

## Jeux de déduction

Generated by Doxygen 1.8.17



<b>1</b>	<b>Projet Cpp S8</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	std Namespace Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	CombiMastermind Class Reference . . . . .	11
6.1.1	Detailed Description . . . . .	12
6.1.2	Constructor & Destructor Documentation . . . . .	12
6.1.2.1	CombiMastermind() . . . . .	12
6.1.3	Member Function Documentation . . . . .	13
6.1.3.1	resultat() . . . . .	13
6.1.3.2	toString() . . . . .	13
6.2	Combinaison Class Reference . . . . .	15
6.2.1	Detailed Description . . . . .	15
6.2.2	Constructor & Destructor Documentation . . . . .	16
6.2.2.1	Combinaison() . . . . .	16
6.2.3	Member Function Documentation . . . . .	16
6.2.3.1	get() . . . . .	16
6.2.3.2	resultat() . . . . .	16
6.2.3.3	setCombinaison() . . . . .	17
6.2.3.4	toString() . . . . .	17
6.3	CombiWordle Class Reference . . . . .	18
6.3.1	Detailed Description . . . . .	19
6.3.2	Constructor & Destructor Documentation . . . . .	19
6.3.2.1	CombiWordle() [1/2] . . . . .	19
6.3.2.2	CombiWordle() [2/2] . . . . .	19
6.3.3	Member Function Documentation . . . . .	20
6.3.3.1	afficherResultat() . . . . .	20
6.3.3.2	resultat() . . . . .	20
6.3.3.3	resultat_couleur() . . . . .	21
6.3.3.4	toString() . . . . .	22
6.4	FonctionsUtiles Class Reference . . . . .	23
6.4.1	Detailed Description . . . . .	24

6.4.2 Member Function Documentation	24
6.4.2.1 blanc()	24
6.4.2.2 bleu()	25
6.4.2.3 blink()	25
6.4.2.4 carreblanc()	26
6.4.2.5 carrebleu()	26
6.4.2.6 carrejaune()	27
6.4.2.7 carremarron()	27
6.4.2.8 carreorange()	28
6.4.2.9 carrerose()	28
6.4.2.10 carrerouge()	29
6.4.2.11 carrevert()	29
6.4.2.12 carrevide()	30
6.4.2.13 carreviolet()	30
6.4.2.14 charToString()	31
6.4.2.15 checkMotFichier()	31
6.4.2.16 cyan()	32
6.4.2.17 equals()	33
6.4.2.18 intToString()	33
6.4.2.19 isNumber()	34
6.4.2.20 jaune()	35
6.4.2.21 rouge()	35
6.4.2.22 split()	36
6.4.2.23 toCarre()	37
6.4.2.24 vert()	37
6.4.2.25 vertclair()	38
6.4.2.26 violet()	38
6.4.2.27 violetclair()	39
6.5 Humain Class Reference	39
6.5.1 Detailed Description	40
6.5.2 Constructor & Destructor Documentation	41
6.5.2.1 Humain()	41
6.5.3 Member Function Documentation	41
6.5.3.1 verifierEntree()	41
6.6 Humain_Codeur Class Reference	42
6.6.1 Detailed Description	43
6.6.2 Member Function Documentation	44
6.6.2.1 entrerMdP()	44
6.6.2.2 getMdP()	44
6.7 Humain_Codeur_Mastermind Class Reference	44
6.7.1 Detailed Description	46
6.7.2 Member Function Documentation	46

6.7.2.1 entrerCode()	46
6.7.2.2 verifierEntree()	47
6.8 Humain_Codeur_Wordle Class Reference	48
6.8.1 Detailed Description	50
6.8.2 Member Function Documentation	50
6.8.2.1 entrerCode()	50
6.8.2.2 verifierEntree()	51
6.9 Humain_Decodeur Class Reference	52
6.9.1 Detailed Description	53
6.10 Humain_Decodeur_Mastermind Class Reference	54
6.10.1 Detailed Description	55
6.10.2 Member Function Documentation	56
6.10.2.1 entrerCombinaison()	56
6.10.2.2 verifierEntree()	56
6.11 Humain_Decodeur_Wordle Class Reference	57
6.11.1 Detailed Description	58
6.11.2 Member Function Documentation	59
6.11.2.1 verifierEntree()	59
6.12 IA Class Reference	60
6.12.1 Detailed Description	61
6.12.2 Constructor & Destructor Documentation	61
6.12.2.1 IA()	61
6.12.3 Member Function Documentation	61
6.12.3.1 choisirCombinaison()	61
6.13 IA_Codeur Class Reference	62
6.13.1 Detailed Description	63
6.13.2 Member Function Documentation	64
6.13.2.1 genererCode()	64
6.13.2.2 genererMdP()	64
6.13.2.3 getMdP()	64
6.14 IA_Codeur_Mastermind Class Reference	65
6.14.1 Detailed Description	66
6.14.2 Member Function Documentation	67
6.14.2.1 choisirCombinaison()	67
6.14.2.2 genererCode()	68
6.15 IA_Codeur_Wordle Class Reference	68
6.15.1 Detailed Description	70
6.15.2 Member Function Documentation	70
6.15.2.1 choisirCombinaison()	70
6.15.2.2 genererCode()	71
6.16 IA_Decodeur Class Reference	72
6.16.1 Detailed Description	73

---

6.16.2 Member Function Documentation	74
6.16.2.1 Combi_possible()	74
6.17 IA_Decodeur_Mastermind Class Reference	74
6.17.1 Detailed Description	75
6.17.2 Member Function Documentation	76
6.17.2.1 choisirCombinaison()	76
6.17.2.2 Combi_possible()	77
6.18 IA_Decodeur_Wordle Class Reference	77
6.18.1 Detailed Description	78
6.18.2 Member Function Documentation	79
6.18.2.1 choisirCombinaison()	79
6.18.2.2 Combi_possible()	80
6.18.2.3 Maj_ensemble()	80
6.19 JeuDeDeduction Class Reference	81
6.19.1 Detailed Description	83
6.19.2 Constructor & Destructor Documentation	83
6.19.2.1 JeuDeDeduction()	83
6.19.3 Member Function Documentation	83
6.19.3.1 detectionVictoire()	83
6.19.3.2 getTour()	84
6.20 Joueur Class Reference	85
6.20.1 Detailed Description	86
6.20.2 Constructor & Destructor Documentation	86
6.20.2.1 Joueur()	86
6.20.3 Member Function Documentation	87
6.20.3.1 getCombinaison()	87
6.20.3.2 getPseudo()	87
6.20.3.3 set_historiqueRes()	87
6.20.3.4 setCombinaison()	88
6.20.3.5 setPseudo()	88
6.20.3.6 toString()	89
6.21 Mastermind Class Reference	89
6.21.1 Detailed Description	90
6.21.2 Constructor & Destructor Documentation	91
6.21.2.1 Mastermind()	91
6.21.3 Member Function Documentation	91
6.21.3.1 partie()	91
6.22 Menu Class Reference	92
6.22.1 Detailed Description	94
6.22.2 Constructor & Destructor Documentation	94
6.22.2.1 Menu()	94
6.22.3 Member Function Documentation	94

---

---

6.22.3.1 appartientDico()	94
6.22.3.2 appartientEE()	95
6.22.3.3 choisirJeu()	95
6.22.3.4 parametreDeJeu()	95
6.23 UnitTest Class Reference	96
6.24 Wordle Class Reference	97
6.24.1 Detailed Description	98
6.24.2 Constructor & Destructor Documentation	99
6.24.2.1 Wordle()	99
6.24.3 Member Function Documentation	99
6.24.3.1 afficherHistorique()	99
<b>Index</b>	<b>101</b>





## Chapter 1

# Projet Cpp S8

Réalisation de jeux de déduction type [Mastermind](#) et [Wordle](#) avec implémentation [IA](#).



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">std</a> . . . . .	9
-------------------------------	---



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

Combinaison . . . . .	15
CombiMastermind . . . . .	11
CombiWordle . . . . .	18
FonctionsUtiles . . . . .	23
JeuDeDeduction . . . . .	81
Mastermind . . . . .	89
Wordle . . . . .	97
Joueur . . . . .	85
Humain . . . . .	39
Humain_Codeur . . . . .	42
Humain_Codeur_Mastermind . . . . .	44
Humain_Codeur_Wordle . . . . .	48
Humain_Decodeur . . . . .	52
Humain_Decodeur_Mastermind . . . . .	54
Humain_Decodeur_Wordle . . . . .	57
IA . . . . .	60
IA_Codeur . . . . .	62
IA_Codeur_Mastermind . . . . .	65
IA_Codeur_Wordle . . . . .	68
IA_Decodeur . . . . .	72
IA_Decodeur_Mastermind . . . . .	74
IA_Decodeur_Wordle . . . . .	77
Menu . . . . .	92
TestFixture	
UnitTest . . . . .	96



