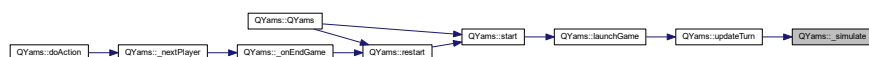
<i>diceSet</i>	: <a href="#">CDiceSet</a> & - Jeu de dés
<i>vSortedSimulationResult</i>	: <code>std::vector&lt;std::pair&lt;QYams::EYamsActions, uint&gt;&gt;&amp;</code> - Vecteur allant contenir les simulations triées de la solution optimale à la moins bonne

Définition à la ligne 68 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.10.4.8 closeEvent()

```
void QYams::closeEvent (
    QCloseEvent * event ) [override], [private]
```

Action effectuée lors de la fermeture de la fenêtre.

Quitte l'application étant donné que cette fenêtre est la fenêtre principale

## Paramètres

<i>event</i>	: QCloseEvent*
--------------	----------------

Définition à la ligne 61 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



#### 5.10.4.9 doAction

```
void QYams::doAction (
    EYamsActions selectedAction ) [slot]
```

Effectue l'action choisie par le joueur.

## Paramètres

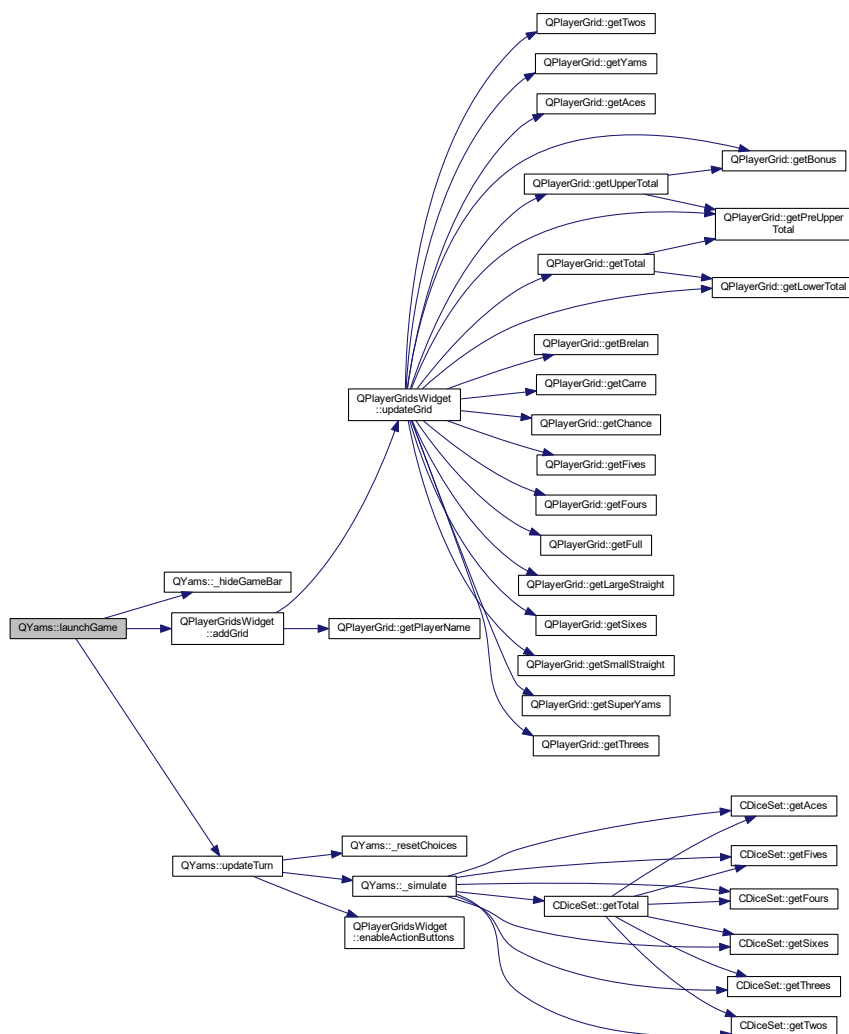
<i>selectedAction</i>	: EYamsActions - Action choisie par le joueur
-----------------------	---

Définition à la ligne 307 du fichier QYams.cpp.

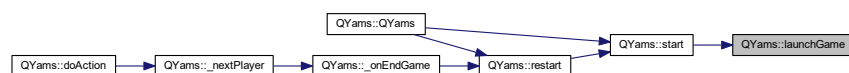




Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.10.4.11 playerUpdated

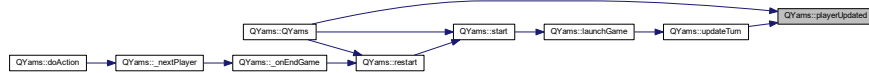
```
void QYams::playerUpdated (
    QPlayerGrid & playerGrid ) [signal]
```

Signal émit lorsque le tour du joueur a évolué

## Paramètres

<i>playerGrid</i>	: <a href="#">QPlayerGrid</a> & - Joueur concerné
-------------------	---

Voici le graphe des appelants de cette fonction :



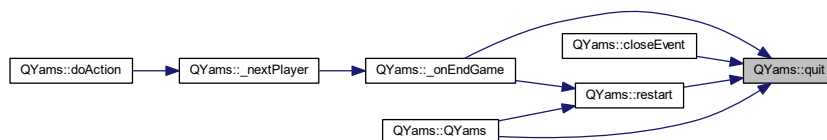
## 5.10.4.12 quit

```
void QYams::quit ( ) [slot]
```

Quitte l'application.

Définition à la ligne 476 du fichier QYams.cpp.

Voici le graphe des appelants de cette fonction :



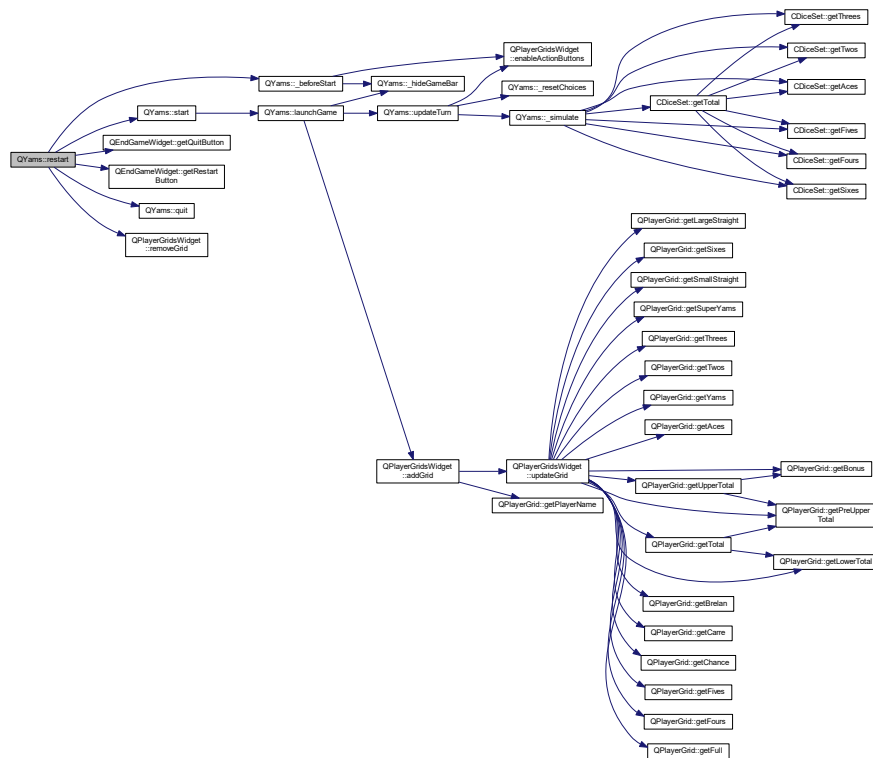
## 5.10.4.13 restart

```
void QYams::restart ( ) [slot]
```

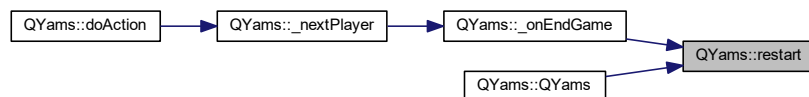
Redémarre le jeu.

Définition à la ligne 453 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.10.4.14 showAboutWindow

```
void QYams::showAboutWindow ( ) [slot]
```

Affiche la fenêtre "À propos".

Définition à la ligne 531 du fichier QYams.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.10.4.15 start

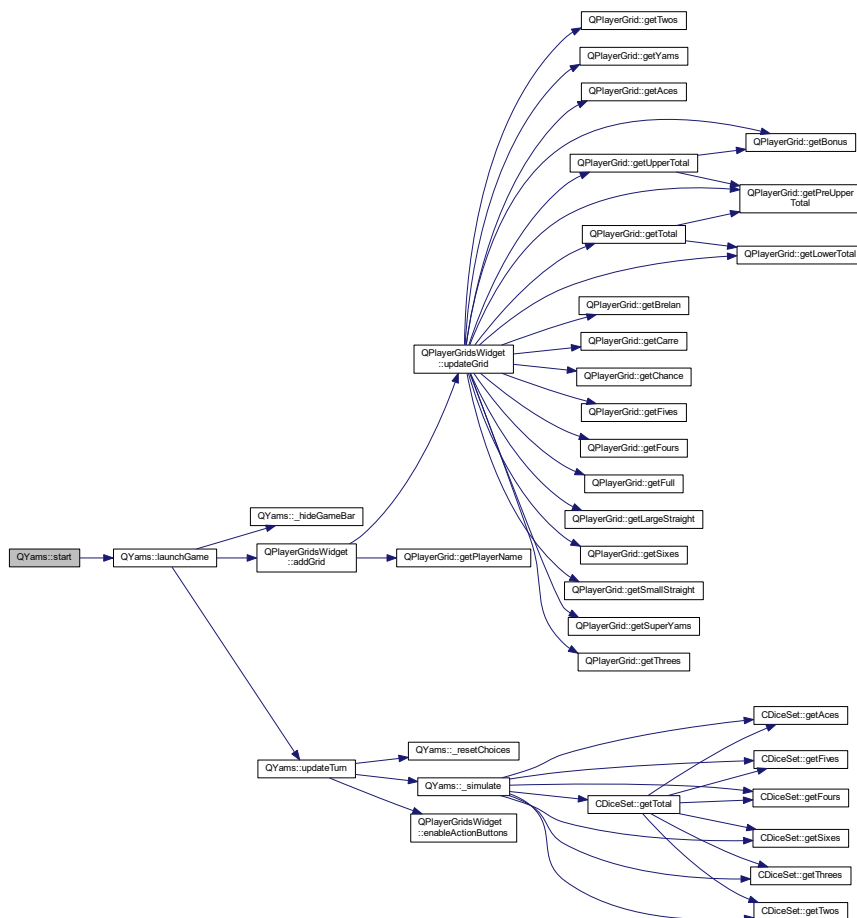
```
void QYams::start ( ) [slot]
```

Démarre une partie.

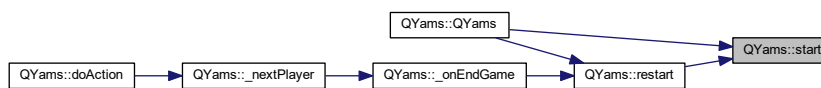
Affiche le menu d'entrée des noms de joueur

Définition à la ligne 443 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.10.4.16 updateTurn

```

void QYams::updateTurn (
    CDiceSet & diceSet,
    bool isDetectionCorrection ) [slot]

```

Réagit lorsqu'un nouveau jeu de dés est détecté

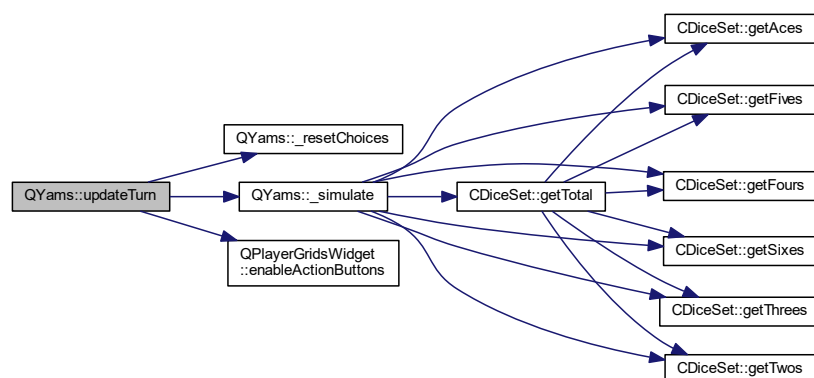
TODO : onDiceSetUpdate

##### Paramètres

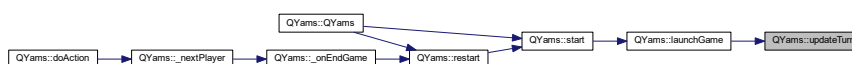
<i>diceSet</i>	: <a href="#">CDiceSet</a> & - Jeu de dés
<i>isDetectionCorrection</i>	: bool - S'agit-il de la suite d'une erreur signalée par le joueur ?

Définition à la ligne 481 du fichier QYams.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



## 5.10.5 Documentation des données membres

### 5.10.5.1 m\_choice1

```
EYamsActions QYams::m_choice1 [private]
```

Action du bouton "Choix 1".

Définition à la ligne 165 du fichier QYams.hpp.

### 5.10.5.2 m\_choice2

```
EYamsActions QYams::m_choice2 [private]
```

Action du bouton "Choix 2".

Définition à la ligne 170 du fichier QYams.hpp.

### 5.10.5.3 m\_choice3

```
EYamsActions QYams::m_choice3 [private]
```

Action du bouton "Choix 3".

Définition à la ligne 175 du fichier QYams.hpp.

### 5.10.5.4 m\_iNbrTurn

```
int QYams::m_iNbrTurn = 0 [private]
```

Nombre de lancer effectués durant le tour présent.

Définition à la ligne 152 du fichier QYams.hpp.

#### 5.10.5.5 m\_itPlayerGrids

```
std::list<QPlayerGrid*>::iterator QYams::m_itPlayerGrids [private]
```

Joueur actuel.

Itérateur de la liste [QYams::m\\_lpQPlayerGrids](#) Incrémenté à chaque fin de jours pour obtenir le joueur suivant

Définition à la ligne 160 du fichier QYams.hpp.

#### 5.10.5.6 m\_lpQPlayerGrids

```
std::list<QPlayerGrid*> QYams::m_lpQPlayerGrids [private]
```

Liste contenant toutes les grilles des joueurs.

Définition à la ligne 147 du fichier QYams.hpp.

#### 5.10.5.7 m\_mapLastSimulation

```
std::map<QYams::EYamsActions, uint> QYams::m_mapLastSimulation [private]
```

Contient la dernière simulation des scores.

La simulation est réutilisée lorsque le joueur choisit quelle action effectuer afin d'éviter de recalculer plusieurs fois les coups

Définition à la ligne 142 du fichier QYams.hpp.

#### 5.10.5.8 m\_ptrAboutWindow

```
QAboutWidget* QYams::m_ptrAboutWindow = nullptr [private]
```

Fenêtre "À propos".

Définition à la ligne 127 du fichier QYams.hpp.

#### 5.10.5.9 m\_ptrEndGameWidget

```
QEndGameWidget* QYams::m_ptrEndGameWidget = nullptr [private]
```

Menu de fin de jeu.

Affiche le nom du gagnant et des boutons permettant de rejouer ou de quitter le jeu

Définition à la ligne 134 du fichier QYams.hpp.

#### 5.10.5.10 m\_ptrQCameraWidget

```
QCameraWidget* QYams::m_ptrQCameraWidget [private]
```

Widget réalisant la reconnaissance d'image.

La reconnaissance d'image est réalisée grâce à la bibliothèque OpenCV

Définition à la ligne 122 du fichier QYams.hpp.

#### 5.10.5.11 m\_ptrQPlayerGridsWidget

```
QPlayerGridsWidget* QYams::m_ptrQPlayerGridsWidget [private]
```

Vue des grilles des joueurs.

Définition à la ligne 108 du fichier QYams.hpp.

#### 5.10.5.12 m\_ptrQYamsStartFrom

```
QYamsStartFormWidget* QYams::m_ptrQYamsStartFrom [private]
```

Menu de début de jeu.

Permet d'entrer les noms des joueurs

Définition à la ligne 115 du fichier QYams.hpp.