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">CombiMastermind</a>	
Tableau d'entier représentant une combinaison de couleur . . . . .	11
<a href="#">Combinaison</a>	
Tableau d'entier représentant une combinaison de couleur ou de lettre . . . . .	15
<a href="#">CombiWordle</a>	
Tableau d'entier représentant une combinaison de lettre . . . . .	18
<a href="#">FonctionsUtiles</a>	
Bibliothèque regroupant des fonctions utiles . . . . .	23
<a href="#">Humain</a>	
Si le joueur est un humain . . . . .	39
<a href="#">Humain_Codeur</a>	
Cette classe abstraite permet de décrire les joueurs humains en mode codeur peu importe le mode de jeu ( <a href="#">Mastermind</a> ou wordle) . . . . .	42
<a href="#">Humain_Codeur_Mastermind</a>	
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu <a href="#">Mastermind</a> . . . . .	44
<a href="#">Humain_Codeur_Wordle</a>	
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu <a href="#">Wordle</a> . . . . .	48
<a href="#">Humain_Decodeur</a>	
Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu ( <a href="#">Mastermind</a> ou wordle) . . . . .	52
<a href="#">Humain_Decodeur_Mastermind</a>	
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu <a href="#">Mastermind</a> . . . . .	54
<a href="#">Humain_Decodeur_Wordle</a>	
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu <a href="#">Wordle</a> . . . . .	57
<a href="#">IA</a>	
Si le joueur est un <a href="#">IA</a> . . . . .	60
<a href="#">IA_Codeur</a>	
Cette classe abstraite permet de décrire l'intelligence artificielle en mode codeur peu importe le mode de jeu ( <a href="#">Mastermind</a> ou wordle) . . . . .	62
<a href="#">IA_Codeur_Mastermind</a>	
Cette classe est utilisée pour représenter un joueur artificiel ( <a href="#">IA</a> ) en mode codeur dans le mode de jeu <a href="#">Mastermind</a> . . . . .	65

<a href="#">IA_Codeur_Wordle</a>	Cette classe est utilisée pour représenter un joueur virtuel ( <a href="#">IA</a> ) en mode codeur dans le mode de jeu <a href="#">Wordle</a> . . . . .	68
<a href="#">IA_Decodeur</a>	Cette classe abstraite permet de décrire l'intelligence artificielle en mode Decodeur peu importe le mode de jeu ( <a href="#">Mastermind</a> ou <a href="#">wordle</a> ) . . . . .	72
<a href="#">IA_Decodeur_Mastermind</a>	Cette classe est utilisée pour représenter un joueur artificiel ( <a href="#">IA</a> ) en mode Decodeur dans le mode de jeu <a href="#">Mastermind</a> . . . . .	74
<a href="#">IA_Decodeur_Wordle</a>	Cette classe est utilisée pour représenter un joueur virtuel ( <a href="#">IA</a> ) en mode Decodeur dans le mode de jeu <a href="#">Wordle</a> . . . . .	77
<a href="#">JeuDeDeduction</a>	Classe principale permettant de lancement d'une partie. Elle comprend le main et . . . . .	81
<a href="#">Joueur</a>	Classe abstraite regroupant tous les joueurs (humain et <a href="#">IA</a> ) . . . . .	85
<a href="#">Mastermind</a>	Classe principale permettant de lancement d'une partie de <a href="#">Mastermind</a> . Elle définit les méthodes abstraites présente dans <a href="#">JeuDeDeduction</a> . . . . .	89
<a href="#">Menu</a>	La classe <a href="#">Menu</a> permet à l'utilisateur de choisir le mode et les parametres du jeu . . . . .	92
<a href="#">UnitTest</a>	. . . . .	96
<a href="#">Wordle</a>	Classe principale permettant de lancement d'une partie de <a href="#">Wordle</a> . Elle définit les méthodes abstraites présente dans <a href="#">JeuDeDeduction</a> . . . . .	97



## Chapter 5

# Namespace Documentation

### 5.1 std Namespace Reference

#### 5.1.1 Detailed Description

Cette classe, héritée de [Combinaison](#), permet de définir l'objet [CombiMastermind](#). Elle transforme une chaîne de caractère (suite de couleur) séparée d'espace en un vecteur de string. Elle possède un seul attribut.

Cette classe permet de définir l'objet [Combinaison](#). Elle transforme une chaîne de caractère (suite de couleur ou de lettre) séparée d'espace en un vecteur de string. Elle possède un seul attribut.

Cette classe, héritée de [Combinaison](#), permet de définir l'objet [CombiWordle](#). Elle transforme une chaîne de caractère (suite de lettre) séparée d'espace en un vecteur de string.

Cette classe permet de regrouper des fonctions utiles.

Classe abstraite représentant un joueur humain codeur ou bien décodeur.

Cette classe hérite de la classe [Joueur](#), elle représente l'ordi/IA qui devra proposer une combinaison à trouver pour le joueur décodeur. Elle possède les mêmes attributs que [Joueur](#) avec un argument en plus, le mot de passe qui sera demandé au moment de voir la combinaison cherchée s'il le souhaite durant une partie.

Cette classe permet de définir un jeu de déduction dans son intégralité, elle a comme attribut les joueurs, la combinaison et le menu pour récupérer les constantes.



## Chapter 6

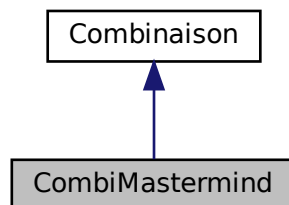
# Class Documentation

### 6.1 CombiMastermind Class Reference

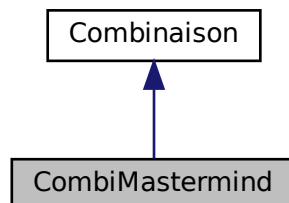
Tableau d'entier représentant une combinaison de couleur.

```
#include <CombiMastermind.hpp>
```

Inheritance diagram for CombiMastermind:



Collaboration diagram for CombiMastermind:



## Public Member Functions

- [CombiMastermind](#) ()  
*Constructeur neutre de [CombiMastermind](#).*
- virtual [~CombiMastermind](#) ()  
*Destructeur neutre de [CombiMastermind](#).*
- **CombiMastermind** (const string &chaîne)
- [CombiMastermind](#) (vector< string > v)  
*Constructeur de la classe [CombiMastermind](#).*
- **CombiMastermind** ([Combinaison](#) c)
- virtual string [resultat](#) (const [Combinaison](#) code)  
*elle permet de retourner le resultat après une tentative*
- string [toString](#) ()  
*permet l'affichage de la combinaison de couleur*

## Additional Inherited Members

### 6.1.1 Detailed Description

Tableau d'entier représentant une combinaison de couleur.

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 CombiMastermind()

```
CombiMastermind::CombiMastermind (
    vector< string > v )
```

Constructeur de la classe [CombiMastermind](#).

#### Parameters

<b>v</b>	est un vecteur de chaîne de caractère comprenant la suite de couleur séparée d'espace
----------	---

## 6.1.3 Member Function Documentation

### 6.1.3.1 resultat()

```
CombiMastermind::resultat (
    const Combinaison code ) [virtual]
```

elle permet de retourner le resultat aorès une tentative

#### Parameters

<i>code</i>	atibut de type combinaison
-------------	----------------------------

#### Returns

une chaîne de caractère

Reimplemented from [Combinaison](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.1.3.2 toString()

```
string CombiMastermind::toString ( ) [virtual]
```

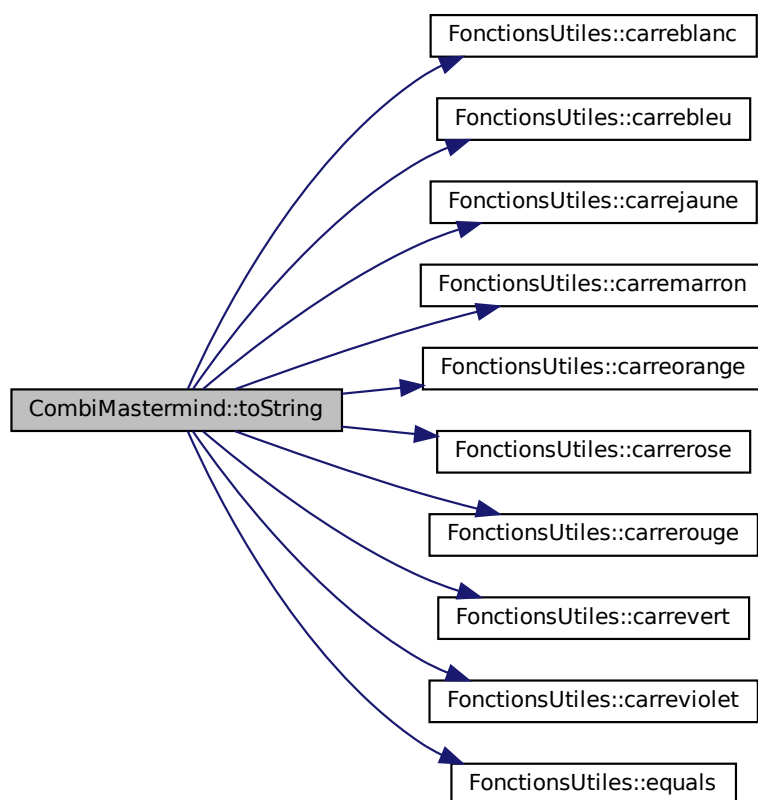
permet l'affichage de la combinaison de couleur

**Returns**

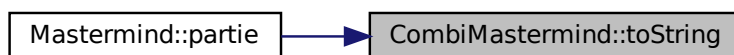
un string

Reimplemented from [Combinaison](#).

Here is the call graph for this function:



Here is the caller graph for this function:



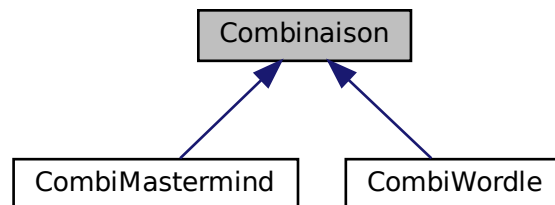
The documentation for this class was generated from the following files:

- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.cpp`

## 6.2 Combinaison Class Reference

Tableau d'entier représentant une combinaison de couleur ou de lettre.