#### 5.10.5.13 m\_ui

```
Ui::QYams QYams::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 103 du fichier QYams.hpp.



## 5.10.5.14 m\_YamsActionsNames

```
std::map<EYamsActions, QString> QYams::m_YamsActionsNames [private]
```

## Valeur initiale :

```
= {
    {EYamsActions::ACES, QString::fromUtf8("As")},
    {EYamsActions::TWOS, QString::fromUtf8("Deux")},
    {EYamsActions::THREES, QString::fromUtf8("Trois")},
    {EYamsActions::FOURS, QString::fromUtf8("Quatre")},
    {EYamsActions::FIVES, QString::fromUtf8("Cinq")},
    {EYamsActions::SIXES, QString::fromUtf8("Six")},
    {EYamsActions::BRELAN, QString::fromUtf8("Brelan")},
    {EYamsActions::CARRE, QString::fromUtf8("Carré")},
    {EYamsActions::FULL, QString::fromUtf8("Full")},
    {EYamsActions::SMALL_STRAIGHT, QString::fromUtf8("Petite suite")},
    {EYamsActions::LARGE_STRAIGHT, QString::fromUtf8("Grande suite")},
    {EYamsActions::YAMS, QString::fromUtf8("Yams")},
    {EYamsActions::SUPER_YAMS, QString::fromUtf8("Super Yams")},
    {EYamsActions::CHANCE, QString::fromUtf8("Chance")},
}
```

Associe une action à un texte pour l'affichage.

Définition à la ligne 82 du fichier QYams.hpp.

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

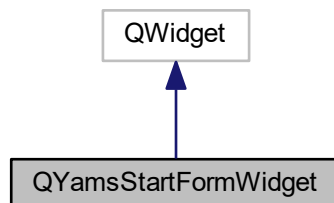
- [src/QYams.hpp](#)
- [src/QYams.cpp](#)

## 5.11 Référence de la classe QYamsStartFormWidget

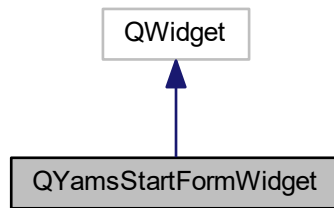
Fenêtre de saisie des noms de joueur.

```
#include <QYamsStartFormWidget.hpp>
```

Graphe d'héritage de QYamsStartFormWidget:



Graphe de collaboration de QYamsStartFormWidget:



## Classes

- class [CPlayerForm](#)  
*Encapsule le numéro et le champ de texte pour un joueur.*

## Signaux

- void [playerNameSetUp](#) (std::list< QString > \*ptrLPlayerNames)  
*Signal que les noms sont saisis.*

## Fonctions membres publiques

- [QYamsStartFormWidget](#) (QWidget \*parent=Q\_NULLPTR)  
*Constructeur.*

## Connecteurs privés

- void [addPlayerNameInput](#) ()  
*Ajoute une entrée supplémentaire pour ajouter un joueur à la liste.*
- void [onPlayerNameSetUp](#) ()  
*Action à réaliser lorsque l'utilisateur confirme la saisie.*

## Attributs privés

- Ui::QYamsStartFormWidget [m\\_ui](#)  
*Interface utilisateur / vue.*
- std::list< std::unique\_ptr< [CPlayerForm](#) > > [m\\_IPlayerNameInputs](#)  
*Contient les différentes entrées de noms des joueurs.*

### 5.11.1 Description détaillée

Fenêtre de saisie des noms de joueur.

**Auteur**

Corentin VÉROT

**Date**

2020-05-15

Définition à la ligne 25 du fichier QYamsStartFormWidget.hpp.

### 5.11.2 Documentation des constructeurs et destructeur

#### 5.11.2.1 QYamsStartFormWidget()

```
QYamsStartFormWidget::QYamsStartFormWidget (
    QWidget * parent = Q_NULLPTR )
```

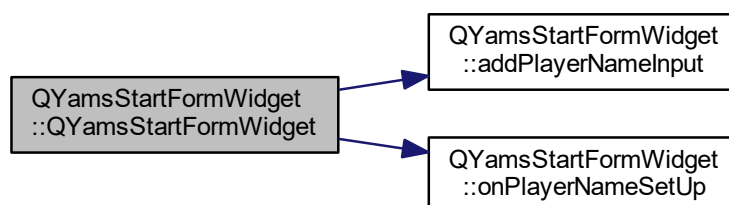
Constructeur.

**Paramètres**

<i>parent</i>	: QWidget* - Widget parent, défaut Q_NULLPTR
---------------	--

Définition à la ligne 19 du fichier QYamsStartFormWidget.cpp.

Voici le graphe d'appel pour cette fonction :



### 5.11.3 Documentation des fonctions membres

### 5.11.3.1 addPlayerNameInput

```
void QYamsStartFormWidget::addPlayerNameInput ( ) [private], [slot]
```

Ajoute une entrée supplémentaire pour ajouter un joueur à la liste.

Définition à la ligne 31 du fichier QYamsStartFormWidget.cpp.

Voici le graphe des appelants de cette fonction :



### 5.11.3.2 onPlayerNameSetUp

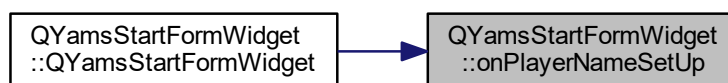
```
void QYamsStartFormWidget::onPlayerNameSetUp ( ) [private], [slot]
```

Action à réaliser lorsque l'utilisateur confirme la saisie.

Vérifie si les saisies ne sont pas vides puis envoie le résultat à [QYams::launchGame](#)

Définition à la ligne 39 du fichier QYamsStartFormWidget.cpp.

Voici le graphe des appelants de cette fonction :



### 5.11.3.3 playerNameSetUp

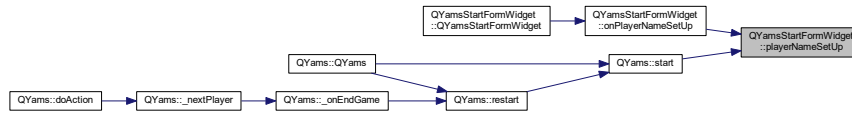
```
void QYamsStartFormWidget::playerNameSetUp (
    std::list< QString > * ptrLPlayerNames ) [signal]
```

Signal que les noms sont saisis.

## Paramètres

<i>ptrLPlayerNames</i>	: <code>std::list&lt;QString&gt;*</code> - Liste des noms saisis
------------------------	--

Voici le graphe des appelants de cette fonction :



## 5.11.4 Documentation des données membres

### 5.11.4.1 m\_lPlayerNameInputs

```
std::list<std::unique_ptr<CPlayerForm> > QYamsStartFormWidget::m_lPlayerNameInputs [private]
```

Contient les différentes entrées de noms des joueurs.

Définition à la ligne 92 du fichier `QYamsStartFormWidget.hpp`.

### 5.11.4.2 m\_ui

```
Ui::QYamsStartFormWidget QYamsStartFormWidget::m_ui [private]
```

Interface utilisateur / vue.

Définition à la ligne 87 du fichier `QYamsStartFormWidget.hpp`.

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

- [src/QYamsStartFormWidget.hpp](#)
- [src/QYamsStartFormWidget.cpp](#)



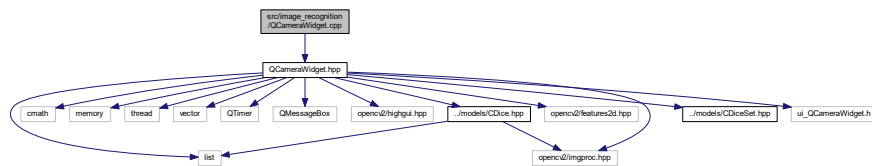
## Chapitre 6

# Documentation des fichiers

### 6.1 Référence du fichier src/image\_recognition/QCameraWidget.cpp

```
#include "QCameraWidget.hpp"
```

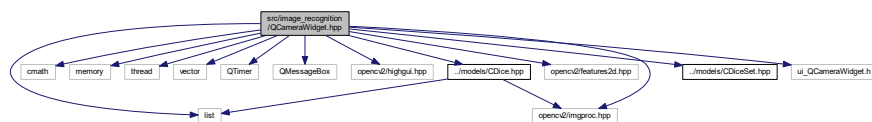
Graphe des dépendances par inclusion de QCameraWidget.cpp:



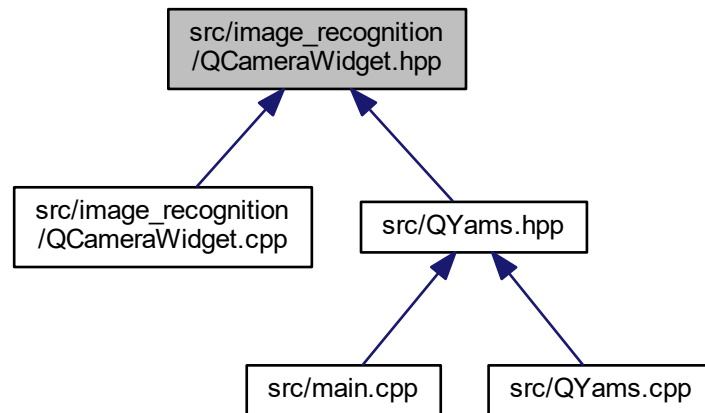
### 6.2 Référence du fichier src/image\_recognition/QCameraWidget.hpp

```
#include <list>
#include <cmath>
#include <memory>
#include <thread>
#include <vector>
#include <QTimer>
#include <QMessageBox>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/features2d.hpp>
#include "../models/CDice.hpp"
#include "../models/CDiceSet.hpp"
#include "ui_QCameraWidget.h"
```

Graphe des dépendances par inclusion de QCameraWidget.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



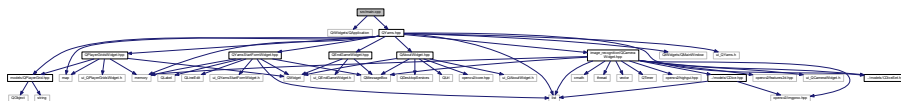
## Classes

- class [QCameraWidget](#)  
*Encapsule la détection d'image et le rendu vidéo.*

## 6.3 Référence du fichier src/main.cpp