Inheritance diagram for Combinaison:



### Public Member Functions

- [Combinaison](#) ()  
*Constructeur neutre de [Combinaison](#).*
- **Combinaison** (vector< string > v)
- void [setCombinaison](#) (vector< string > c)  
*setteur de l'attribut combinaison*
- vector< string > [get](#) () const  
*getteur de l'attribut combinaison*
- string **get** (const int i) const
- virtual string [resultat](#) ([Combinaison](#) code)  
*elle permet de retourner le resultat après une tentative*
- virtual string [toString](#) ()  
*permet l'affichage de la combinaison*

### Protected Attributes

- vector< string > **combinaison**

### 6.2.1 Detailed Description

Tableau d'entier représentant une combinaison de couleur ou de lettre.

Author

Groupe A7

Version

1.0

Date

avril 2022





## Parameters

<i>code</i>	attribut de type combinaison
-------------	------------------------------

## Returns

une chaîne de caractère

Reimplemented in [CombiWordle](#), and [CombiMastermind](#).

## 6.2.3.3 setCombinaison()

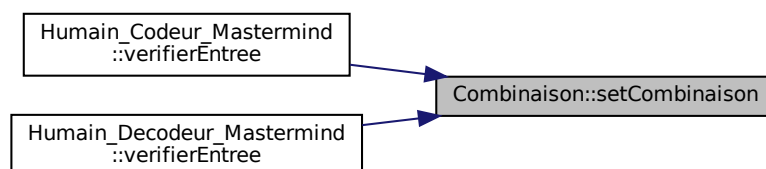
```
void Combinaison::setCombinaison (
    vector< string > c )
```

setteur de l'attribut combinaison

## Parameters

<i>c</i>	: un vecteur de chaîne de caractères
----------	--------------------------------------

Here is the caller graph for this function:



## 6.2.3.4 toString()

```
string Combinaison::toString ( ) [virtual]
```

permet l'affichage de la combinaison

## Returns

un string

Reimplemented in [CombiWordle](#), and [CombiMastermind](#).

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

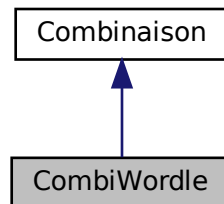
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Combinaison.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Combinaison.cpp

## 6.3 CombiWordle Class Reference

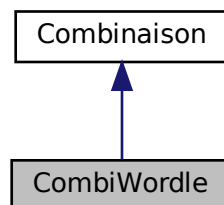
Tableau d'entier représentant une combinaison de lettre.

```
#include <CombiWordle.hpp>
```

Inheritance diagram for CombiWordle:



Collaboration diagram for CombiWordle:



### Public Member Functions

- [CombiWordle](#) ()  
*Constructeur neutre de [CombiWordle](#).*
- virtual [~CombiWordle](#) ()  
*Destructeur neutre de [CombiWordle](#).*
- **CombiWordle** (const string &chaîne)
- [CombiWordle](#) (vector< string > v)  
*Constructeur de la classe [CombiWordle](#) à partir d'un vecteur.*
- [CombiWordle](#) ([Combinaison](#) c)  
*Constructeur de la classe [CombiWordle](#) à partir d'une combinaison.*
- virtual string [resultat](#) (const [Combinaison](#) code)  
*renvoie les indices des lettres placée*
- string [resultat\\_couleur](#) (const [Combinaison](#) code)

- void `afficherResultat` (string `resultat`)

*Affiche les caractères rentrés par le décodeur avec un code de couleur précis : Vert le caractère est dans la bonne place , Jaune le caractère est dans la mauvaise place , Rouge le caractère existe pas dans le code.*

- string `toString` ()

*permet l'affichage du mot*

## Additional Inherited Members

### 6.3.1 Detailed Description

Tableau d'entier représentant une combinaison de lettre.

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 CombiWordle() [1/2]

```
CombiWordle::CombiWordle (
    vector< string > v )
```

Constructeur de la classe `CombiWordle` à partir d'un vecteur.

#### Parameters

<code>v</code>	est un vecteur contenant tous les caractères d'un mot
----------------	---

#### 6.3.2.2 CombiWordle() [2/2]

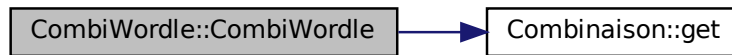
```
CombiWordle::CombiWordle (
    Combinaison c )
```

Constructeur de la classe `CombiWordle` à partir d'une combinaison.

**Parameters**

<code>c</code>	est une combinaison correspondant à un mot
----------------	--

Here is the call graph for this function:



### 6.3.3 Member Function Documentation

#### 6.3.3.1 afficherResultat()

```
CombiWordle::afficherResultat (
    string resultat )
```

Affiche les caractères rentrés par le décodeur avec un code de couleur précis : Vert le caractère est dans la bonne place , Jaune le caractère est dans la mauvaise place , Rouge le caractère existe pas dans le code.

**Parameters**

<code>un</code>	string correspondant au résultat de la vérification du mot rentré par le décodeur
-----------------	---

#### 6.3.3.2 resultat()

```
CombiWordle::resultat (
    const Combinaison code ) [virtual]
```

renvoie les indices des lettres placée

**Parameters**

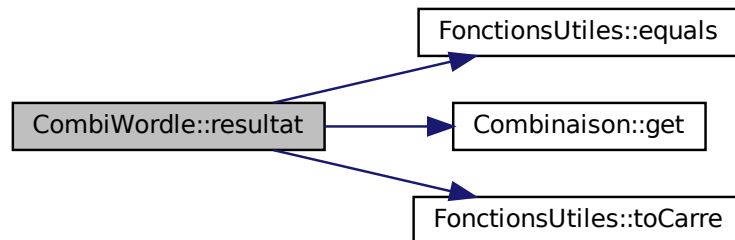
<code>code</code>	<code>Combinaison</code> : le mot recherché par le décodeur
-------------------	---

**Returns**

une chaîne de caractère

Reimplemented from [Combinaison](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.3.3.3 resultat\_couleur()

```
CombiWordle::resultat_couleur (
    const Combinaison code )
```

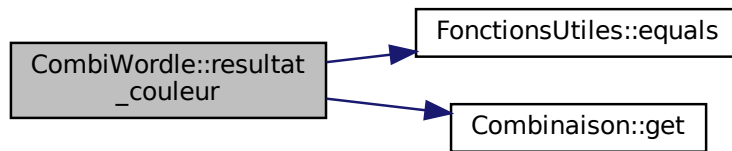
#### Parameters

<i>code</i>	<a href="#">Combinaison</a> : le mot recherché par le décodeur
-------------	--

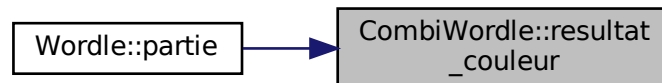
**Returns**

une chaîne de caractère

Here is the call graph for this function:



Here is the caller graph for this function:

**6.3.3.4 toString()**

```
string CombiWordle::toString ( ) [virtual]
```

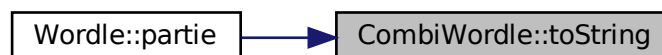
permet l'affichage du mot

**Returns**

un string

Reimplemented from [Combinaison](#).

Here is the caller graph for this function:



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

- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/CombiWordle.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/CombiWordle.cpp

## 6.4 FonctionsUtiles Class Reference

Bibliothèque regroupant des fonctions utiles.

```
#include <FonctionsUtiles.hpp>
```

### Public Member Functions

- [FonctionsUtiles](#) ()  
*Constructeur neutre de la classe [FonctionsUtiles](#).*
- `vector< string > split` (const string &chaine, char delimiteur)  
*méthode permettant de découper une chaine de caractère en plusieurs grâce à un delimiteur*
- `bool checkMotFichier` (string fichier, string mot)  
*méthode qui permet de vérifier si un mot appartient au fichier*
- `bool isNumber` (const string &str)  
*méthode permettant de savoir si la chaine est un nombre entier supérieur à 0 ou non*
- `string intToString` (int a)  
*méthode permettant de transformer un entier en chaine de caractère*
- `string charToString` (char c)  
*méthode permettant de transformer un caractère en chaine de caractère*
- `void clear` ()  
*méthode permettant de nettoyer le terminal*
- `string vert` (string s)  
*méthode permettant de mettre le texte en vert*
- `string rouge` (string s)  
*méthode permettant de mettre le texte en rouge*
- `string jaune` (string s)  
*méthode permettant de mettre le texte en jaune*
- `string bleu` (string s)  
*méthode permettant de mettre le texte en bleu*
- `string violet` (string s)  
*méthode permettant de mettre le texte en violet*
- `string blanc` (string s)  
*méthode permettant de mettre le texte en blanc*
- `string cyan` (string s)  
*méthode permettant de mettre le texte en cyan*
- `string carrevert` ()  
*méthode permettant de renvoyer une carré vert*
- `string carrerouge` ()  
*méthode permettant de renvoyer une carré rouge*
- `string carrejaune` ()  
*méthode permettant de renvoyer une carré jaune*
- `string carrebleu` ()  
*méthode permettant de renvoyer une carré bleu*
- `string carreviolet` ()

- méthode permettant de renvoyer un carré violet*
- string `carreblanc` ()
- méthode permettant de renvoyer un carré blanc*
- string `carreorange` ()
- méthode permettant de renvoyer un carré orange*
- string `carrerose` ()
- méthode permettant de renvoyer un carré rose*
- string `carremarron` ()
- méthode permettant de renvoyer un carré marron*
- string `carrevide` ()
- méthode permettant de renvoyer un carré vide*
- string `toCarre` (string s)
- méthode permettant de transformer une chaîne en suite de lettres carrées*
- string `blink` (string s)
- méthode permettant de rendre une chaîne de caractère clignotante*
- string `violetclair` (string s)
- méthode permettant de mettre le texte en violetclair et en gras*
- string `vertclair` (string s)
- méthode permettant de mettre le texte en vertclair et en gras*
- bool `equals` (const string &s1, const string &s2)
- méthode permettant de comparer deux chaînes de caractère*

### 6.4.1 Detailed Description

Bibliothèque regroupant des fonctions utiles.