```
#include <QtWidgets/QApplication>
#include "QYams.hpp"
```

Graphe des dépendances par inclusion de main.cpp:



## Fonctions

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

### 6.3.1 Documentation des fonctions



### 6.3.1.1 main()

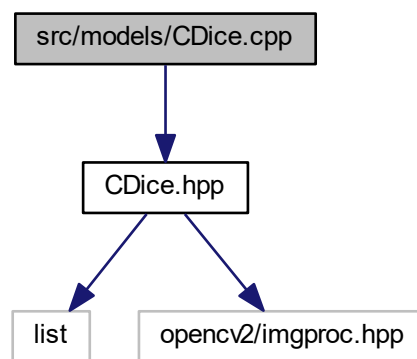
```
int main (
    int argc,
    char * argv[] )
```

Définition à la ligne 55 du fichier main.cpp.

## 6.4 Référence du fichier src/models/CDice.cpp

```
#include "CDice.hpp"
```

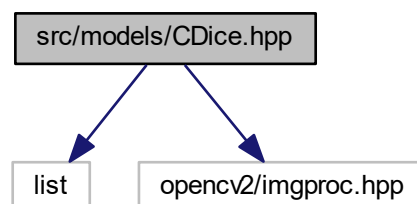
Graphe des dépendances par inclusion de CDice.cpp:



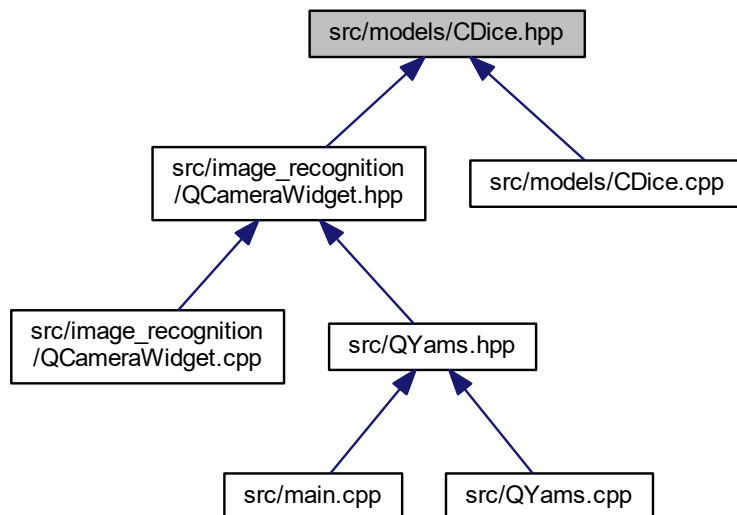
## 6.5 Référence du fichier src/models/CDice.hpp

```
#include <list>
#include <opencv2/imgproc.hpp>
```

Graphe des dépendances par inclusion de CDice.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

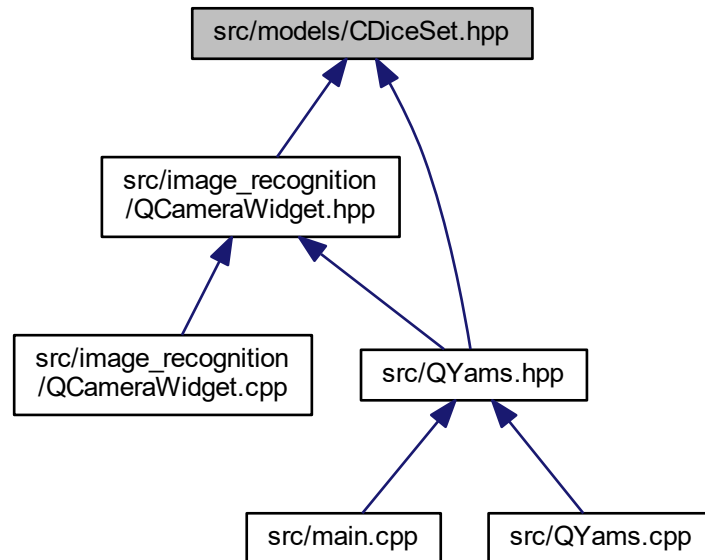


## Classes

— class `CDice`  
*Encapsule un dé détecté*

## 6.6 Référence du fichier src/models/CDiceSet.hpp

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

- class [CDiceSet](#)  
*Encapsule un jeu de dés.*

### Définitions de type

- typedef unsigned int [uint](#)

#### 6.6.1 Documentation des définitions de type

##### 6.6.1.1 uint

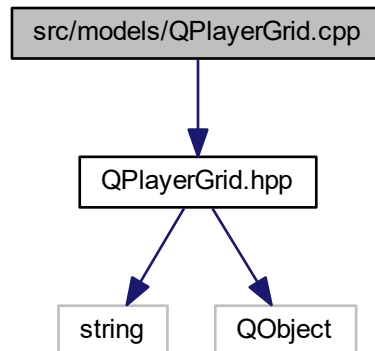
```
typedef unsigned int uint
```

Définition à la ligne 6 du fichier CDiceSet.hpp.

## 6.7 Référence du fichier src/models/QPlayerGrid.cpp

```
#include "QPlayerGrid.hpp"
```

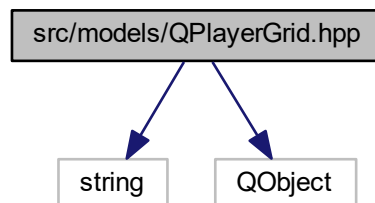
Graphe des dépendances par inclusion de QPlayerGrid.cpp:



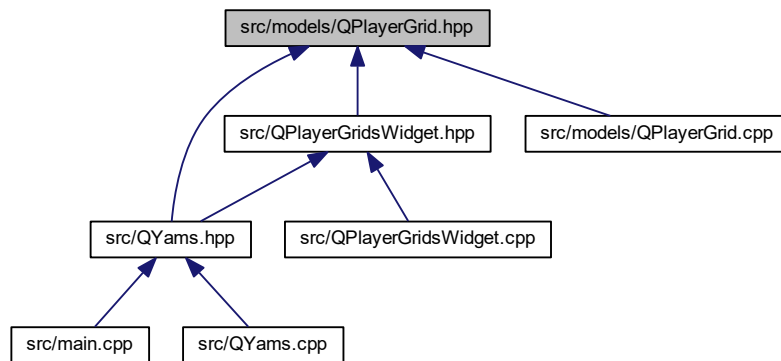
## 6.8 Référence du fichier src/models/QPlayerGrid.hpp

```
#include <string>  
#include <QObject>
```

Graphe des dépendances par inclusion de QPlayerGrid.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



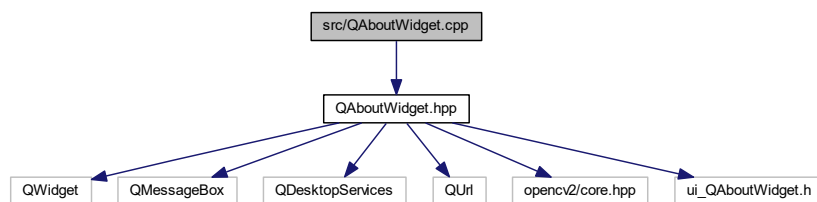
## Classes

- class `QPlayerGrid`  
Encapsule une grille de jeu.

## 6.9 Référence du fichier src/QAboutWidget.cpp

```
#include "QAboutWidget.hpp"
```

Graphe des dépendances par inclusion de QAboutWidget.cpp:

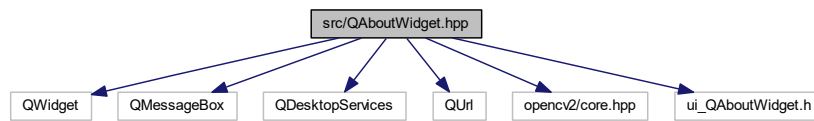


## 6.10 Référence du fichier src/QAboutWidget.hpp

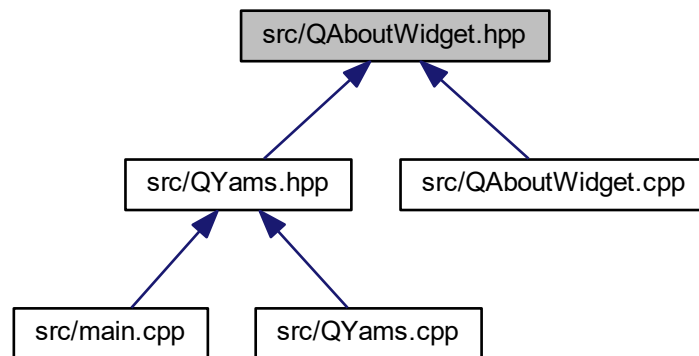
```
#include <QWidget>
#include <QMessageBox>
#include <QDesktopServices>
#include <QUrl>
#include <opencv2/core.hpp>
```