Author

Groupe A7

Version

1.0

Date

avril 2022

### 6.4.2 Member Function Documentation

#### 6.4.2.1 `blanc()`

```
string FonctionsUtiles::blanc (
    string s )
```

méthode permettant de mettre le texte en blanc



**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

**6.4.2.2 bleu()**

```
string FonctionsUtiles::bleu (  
    string s )
```

méthode permettant de mettre le texte en bleu

**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.3 blink()**

```
string FonctionsUtiles::blink (  
    string s )
```

méthode permettant de rendre une chaine de caractère clignotante

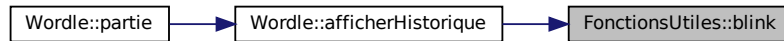
**Parameters**

<b>s</b>	: chaine à transformer
----------	------------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.4 carreblanc()**

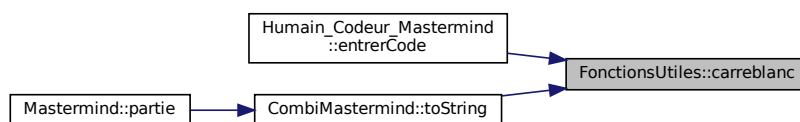
```
string FonctionsUtiles::carreblanc ( )
```

méthode permettant de renvoyer une carré blanc

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.5 carrebleu()**

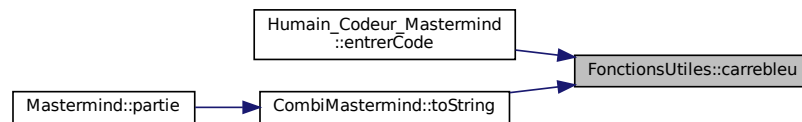
```
string FonctionsUtiles::carrebleu ( )
```

méthode permettant de renvoyer une carré bleu

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.6 carrejaune()**

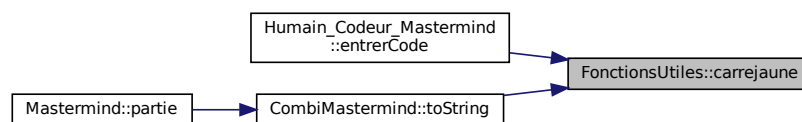
```
string FonctionsUtiles::carrejaune ( )
```

méthode permettant de renvoyer une carré jaune

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.7 carremarron()**

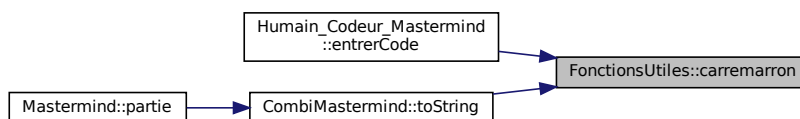
```
string FonctionsUtiles::carremarron ( )
```

méthode permettant de renvoyer une carré marron

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.8 carreorange()**

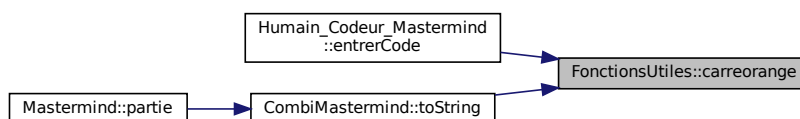
```
string FonctionsUtiles::carreorange ( )
```

méthode permettant de renvoyer une carré orange

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.9 carrerose()**

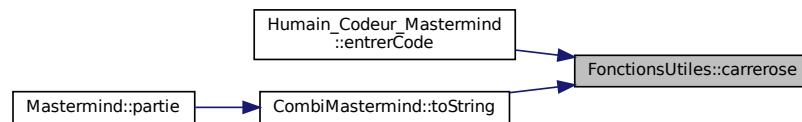
```
string FonctionsUtiles::carrerose ( )
```

méthode permettant de renvoyer une carré rose

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.10 carrerouge()**

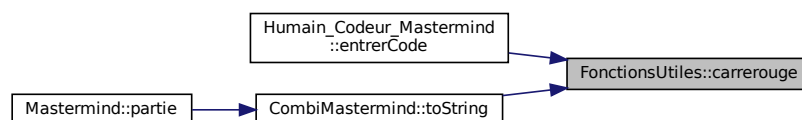
```
string FonctionsUtiles::carrerouge ( )
```

méthode permettant de renvoyer une carré rouge

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.11 carrevert()**

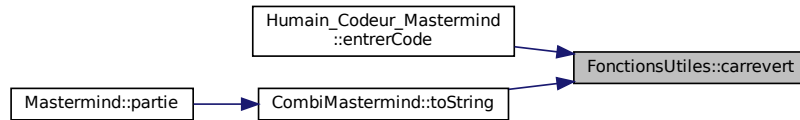
```
string FonctionsUtiles::carrevert ( )
```

méthode permettant de renvoyer une carré vert

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.12 carrevide()**

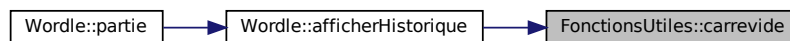
```
string FonctionsUtiles::carrevide ( )
```

méthode permettant de renvoyer une carré vide

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.13 carreviolet()**

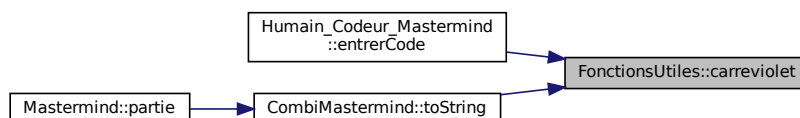
```
string FonctionsUtiles::carreviolet ( )
```

méthode permettant de renvoyer une carré violet

**Returns**

une chaine de caractère

Here is the caller graph for this function:



#### 6.4.2.14 charToString()

```
string FonctionsUtiles::charToString (
    char a )
```

méthode permettant de transformer un caractère en chaîne de caractère

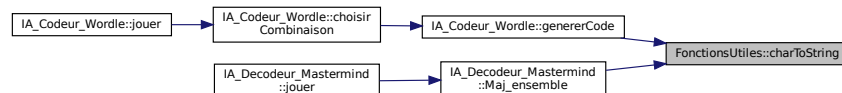
##### Parameters

<i>c</i>	: caractère à transformer
----------	---------------------------

##### Returns

une chaîne de caractère

Here is the caller graph for this function:



#### 6.4.2.15 checkMotFichier()

```
bool FonctionsUtiles::checkMotFichier (
    string fichier,
    string mot )
```

méthode qui permet de vérifier si un mot appartient au fichier

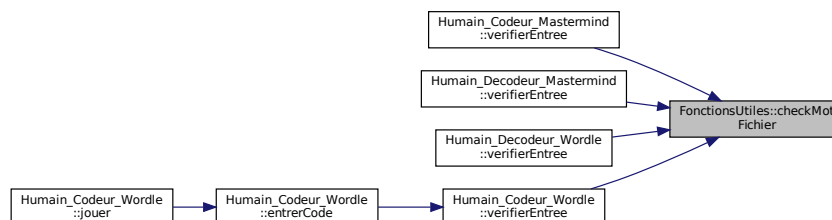
##### Parameters

<i>fichier</i>	: le chemin d'accès au fichier pour vérifier
<i>mot</i>	: le mot à vérifier

**Returns**

un booléen si le mot est dans le fichier

Here is the caller graph for this function:

**6.4.2.16 cyan()**

```
string FonctionsUtilites::cyan (
    string s )
```

méthode permettant de mettre le texte en cyan

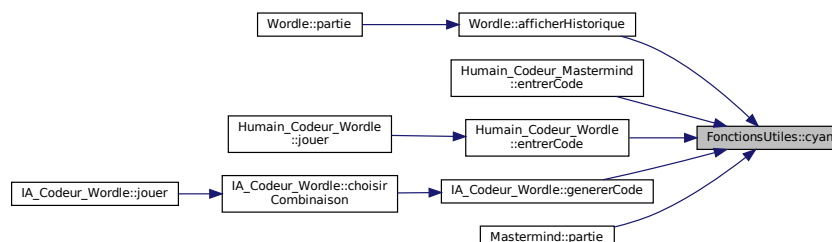
**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:





### 6.4.2.17 equals()

```
bool FonctionsUtiles::equals (
    const string & s1,
    const string & s2 )
```

méthode permettant de comparer deux chaînes de caractère

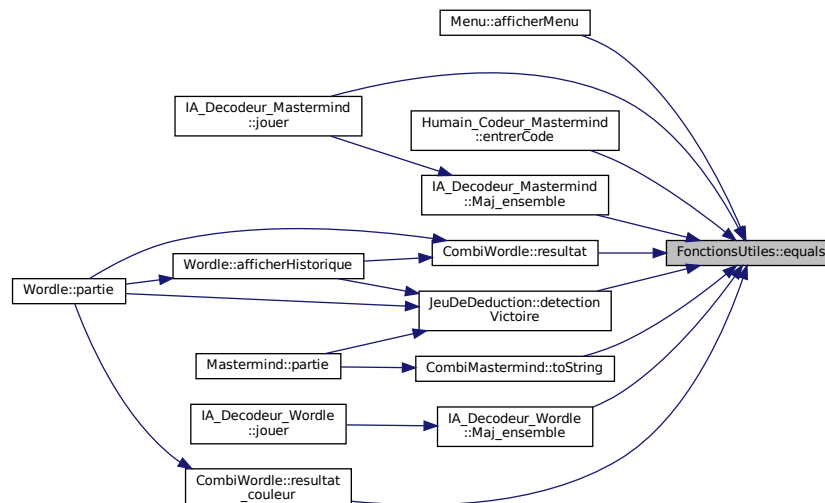
#### Parameters

<i>s1</i>	: chaîne à comparer
<i>s2</i>	: chaîne à comparer

#### Returns

un booléen

Here is the caller graph for this function:



### 6.4.2.18 intToString()

```
string FonctionsUtiles::intToString (
    int a )
```

méthode permettant de transformer un entier en chaîne de caractère

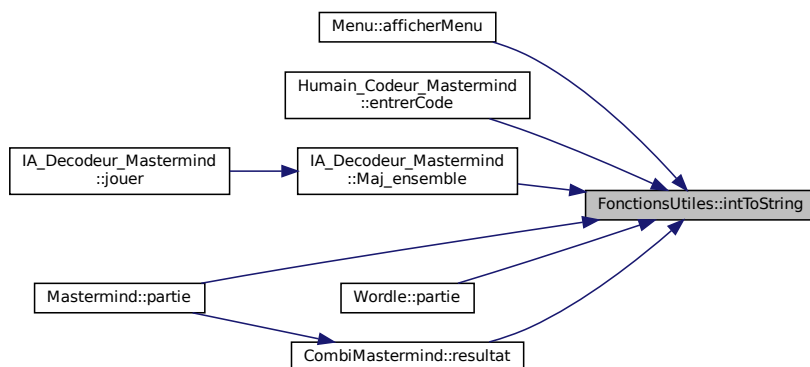
#### Parameters

<i>a</i>	: entier à transformer
----------	------------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.19 isNumber()**

```
bool FonctionsUtiles::isNumber (
    const string & str )
```

méthode permettant de savoir si la chaine est un nombre entier supérieur à 0 ou non

**Parameters**

<i>str</i>	: chaine à analyser
------------	---------------------

**Returns**

un booléen

Here is the caller graph for this function:



#### 6.4.2.20 jaune()

```
string FonctionsUtiles::jaune (  
    string s )
```

méthode permettant de mettre le texte en jaune

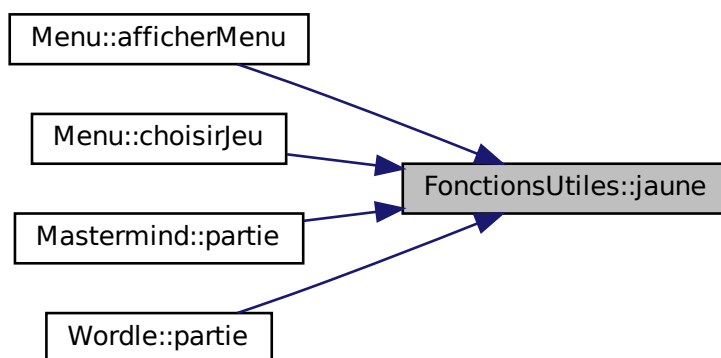
##### Parameters

<b>s</b>	: texte à transformer
----------	-----------------------

##### Returns

une chaine de caractère

Here is the caller graph for this function:



#### 6.4.2.21 rouge()

```
string FonctionsUtiles::rouge (  
    string s )
```

méthode permettant de mettre le texte en rouge

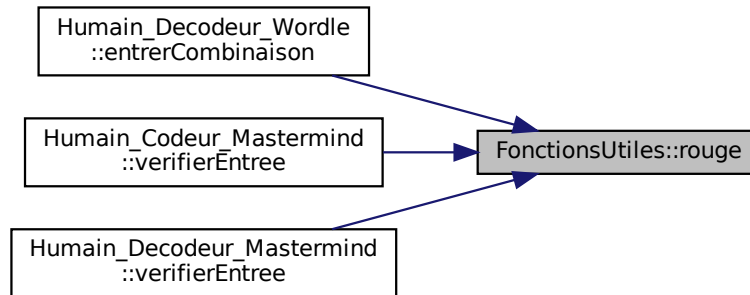
##### Parameters

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.22 split()**

```
vector< string > FonctionsUtile::split (
    const string & chaine,
    char delimiteur )
```

méthode permettant de découper une chaine de caractère en plusieurs grâce à un delimiteur

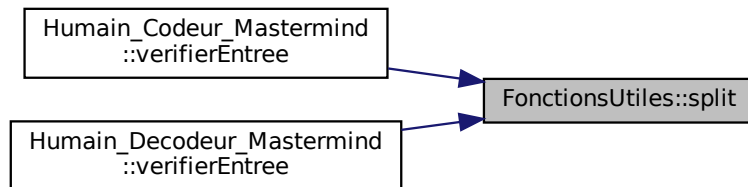
**Parameters**

<i>chaine</i>	: chaine à découper
<i>delimiteur</i>	: le séparateur de mot, ici l'espace

**Returns**

un tableau de string

Here is the caller graph for this function:

**6.4.2.23 toCarre()**

```
string FonctionsUtiles::toCarre (
    string s )
```

méthode permettant de transformer une chaine en suite de lettres carrées

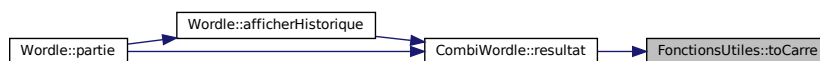
**Parameters**

<b>s</b>	: chaine à transformer
----------	------------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.24 vert()**

```
string FonctionsUtiles::vert (
    string s )
```

méthode permettant de mettre le texte en vert

**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

**6.4.2.25 vertclair()**

```
string FonctionsUtiles::vertclair (  
    string s )
```

méthode permettant de mettre le texte en vertclair et en gras

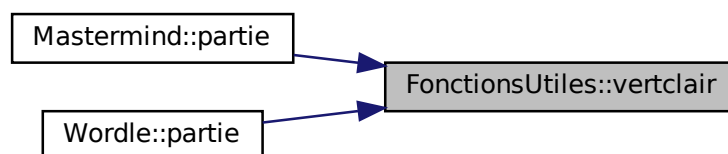
**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:

**6.4.2.26 violet()**

```
string FonctionsUtiles::violet (  
    string s )
```

méthode permettant de mettre le texte en violet

**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

**6.4.2.27 violetclair()**

```
string FonctionsUtiles::violetclair (  
    string s )
```

méthode permettant de mettre le texte en violetclair et en gras

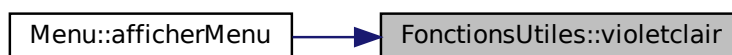
**Parameters**

<b>s</b>	: texte à transformer
----------	-----------------------

**Returns**

une chaine de caractère

Here is the caller graph for this function:



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

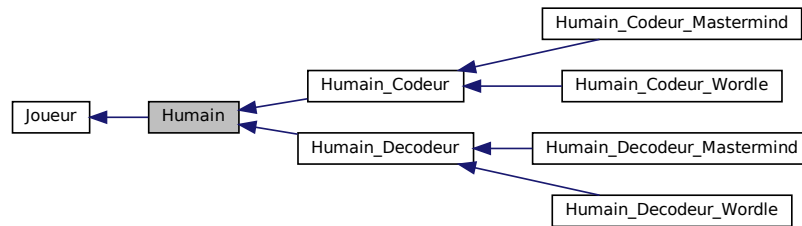
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/FonctionsUtiles.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/FonctionsUtiles.cpp

## 6.5 Humain Class Reference

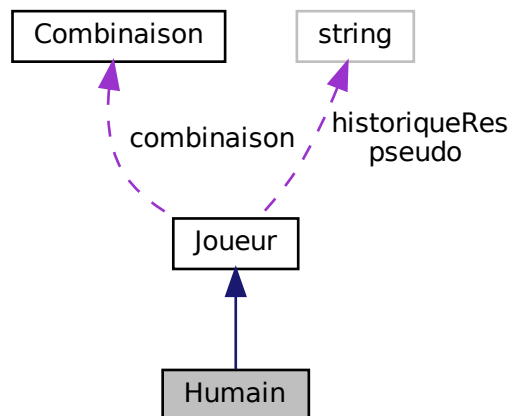
Si le joueur est un humain.

```
#include <Humain.hpp>
```

Inheritance diagram for Humain:



Collaboration diagram for Humain:



## Public Member Functions

- [Humain](#) ()  
*Constructeur de la classe [Humain](#).*
- [Humain](#) (string pseudo1)  
*Constructeur de la classe [Humain](#) avec paramètre.*
- virtual bool [verifierEntree](#) (string entree)=0  
*une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu*

## Additional Inherited Members

### 6.5.1 Detailed Description

Si le joueur est un humain.



**Author**

Groupe A7

**Version**

1.0

**Date**

avril 2022

## 6.5.2 Constructor & Destructor Documentation

### 6.5.2.1 Humain()

```
Humain::Humain (
    string pseudo1 )
```

Constructeur de la classe [Humain](#) avec paramètre.

**Parameters**

<i>pseudo1</i>	du joueur
----------------	-----------

## 6.5.3 Member Function Documentation

### 6.5.3.1 verifierEntree()

```
Humain::verifierEntree (
    string entree ) [pure virtual]
```

une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu

**Parameters**

<i>string</i>	<i>entree</i>
---------------	---------------

**Returns**

retourne un booléen (vraie seulement si la syntaxe est bonne)

Implemented in [Humain\\_Codeur\\_Wordle](#), [Humain\\_Decodeur\\_Wordle](#), [Humain\\_Decodeur\\_Mastermind](#), and [Humain\\_Codeur\\_Mastermind](#).