```
#include "ui_QAboutWidget.h"
```

Graphe des dépendances par inclusion de QAboutWidget.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



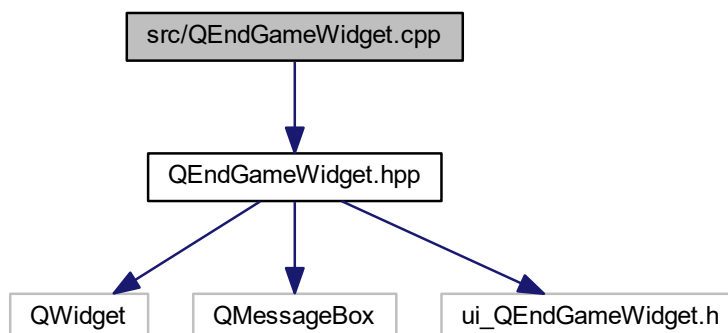
## Classes

— class [QAboutWidget](#)  
Menu À propos de l'application.

## 6.11 Référence du fichier src/QEndGameWidget.cpp

```
#include "QEndGameWidget.hpp"
```

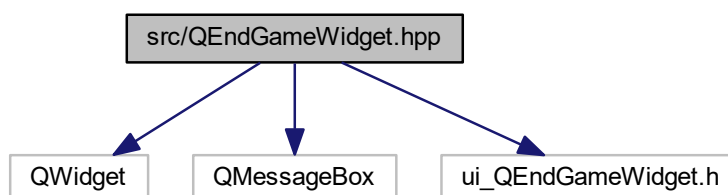
Graphe des dépendances par inclusion de QEndGameWidget.cpp:



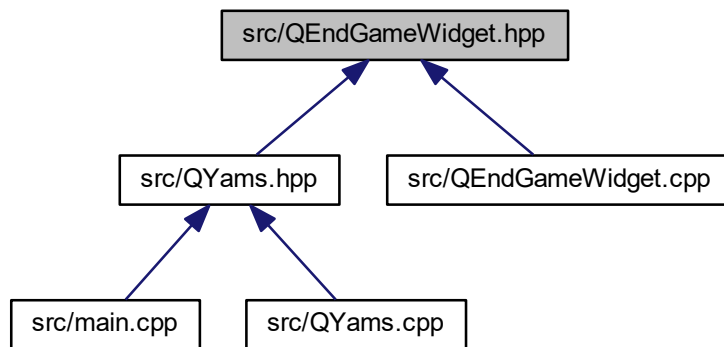
## 6.12 Référence du fichier src/QEndGameWidget.hpp

```
#include <QWidget>  
#include <QMessageBox>  
#include "ui_QEndGameWidget.h"
```

Graphe des dépendances par inclusion de QEndGameWidget.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



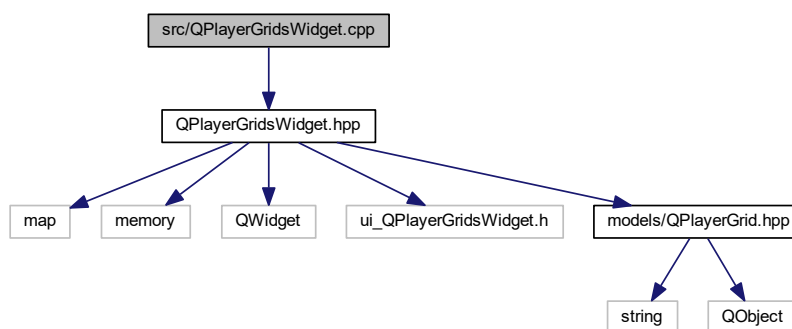
## Classes

- class [QEndGameWidget](#)  
*Menu de fin de partie.*

## 6.13 Référence du fichier `src/QPlayerGridsWidget.cpp`

```
#include "QPlayerGridsWidget.hpp"
```

Graphe des dépendances par inclusion de `QPlayerGridsWidget.cpp`:

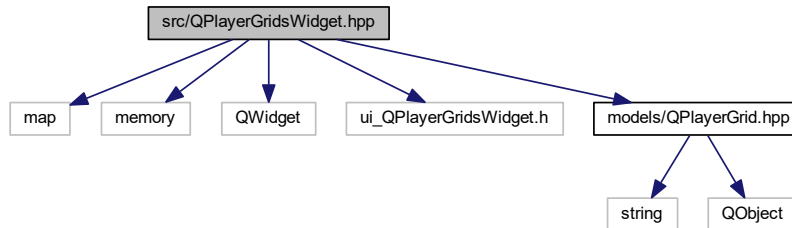


## 6.14 Référence du fichier `src/QPlayerGridsWidget.hpp`

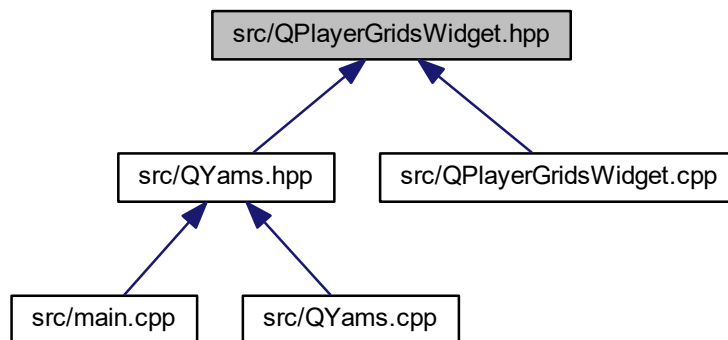
```
#include <map>
#include <memory>
```



```
#include <QWidget>
#include "ui_QPlayerGridsWidget.h"
#include "models/QPlayerGrid.hpp"
Graphe des dépendances par inclusion de QPlayerGridsWidget.hpp:
```



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class [QPlayerGridsWidget](#)  
*Vue des grilles de jeu.*
- class [QPlayerGridsWidget::CPlayerGridView](#)  
*Encapsule une vue de grille de jeu.*

## 6.15 Référence du fichier src/QYams.cpp

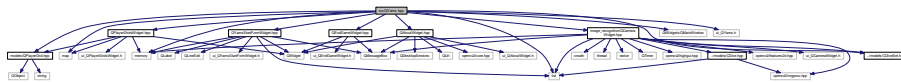
```
#include "QYams.hpp"
Graphe des dépendances par inclusion de QYams.cpp:
```



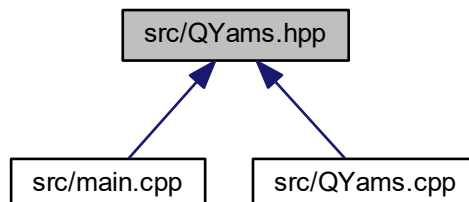
## 6.16 Référence du fichier src/QYams.hpp

```
#include <list>
#include <map>
#include <memory>
#include <QtWidgets/QMainWindow>
#include "ui_QYams.h"
#include "image_recognition/QCameraWidget.hpp"
#include "models/QPlayerGrid.hpp"
#include "models/CDiceSet.hpp"
#include "QAboutWidget.hpp"
#include "QEndGameWidget.hpp"
#include "QPlayerGridsWidget.hpp"
#include "QYamsStartFormWidget.hpp"
```

Graphe des dépendances par inclusion de QYams.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



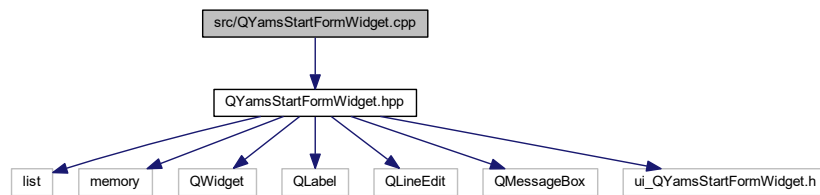
## Classes

— class [QYams](#)  
*Contrôleur du jeu Yams.*

## 6.17 Référence du fichier src/QYamsStartFormWidget.cpp

```
#include "QYamsStartFormWidget.hpp"
```

Graphe des dépendances par inclusion de QYamsStartFormWidget.cpp:



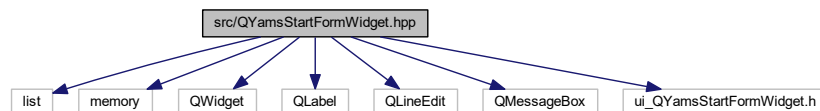
## 6.18 Référence du fichier src/QYamsStartFormWidget.hpp