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

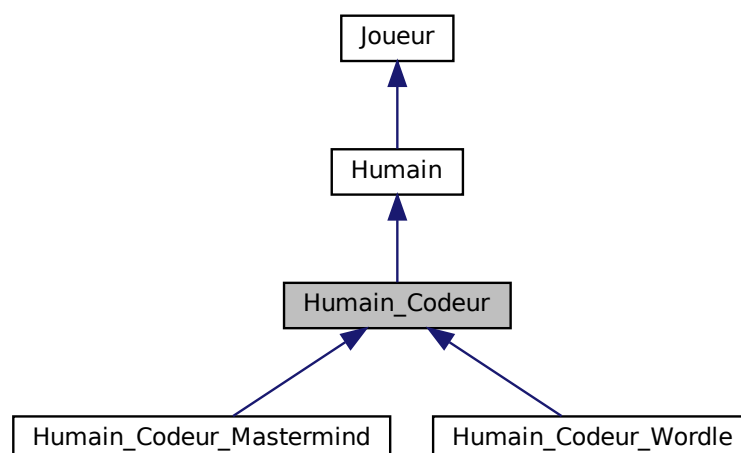
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain.cpp

## 6.6 Humain\_Codeur Class Reference

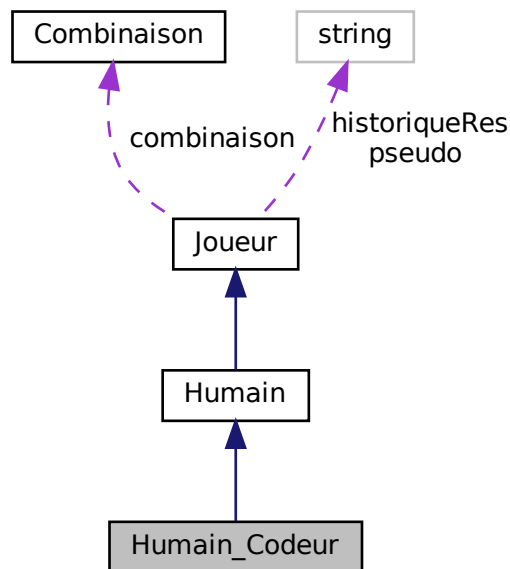
Cette classe abstraite permet de décrire les joueurs humains en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#include <Humain_Codeur.hpp>
```

Inheritance diagram for Humain\_Codeur:



Collaboration diagram for Humain\_Codeur:



## Public Member Functions

- [Humain\\_Codeur](#) ()  
*Constructeur de la classe [Humain\\_Codeur](#).*
- virtual [Combinaison entrerCode](#) ()=0  
*methode abstraite utilisée pour demander un code au joueur codeur. Cette méthode devra être redefini pour les différents modes de jeu*
- int [entrerMdP](#) ()  
*Méthode qui permet à l'utilisateur d'entrer son mot de passe.*
- int [getMdP](#) ()  
*accesseur de l'attribut MdP*

## Additional Inherited Members

### 6.6.1 Detailed Description

Cette classe abstraite permet de décrire les joueurs humains en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

Author

Groupe A7

Version

1.0

Date

avril 2022

## 6.6.2 Member Function Documentation

### 6.6.2.1 entrerMdP()

```
Humain_Codeur::entrerMdP ( )
```

Méthode qui permet à l'utilisateur d'entrer son mot de passe.

#### Returns

retourne l'entier entré par l'utilisateur

### 6.6.2.2 getMdP()

```
Humain_Codeur::getMdP ( )
```

accesseur de l'attribut MdP

#### Returns

retourne l'attribut MdP de la classe [Humain\\_Codeur](#)

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

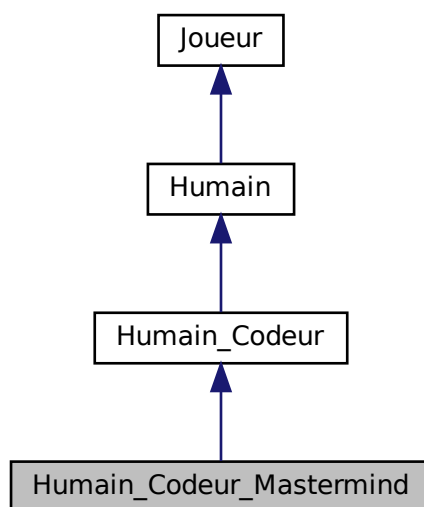
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Codeur.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Codeur.cpp

## 6.7 Humain\_Codeur\_Mastermind Class Reference

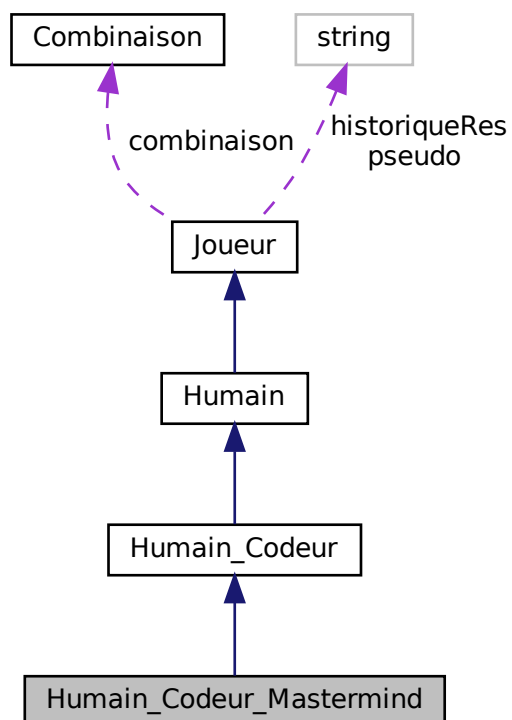
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Mastermind](#).

```
#include <Humain_Codeur_Mastermind.hpp>
```

Inheritance diagram for Humain\_Codeur\_Mastermind:



Collaboration diagram for Humain\_Codeur\_Mastermind:



## Public Member Functions

- [Humain\\_Codeur\\_Mastermind](#) ()  
*Constructeur de la classe [Humain\\_Codeur\\_Mastermind](#).*
- virtual [Combinaison entrerCode](#) ()  
*méthode permettant à l'utilisateur de saisir le code que le joueur décodeur devra ensuite deviner*
- virtual bool [verifierEntree](#) (string entree)  
*une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie mastermind avec humain codeur.*

## Additional Inherited Members

### 6.7.1 Detailed Description

Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Mastermind](#).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.7.2 Member Function Documentation

#### 6.7.2.1 entrerCode()

```
Combinaison Humain_Codeur_Mastermind::entrerCode ( ) [virtual]
```

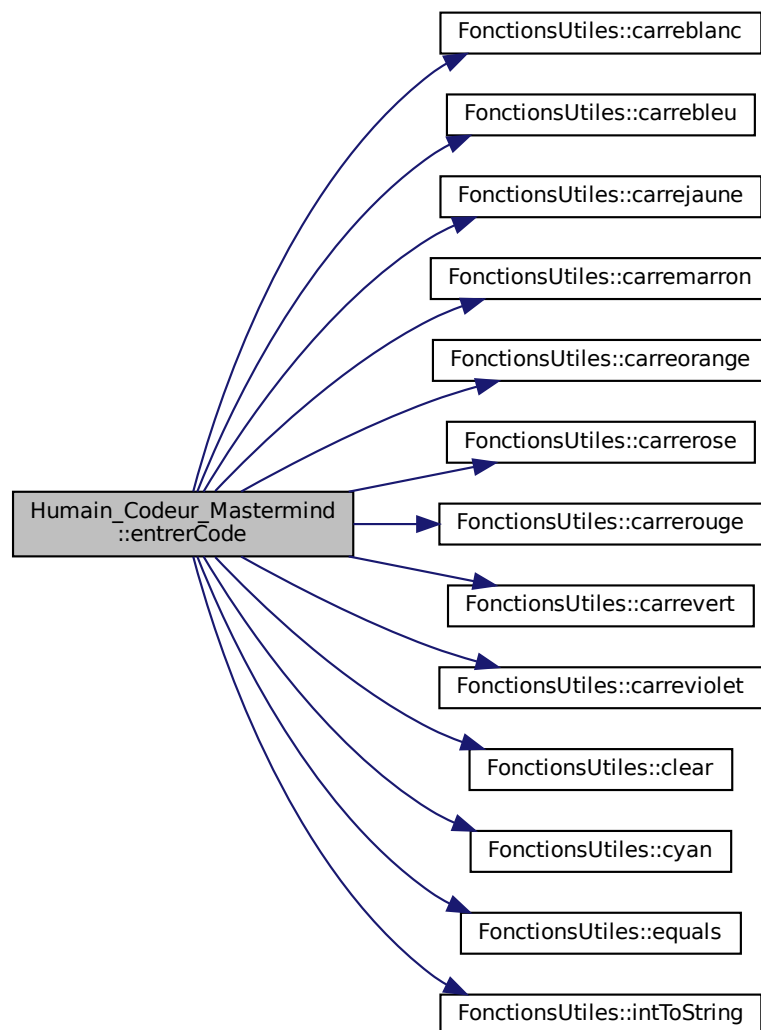
méthode permettant à l'utilisateur de saisir le code que le joueur décodeur devra ensuite deviner

**Returns**

retourne le code saisi par le joueur codeur

Implements [Humain\\_Codeur](#).

Here is the call graph for this function:

**6.7.2.2 verifierEntree()**