```

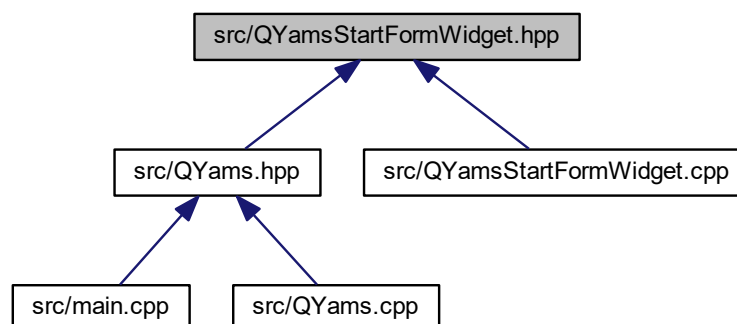
#include <list>
#include <memory>
#include <QWidget>
#include <QLabel>
#include <QLineEdit>
#include <QMessageBox>
#include "ui_QYamsStartFormWidget.h"

```

Graphe des dépendances par inclusion de QYamsStartFormWidget.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class [QYamsStartFormWidget](#)  
*Fenêtre de saisie des noms de joueur.*
- class [QYamsStartFormWidget::CPlayerForm](#)  
*Encapsule le numéro et le champ de texte pour un joueur.*

# Index

- \_beforeStart
    - QYams, 95
  - \_checkBonus
    - QPlayerGrid, 49
  - \_findDices
    - QCameraWidget, 34
  - \_findDicesByBlob
    - QCameraWidget, 35
  - \_findDicesByMinArea
    - QCameraWidget, 35
  - \_hideGameBar
    - QYams, 96
  - \_nextPlayer
    - QYams, 97
  - \_onEndGame
    - QYams, 97
  - \_resetChoices
    - QYams, 98
  - \_showRules
    - QYams, 99
  - \_simulate
    - QYams, 99
  - \_updateImage
    - QCameraWidget, 36
- ~CPlayerGridView
  - QPlayerGridsWidget::CPlayerGridView, 23
- ~QCameraWidget
  - QCameraWidget, 34
- ~QYams
  - QYams, 95
- ACES
  - QYams, 93
- actualizeActionButtons
  - QPlayerGridsWidget, 76
- addDotRect
  - CDice, 10
- addGrid
  - QPlayerGridsWidget, 79
- addPlayerNameInput
  - QYamsStartFormWidget, 113
- ANYTHING
  - QYams, 93
- BRELAN
  - QYams, 93
- cameraImgUpdated
  - QCameraWidget, 36
- CARRE
  - QYams, 93
- CDice, 9
  - addDotRect, 10
  - CDice, 10
  - getCount, 10
  - getDiceRect, 10
  - getDotRects, 11
  - m\_diceRect, 11
  - m\_IDotRects, 12
  - setDiceRect, 11
- CDiceSet, 12
  - CDiceSet, 13
  - getAces, 13
  - getFives, 14
  - getFours, 14
  - getSixes, 14
  - getThrees, 15
  - getTotal, 15
  - getTwos, 16
  - m\_iAs, 18
  - m\_iFives, 19
  - m\_iFours, 19
  - m\_iSixes, 19
  - m\_iThrees, 19
  - m\_iTwos, 19
  - operator!=, 17
  - operator==, 17
  - operator[], 18
- CDiceSet.hpp
  - uint, 121
- CHANCE
  - QYams, 94
- closeEvent
  - QYams, 100
- CPlayerForm
  - QYamsStartFormWidget::CPlayerForm, 20
- CPlayerGridView
  - QPlayerGridsWidget::CPlayerGridView, 23
- DEFAULT\_KERNEL
  - QCameraWidget, 37
- dicesUpdated
  - QCameraWidget, 36
- doAction
  - QYams, 101
- EChoices
  - QYams, 93
- enableActionButtons
  - QPlayerGridsWidget, 80

EYamsActions  
     QYams, 93

FIVES  
     QYams, 93

FOURS  
     QYams, 93

FULL  
     QYams, 93

getAces  
     CDiceSet, 13  
     QPlayerGrid, 49

getAcesButton  
     QPlayerGridsWidget, 81

getBonus  
     QPlayerGrid, 49

getBrelan  
     QPlayerGrid, 50

getBrelanButton  
     QPlayerGridsWidget, 81

getCarre  
     QPlayerGrid, 50

getCarreButton  
     QPlayerGridsWidget, 82

getChance  
     QPlayerGrid, 50

getChanceButton  
     QPlayerGridsWidget, 82

getCount  
     CDice, 10

getDiceRect  
     CDice, 10

getDotRects  
     CDice, 11

getFives  
     CDiceSet, 14  
     QPlayerGrid, 51

getFivesButton  
     QPlayerGridsWidget, 83

getFours  
     CDiceSet, 14  
     QPlayerGrid, 51

getFoursButton  
     QPlayerGridsWidget, 83

getFull  
     QPlayerGrid, 51

getFullButton  
     QPlayerGridsWidget, 84

getInputName  
     QYamsStartFormWidget::CPlayerForm, 21

getLargeStraight  
     QPlayerGrid, 52

getLargeStraightButton  
     QPlayerGridsWidget, 84

getLowerTotal  
     QPlayerGrid, 52

getPlayerName  
     QPlayerGrid, 52

getPreUpperTotal  
     QPlayerGrid, 53

getQuitButton  
     QEndGameWidget, 44

getQWidget  
     QYamsStartFormWidget::CPlayerForm, 21

getRestartButton  
     QEndGameWidget, 44

getSixes  
     CDiceSet, 14  
     QPlayerGrid, 53

getSixesButton  
     QPlayerGridsWidget, 85

getSmallStraight  
     QPlayerGrid, 53

getSmallStraightButton  
     QPlayerGridsWidget, 85

getSuperYams  
     QPlayerGrid, 54

getSuperYamsButton  
     QPlayerGridsWidget, 86

getThreeButton  
     QPlayerGridsWidget, 86

getThrees  
     CDiceSet, 15  
     QPlayerGrid, 54

getTotal  
     CDiceSet, 15  
     QPlayerGrid, 54

getTwos  
     CDiceSet, 16  
     QPlayerGrid, 55

getTwosButton  
     QPlayerGridsWidget, 87

getUpperTotal  
     QPlayerGrid, 55

getYams  
     QPlayerGrid, 56

getYamsButton  
     QPlayerGridsWidget, 87

gridUpdated  
     QPlayerGrid, 56

isAcesAlreadySet  
     QPlayerGrid, 58

isBrelanAlreadySet  
     QPlayerGrid, 58

isCarreAlreadySet  
     QPlayerGrid, 58

isChanceAlreadySet  
     QPlayerGrid, 59

isFivesAlreadySet  
     QPlayerGrid, 59

isFoursAlreadySet  
     QPlayerGrid, 60

isFullAlreadySet  
     QPlayerGrid, 60

isGridFinished  
     QPlayerGrid, 61

- isLargeStraightAlreadySet
  - QPlayerGrid, [61](#)
- isSixesAlreadySet
  - QPlayerGrid, [61](#)
- isSmallStraightAlreadySet
  - QPlayerGrid, [62](#)
- isSuperYamsAlreadySet
  - QPlayerGrid, [62](#)
- isThreesAlreadySet
  - QPlayerGrid, [63](#)
- isTwosAlreadySet
  - QPlayerGrid, [63](#)
- isYamsAlreadySet
  - QPlayerGrid, [64](#)
- LARGE\_STRAIGHT
  - QYams, [94](#)
- launchGame
  - QYams, [102](#)
- lblAs
  - QPlayerGridsWidget::CPlayerGridView, [24](#)
- lblBonus
  - QPlayerGridsWidget::CPlayerGridView, [24](#)
- lblBrelan
  - QPlayerGridsWidget::CPlayerGridView, [24](#)
- lblCarre
  - QPlayerGridsWidget::CPlayerGridView, [24](#)
- lblChance
  - QPlayerGridsWidget::CPlayerGridView, [24](#)
- lblFives
  - QPlayerGridsWidget::CPlayerGridView, [25](#)
- lblFours
  - QPlayerGridsWidget::CPlayerGridView, [25](#)
- lblFull
  - QPlayerGridsWidget::CPlayerGridView, [25](#)
- lblLargeStraight
  - QPlayerGridsWidget::CPlayerGridView, [25](#)
- lblPlayerName
  - QPlayerGridsWidget::CPlayerGridView, [25](#)
- lblPreTotalSup
  - QPlayerGridsWidget::CPlayerGridView, [26](#)
- lblSixes
  - QPlayerGridsWidget::CPlayerGridView, [26](#)
- lblSmallStraight
  - QPlayerGridsWidget::CPlayerGridView, [26](#)