```
bool Humain_Codeur_Mastermind::verifierEntree (  
    string entree ) [virtual]
```

une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu

**Parameters**

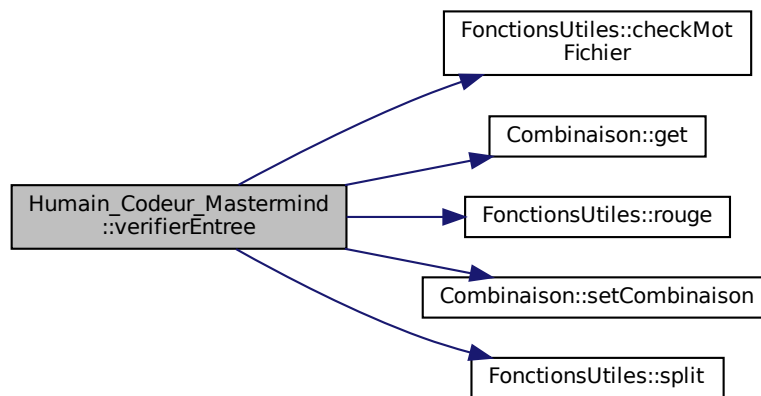
<i>string</i>	entree
---------------	--------

**Returns**

retourne un booléen (vraie seulement si la syntaxe est bonne)

Implements [Humain](#).

Here is the call graph for this function:



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

- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.cpp`

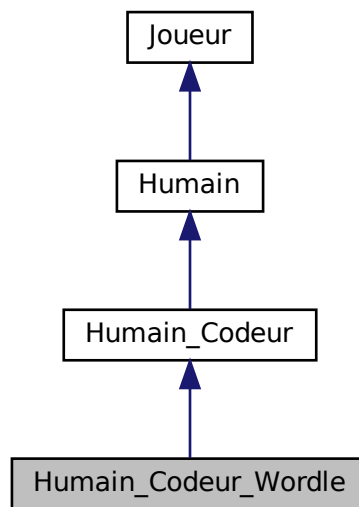
## 6.8 Humain\_Codeur\_Wordle Class Reference

Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Wordle](#).

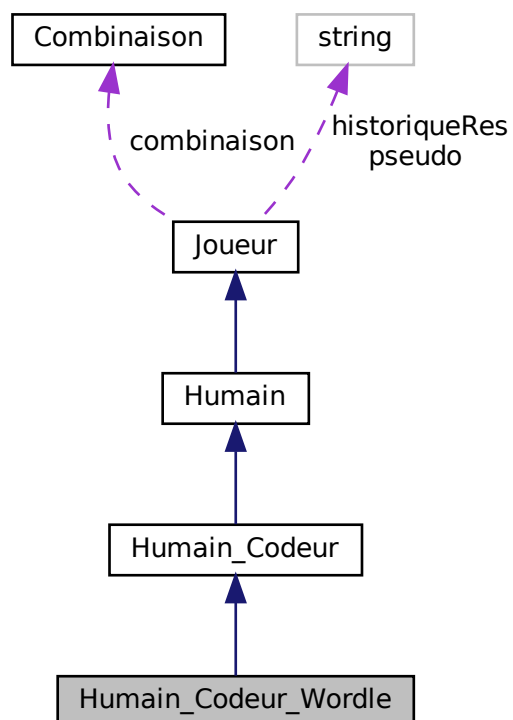
```
#include <Humain_Codeur_Wordle.hpp>
```



Inheritance diagram for Humain\_Codeur\_Wordle:



Collaboration diagram for Humain\_Codeur\_Wordle:



## Public Member Functions

- [Humain\\_Codeur\\_Wordle](#) ()  
*Constructeur de la classe [Humain\\_Codeur\\_Wordle](#).*
- virtual [Combinaison](#) [entrerCode](#) ()  
*méthode permettant à l'utilisateur de saisir le code que le joueur décodeur devra ensuite deviner*
- virtual bool [verifierEntree](#) (string entree)  
*une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie wordle avec l'humain codeur.*

## Additional Inherited Members

### 6.8.1 Detailed Description

Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Wordle](#).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.8.2 Member Function Documentation

#### 6.8.2.1 [entrerCode\(\)](#)

```
Combinaison Humain_Codeur_Wordle::entrerCode ( ) [virtual]
```

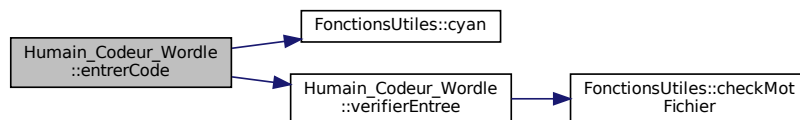
méthode permettant à l'utilisateur de saisir le code que le joueur décodeur devra ensuite deviner

**Returns**

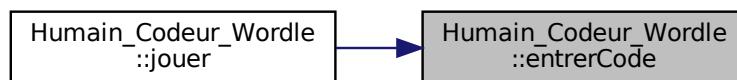
retourne le code saisi par le joueur codeur

Implements [Humain\\_Codeur](#).

Here is the call graph for this function:



Here is the caller graph for this function:

**6.8.2.2 verifierEntree()**

```
bool Humain_Codeur_Wordle::verifierEntree (
    string entree ) [virtual]
```

une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu

**Parameters**

<i>string</i>	entree
---------------	--------

**Returns**

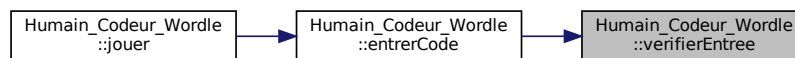
retourne un booléen (vraie seulement si la syntaxe est bonne)

Implements [Humain](#).

Here is the call graph for this function:



Here is the caller graph for this function:



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

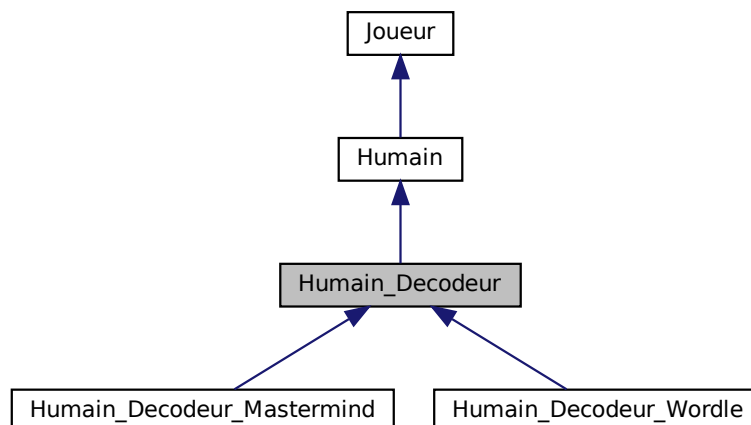
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Codeur\_Wordle.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Codeur\_Wordle.cpp

## 6.9 Humain\_Decodeur Class Reference

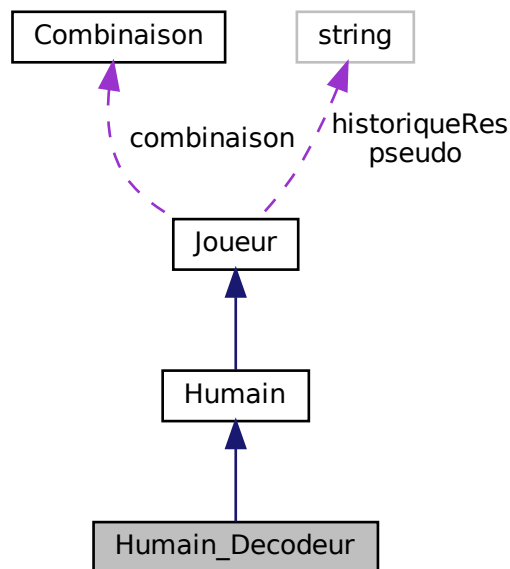
Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#include <Humain_Decodeur.hpp>
```

Inheritance diagram for Humain\_Decodeur:



Collaboration diagram for Humain\_Decodeur:



## Public Member Functions

- [Humain\\_Decodeur](#) ()  
*Constructeur de la classe [Humain\\_Decodeur](#).*
- virtual [Combinaison entrerCombinaison](#) ()=0  
*methode abstraite à redefinir pour les différents modes de jeu*

## Additional Inherited Members

### 6.9.1 Detailed Description

Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

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

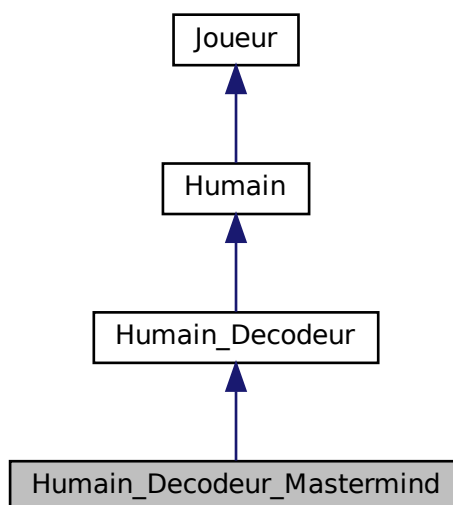
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Decodeur.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Humain\_Decodeur.cpp

## 6.10 Humain\_Decodeur\_Mastermind Class Reference

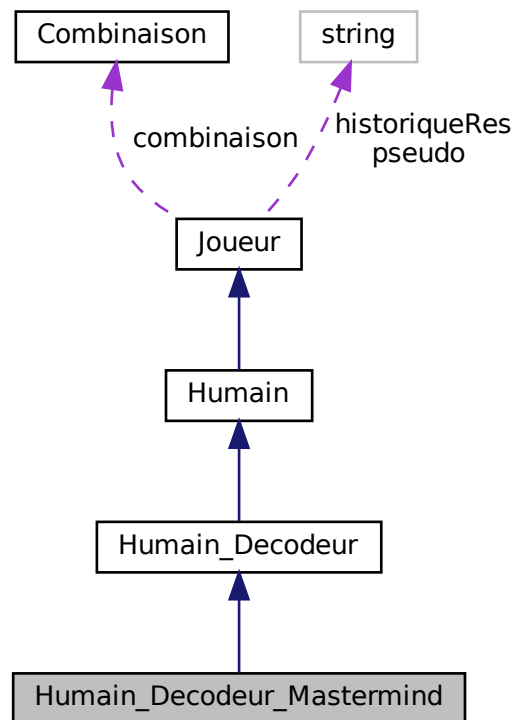
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Mastermind](#).

```
#include <Humain_Decodeur_Mastermind.hpp>
```

Inheritance diagram for Humain\_Decodeur\_Mastermind:



Collaboration diagram for Humain\_Decodeur\_Mastermind:



## Public Member Functions

- [Humain\\_Decodeur\\_Mastermind](#) ()  
*Constructeur de la classe [Humain\\_Decodeur\\_Mastermind](#).*
- virtual [Combinaison entrerCombinaison](#) ()  
*méthode permettant à l'utilisateur de saisir une combinaison qu'il propose pour deviner le code donnée par le joueur codeur*
- bool [verifierEntree](#) (string entree)  
*une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie mastermind avec humain decodeur.*

## Additional Inherited Members

### 6.10.1 Detailed Description

Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Mastermind](#).