- lblSuperYams
  - QPlayerGridsWidget::CPlayerGridView, [26](#)
- lblThrees
  - QPlayerGridsWidget::CPlayerGridView, [26](#)
- lblTotal
  - QPlayerGridsWidget::CPlayerGridView, [27](#)
- lblTotalInf
  - QPlayerGridsWidget::CPlayerGridView, [27](#)
- lblTotalSup
  - QPlayerGridsWidget::CPlayerGridView, [27](#)
- lblTwos
  - QPlayerGridsWidget::CPlayerGridView, [27](#)
- lblYams
  - QPlayerGridsWidget::CPlayerGridView, [27](#)
- m\_blsAcesAlreadySet
  - QPlayerGrid, [68](#)
- m\_blsBonus
  - QPlayerGrid, [68](#)
- m\_blsBrelanAlreadySet
  - QPlayerGrid, [68](#)
- m\_blsCarreAlreadySet
  - QPlayerGrid, [69](#)
- m\_blsChanceAlreadySet
  - QPlayerGrid, [69](#)
- m\_blsFivesAlreadySet
  - QPlayerGrid, [69](#)
- m\_blsFoursAlreadySet
  - QPlayerGrid, [69](#)
- m\_blsFull
  - QPlayerGrid, [69](#)
- m\_blsFullAlreadySet
  - QPlayerGrid, [70](#)
- m\_blsLargeStraight
  - QPlayerGrid, [70](#)
- m\_blsLargeStraightAlreadySet
  - QPlayerGrid, [70](#)
- m\_blsSixesAlreadySet
  - QPlayerGrid, [70](#)
- m\_blsSmallStraight
  - QPlayerGrid, [70](#)
- m\_blsSmallStraightAlreadySet
  - QPlayerGrid, [71](#)
- m\_blsSuperYams
  - QPlayerGrid, [71](#)
- m\_blsSuperYamsAlreadySet
  - QPlayerGrid, [71](#)
- m\_blsThreesAlreadySet
  - QPlayerGrid, [71](#)
- m\_blsTwosAlreadySet
  - QPlayerGrid, [71](#)
- m\_blsWrongDetection
  - QCameraWidget, [37](#)
- m\_blsYams
  - QPlayerGrid, [72](#)
- m\_blsYamsAlreadySet
  - QPlayerGrid, [72](#)
- m\_bThreadsEnabled
  - QCameraWidget, [38](#)
- m\_choice1
  - QYams, [108](#)
- m\_choice2
  - QYams, [108](#)
- m\_choice3
  - QYams, [108](#)
- m\_diceRect
  - CDice, [11](#)
- m\_HBoxLayout
  - QYamsStartFormWidget::CPlayerForm, [21](#)
- m\_iAces
  - QPlayerGrid, [72](#)
- m\_iAs
  - CDiceSet, [18](#)

- m\_iBrelan
  - QPlayerGrid, 72
- m\_iCarre
  - QPlayerGrid, 72
- m\_iChance
  - QPlayerGrid, 73
- m\_iFives
  - CDiceSet, 19
  - QPlayerGrid, 73
- m\_iFours
  - CDiceSet, 19
  - QPlayerGrid, 73
- m\_iNbrTurn
  - QYams, 108
- m\_iptrPlayerName
  - QYamsStartFormWidget::CPlayerForm, 21
- m\_iSixes
  - CDiceSet, 19
  - QPlayerGrid, 73
- m\_iThrees
  - CDiceSet, 19
  - QPlayerGrid, 73
- m\_itPlayerGrids
  - QYams, 108
- m\_iTwos
  - CDiceSet, 19
  - QPlayerGrid, 74
- m\_lblPlayerNum
  - QYamsStartFormWidget::CPlayerForm, 22
- m\_IDices
  - QCameraWidget, 38
- m\_IDotRects
  - CDice, 12
- m\_IPlayerNameInputs
  - QYamsStartFormWidget, 115
- m\_lpQPlayerGrids
  - QYams, 109
- m\_mapLastSimulation
  - QYams, 109
- m\_mapPlayerGridViews
  - QPlayerGridsWidget, 90
- m\_matImageCaptured
  - QCameraWidget, 38
- m\_pLastDiceSet
  - QCameraWidget, 38
- m\_pLastValidatedDiceSet
  - QCameraWidget, 38
- m\_ptrAboutWindow
  - QYams, 109
- m\_ptrEndGameWidget
  - QYams, 109
- m\_ptrQCameraWidget
  - QYams, 109
- m\_ptrQPlayerGridsWidget
  - QYams, 110
- m\_ptrQYamsStartForm
  - QYams, 110
- m\_qTimerFrame
  - QCameraWidget, 39
- m\_sPlayerName
  - QPlayerGrid, 74
- m\_ui
  - QAboutWidget, 31
  - QCameraWidget, 39
  - QEndGameWidget, 45
  - QPlayerGridsWidget, 90
  - QYams, 110
  - QYamsStartFormWidget, 115
- m\_videoCapture
  - QCameraWidget, 39
- m\_YamsActionsNames
  - QYams, 110
- main
  - main.cpp, 118
- main.cpp
  - main, 118
- MAX\_BLOB\_AREA
  - QCameraWidget, 39
- MAX\_BLOB\_THRESHOLD
  - QCameraWidget, 39
- MAX\_CANNY\_THRESHOLD
  - QCameraWidget, 40
- MAX\_DICE\_DOT\_WIDTH
  - QCameraWidget, 40
- MAX\_DICE\_WIDTH
  - QCameraWidget, 40
- MIN\_BLOB\_AREA
  - QCameraWidget, 40
- MIN\_BLOB\_THRESHOLD
  - QCameraWidget, 40
- MIN\_CANNY\_THRESHOLD
  - QCameraWidget, 41
- MIN\_CIRCULARITY
  - QCameraWidget, 41
- MIN\_DICE\_DOT\_WIDTH
  - QCameraWidget, 41
- MIN\_DICE\_WIDTH
  - QCameraWidget, 41
- MIN\_DISTANCE\_FROM\_DICE\_CENTER
  - QCameraWidget, 41
- MIN\_INERTIA\_RATIO
  - QCameraWidget, 42
- NOTHING
  - QYams, 93
- ONE
  - QYams, 93
- onPlayerNameSetUp
  - QYamsStartFormWidget, 114
- onWrongDetection
  - QCameraWidget, 37
- operator!=
  - CDiceSet, 17
- operator==
  - CDiceSet, 17
- operator[]



- CDiceSet, 18
- playerNameSetUp
  - QYamsStartFormWidget, 114
- playerUpdated
  - QYams, 103
- QAboutWidget, 28
  - m\_ui, 31
  - QAboutWidget, 29
  - showAboutOpenCVWindow, 30
  - showAboutQtWindow, 30
- QCameraWidget, 31
  - \_findDices, 34
  - \_findDicesByBlob, 35
  - \_findDicesByMinArea, 35
  - \_updateImage, 36
  - ~QCameraWidget, 34
  - cameraImgUpdated, 36
  - DEFAULT\_KERNEL, 37
  - dicesUpdated, 36
  - m\_bIsWrongDetection, 37
  - m\_bThreadsEnabled, 38
  - m\_IDices, 38
  - m\_matImageCaptured, 38
  - m\_pLastDiceSet, 38
  - m\_pLastValidatedDiceSet, 38
  - m\_qTimerFrame, 39
  - m\_ui, 39
  - m\_videoCapture, 39
  - MAX\_BLOB\_AREA, 39
  - MAX\_BLOB\_THRESHOLD, 39
  - MAX\_CANNY\_THRESHOLD, 40
  - MAX\_DICE\_DOT\_WIDTH, 40
  - MAX\_DICE\_WIDTH, 40
  - MIN\_BLOB\_AREA, 40
  - MIN\_BLOB\_THRESHOLD, 40
  - MIN\_CANNY\_THRESHOLD, 41
  - MIN\_CIRCULARITY, 41
  - MIN\_DICE\_DOT\_WIDTH, 41
  - MIN\_DICE\_WIDTH, 41
  - MIN\_DISTANCE\_FROM\_DICE\_CENTER, 41
  - MIN\_INERTIA\_RATIO, 42
  - onWrongDetection, 37
  - QCameraWidget, 33
  - SAME\_POS\_TOLERANCE, 42
- QEndGameWidget, 42
  - getQuitButton, 44
  - getRestartButton, 44
  - m\_ui, 45
  - QEndGameWidget, 43
- QPlayerGrid, 45
  - \_checkBonus, 49
  - getAces, 49
  - getBonus, 49
  - getBrelan, 50
  - getCarre, 50
  - getChance, 50
  - getFives, 51
  - getFours, 51
  - getFull, 51
  - getLargeStraight, 52
  - getLowerTotal, 52
  - getPlayerName, 52
  - getPreUpperTotal, 53
  - getSixes, 53
  - getSmallStraight, 53
  - getSuperYams, 54
  - getThrees, 54
  - getTotal, 54
  - getTwos, 55
  - getUpperTotal, 55
  - getYams, 56
  - gridUpdated, 56
  - isAcesAlreadySet, 58
  - isBrelanAlreadySet, 58
  - isCarreAlreadySet, 58
  - isChanceAlreadySet, 59
  - isFivesAlreadySet, 59
  - isFoursAlreadySet, 60
  - isFullAlreadySet, 60
  - isGridFinished, 61
  - isLargeStraightAlreadySet, 61
  - isSixesAlreadySet, 61
  - isSmallStraightAlreadySet, 62
  - isSuperYamsAlreadySet, 62
  - isThreesAlreadySet, 63
  - isTwosAlreadySet, 63
  - isYamsAlreadySet, 64
  - m\_bIsAcesAlreadySet, 68
  - m\_bIsBonus, 68
  - m\_bIsBrelanAlreadySet, 68
  - m\_bIsCarreAlreadySet, 69
  - m\_bIsChanceAlreadySet, 69
  - m\_bIsFivesAlreadySet, 69
  - m\_bIsFoursAlreadySet, 69
  - m\_bIsFull, 69
  - m\_bIsFullAlreadySet, 70
  - m\_bIsLargeStraight, 70
  - m\_bIsLargeStraightAlreadySet, 70
  - m\_bIsSixesAlreadySet, 70
  - m\_bIsSmallStraight, 70
  - m\_bIsSmallStraightAlreadySet, 71
  - m\_bIsSuperYams, 71
  - m\_bIsSuperYamsAlreadySet, 71
  - m\_bIsThreesAlreadySet, 71
  - m\_bIsTwosAlreadySet, 71
  - m\_bIsYams, 72
  - m\_bIsYamsAlreadySet, 72
  - m\_iAces, 72
  - m\_iBrelan, 72
  - m\_iCarre, 72
  - m\_iChance, 73
  - m\_iFives, 73
  - m\_iFours, 73
  - m\_iSixes, 73
  - m\_iThrees, 73

- m\_iTwos, 74
- m\_sPlayerName, 74
- QPlayerGrid, 48
- setAces, 64
- setBrelan, 65
- setCarre, 65
- setChance, 65
- setFives, 65
- setFours, 66
- setFull, 66
- setLargeStraight, 66
- setSixes, 66
- setSmallStraight, 67
- setSuperYams, 67
- setThrees, 67
- setTwos, 67
- setYams, 68
- QPlayerGridsWidget, 74
  - actualizeActionButtons, 76
  - addGrid, 79
  - enableActionButtons, 80
  - getAcesButton, 81
  - getBrelanButton, 81
  - getCarreButton, 82
  - getChanceButton, 82
  - getFivesButton, 83
  - getFoursButton, 83
  - getFullButton, 84
  - getLargeStraightButton, 84
  - getSixesButton, 85
  - getSmallStraightButton, 85
  - getSuperYamsButton, 86
  - getThreeButton, 86
  - getTwosButton, 87
  - getYamsButton, 87
  - m\_mapPlayerGridViews, 90
  - m\_ui, 90
  - QPlayerGridsWidget, 76
  - removeGrid, 88
  - updateGrid, 88
- QPlayerGridsWidget::CPlayerGridView, 22
  - ~CPlayerGridView, 23
  - CPlayerGridView, 23
  - lblAs, 24
  - lblBonus, 24
  - lblBrelan, 24
  - lblCarre, 24
  - lblChance, 24
  - lblFives, 25
  - lblFours, 25
  - lblFull, 25
  - lblLargeStraight, 25
  - lblPlayerName, 25
  - lblPreTotalSup, 26
  - lblSixes, 26
  - lblSmallStraight, 26
  - lblSuperYams, 26
  - lblThrees, 26
  - lblTotal, 27
  - lblTotalInf, 27
  - lblTotalSup, 27
  - lblTwos, 27
  - lblYams, 27
- quit
  - QYams, 104
- QYams, 90
  - \_beforeStart, 95
  - \_hideGameBar, 96
  - \_nextPlayer, 97
  - \_onEndGame, 97
  - \_resetChoices, 98
  - \_showRules, 99
  - \_simulate, 99
  - ~QYams, 95
  - ACES, 93
  - ANYTHING, 93
  - BRELAN, 93
  - CARRE, 93
  - CHANCE, 94
  - closeEvent, 100
  - doAction, 101
  - EChoices, 93
  - EYamsActions, 93
  - FIVES, 93
  - FOURS, 93
  - FULL, 93
  - LARGE\_STRAIGHT, 94
  - launchGame, 102
  - m\_choice1, 108
  - m\_choice2, 108
  - m\_choice3, 108
  - m\_iNbrTurn, 108
  - m\_itPlayerGrids, 108
  - m\_lpQPlayerGrids, 109
  - m\_mapLastSimulation, 109
  - m\_ptrAboutWindow, 109
  - m\_ptrEndGameWidget, 109
  - m\_ptrQCameraWidget, 109
  - m\_ptrQPlayerGridsWidget, 110
  - m\_ptrQYamsStartFrom, 110
  - m\_ui, 110
  - m\_YamsActionsNames, 110
  - NOTHING, 93
  - ONE, 93
  - playerUpdated, 103
  - quit, 104
  - QYams, 94
  - restart, 104
  - showAboutWindow, 105
  - SIXES, 93
  - SMALL\_STRAIGHT, 93
  - start, 106
  - SUPER\_YAMS, 94
  - THREE, 93
  - THREES, 93
  - TWO, 93

- TWOS, [93](#)
  - updateTurn, [107](#)
  - YAMS, [94](#)
- QYamsStartFormWidget, [111](#)
  - addPlayerNameInput, [113](#)
  - m\_IPlayerNameInputs, [115](#)
  - m\_ui, [115](#)
  - onPlayerNameSetUp, [114](#)
  - playerNameSetUp, [114](#)
  - QYamsStartFormWidget, [113](#)
- QYamsStartFormWidget::CPlayerForm, [20](#)
  - CPlayerForm, [20](#)
  - getInputName, [21](#)
  - getQWidget, [21](#)
  - m\_HBoxLayout, [21](#)
  - m\_ipPlayerName, [21](#)
  - m\_lblPlayerNum, [22](#)
- removeGrid
  - QPlayerGridsWidget, [88](#)
- restart
  - QYams, [104](#)
- SAME\_POS\_TOLERANCE
  - QCameraWidget, [42](#)
- setAces
  - QPlayerGrid, [64](#)
- setBrelan
  - QPlayerGrid, [65](#)
- setCarre
  - QPlayerGrid, [65](#)
- setChance
  - QPlayerGrid, [65](#)
- setDiceRect
  - CDice, [11](#)
- setFives
  - QPlayerGrid, [65](#)
- setFours
  - QPlayerGrid, [66](#)
- setFull
  - QPlayerGrid, [66](#)
- setLargeStraight
  - QPlayerGrid, [66](#)
- setSixes
  - QPlayerGrid, [66](#)
- setSmallStraight
  - QPlayerGrid, [67](#)
- setSuperYams
  - QPlayerGrid, [67](#)
- setThrees
  - QPlayerGrid, [67](#)
- setTwos
  - QPlayerGrid, [67](#)
- setYams
  - QPlayerGrid, [68](#)
- showAboutOpenCVWindow
  - QAboutWidget, [30](#)
- showAboutQtWindow
  - QAboutWidget, [30](#)
- showAboutWindow
  - QYams, [105](#)
- SIXES
  - QYams, [93](#)
- SMALL\_STRAIGHT
  - QYams, [93](#)
- src/image\_recognition/QCameraWidget.cpp, [117](#)
- src/image\_recognition/QCameraWidget.hpp, [117](#)
- src/main.cpp, [118](#)
- src/models/CDice.cpp, [119](#)
- src/models/CDice.hpp, [119](#)
- src/models/CDiceSet.hpp, [121](#)
- src/models/QPlayerGrid.cpp, [122](#)
- src/models/QPlayerGrid.hpp, [122](#)
- src/QAboutWidget.cpp, [123](#)
- src/QAboutWidget.hpp, [123](#)
- src/QEndGameWidget.cpp, [124](#)
- src/QEndGameWidget.hpp, [125](#)
- src/QPlayerGridsWidget.cpp, [126](#)
- src/QPlayerGridsWidget.hpp, [126](#)
- src/QYams.cpp, [127](#)
- src/QYams.hpp, [128](#)
- src/QYamsStartFormWidget.cpp, [128](#)
- src/QYamsStartFormWidget.hpp, [129](#)
- start
  - QYams, [106](#)
- SUPER\_YAMS
  - QYams, [94](#)
- THREE
  - QYams, [93](#)
- THREES
  - QYams, [93](#)
- TWO
  - QYams, [93](#)
- TWOS
  - QYams, [93](#)
- uint
  - CDiceSet.hpp, [121](#)
- updateGrid
  - QPlayerGridsWidget, [88](#)
- updateTurn
  - QYams, [107](#)
- YAMS
  - QYams, [94](#)