**Author**

Groupe A7

**Version**

1.0

**Date**

avril 2022

## 6.10.2 Member Function Documentation

### 6.10.2.1 entrerCombinaison()

```
Combinaison Humain_Decodeur_Mastermind::entrerCombinaison ( ) [virtual]
```

méthode permettant à l'utilisateur de saisir une combinaison qu'il propose pour deviner le code donnée par le joueur codeur

**Returns**

retourne la combinaison saisi par le joueur decodeur

Implements [Humain\\_Decodeur](#).

### 6.10.2.2 verifierEntree()

```
bool Humain_Decodeur_Mastermind::verifierEntree (
    string entree ) [virtual]
```

une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu

**Parameters**

<i>string</i>	entree
---------------	--------

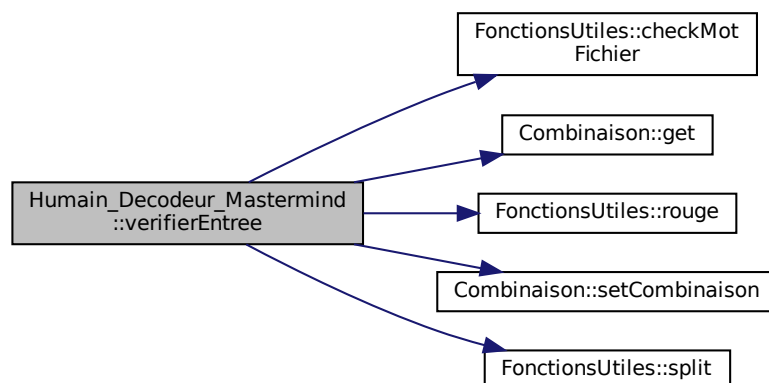
**Returns**

retourne un booléen (vraie seulement si la syntaxe est bonne)

Implements [Humain](#).



Here is the call graph for this function:



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

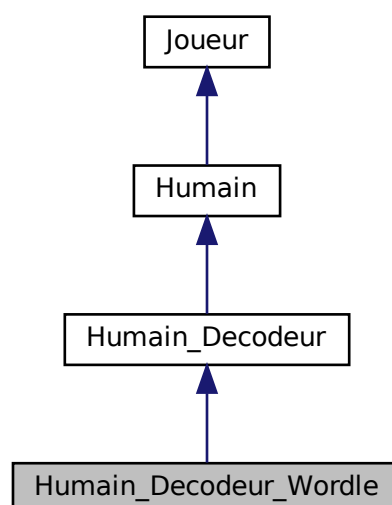
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.cpp`

## 6.11 Humain\_Decodeur\_Wordle Class Reference

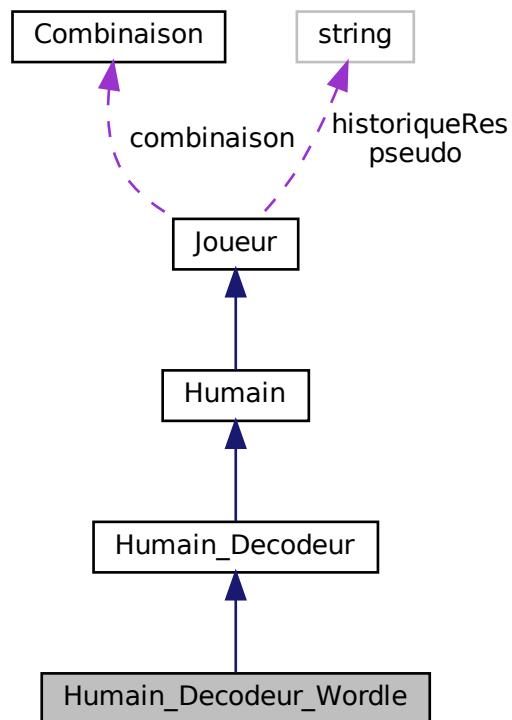
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Wordle](#).

```
#include <Humain_Decodeur_Wordle.hpp>
```

Inheritance diagram for `Humain_Decodeur_Wordle`:



Collaboration diagram for Humain\_Decodeur\_Wordle:



## Public Member Functions

- [Humain\\_Decodeur\\_Wordle](#) ()  
*Constructeur de la classe Humain\_Codeur\_Decodeur.*
- virtual [Combinaison entrerCombinaison](#) ()  
*methode abstraite à redefinir pour les différents modes de jeu*
- bool [verifierEntree](#) (string entree)  
*Méthode qui Permet de vérifier si l'entrée du joueur codeur peut être interpréter en un mot du dictionnaire du jeu [Wordle](#).*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie wordle de l'humain décodeur.*

## Additional Inherited Members

### 6.11.1 Detailed Description

Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Wordle](#).

**Author**

Groupe A7

**Version**

1.0

**Date**

avril 2022

## 6.11.2 Member Function Documentation

### 6.11.2.1 verifierEntree()

```
Humain_Decodeur_Wordle::verifierEntree (  
    string entree ) [virtual]
```

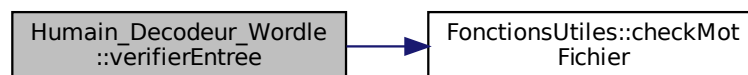
Méthode qui Permet de vérifier si l'entrée du joueur codeur peut être interpréter en un mot du dictionnaire du jeu [Wordle](#).

**Returns**

retourne un booleen, true si la chaîne peut être considérée comme un mot du dictionnaire et false sinon

Implements [Humain](#).

Here is the call graph for this function:



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

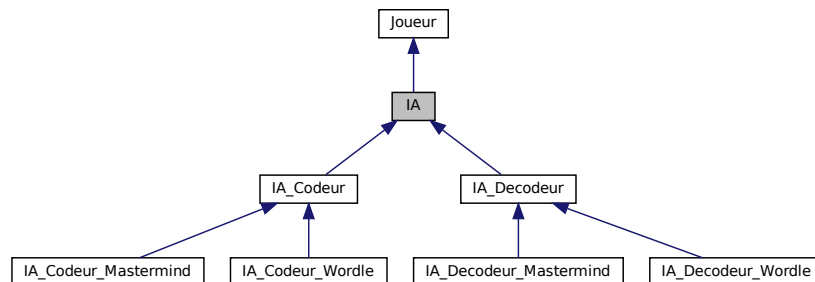
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.cpp`

## 6.12 IA Class Reference

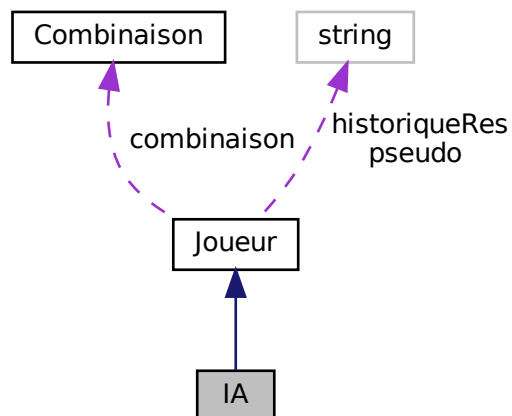
Si le joueur est un [IA](#).

```
#include <IA.hpp>
```

Inheritance diagram for IA:



Collaboration diagram for IA:



### Public Member Functions

- [IA](#) ()  
*Constructeur de la classe [IA](#).*
- [IA](#) (string pseudo1)  
*Constructeur de la classe [IA](#) avec paramètre.*
- virtual [Combinaison](#) choisirCombinaison ()=0  
*une méthode abstraite qui permettra à l'[IA](#) de trouver la combinaison correct selon le mode de jeu*

## Protected Attributes

- int **difficulte**

### 6.12.1 Detailed Description

Si le joueur est un [IA](#).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 IA()

```
IA::IA (
    string pseudo1 )
```

Constructeur de la classe [IA](#) avec paramètre.

#### Parameters

<i>pseudo1</i>	du joueur
----------------	-----------

### 6.12.3 Member Function Documentation

#### 6.12.3.1 choisirCombinaison()

```
Combinaison IA::choisirCombinaison ( ) [pure virtual]
```

une méthode abstraite qui permettra à l'[IA](#) de trouver la combinaison correct selon le mode de jeu

**Returns**

une combinaison

Implemented in [IA\\_Decodeur\\_Wordle](#), [IA\\_Decodeur\\_Mastermind](#), [IA\\_Codeur\\_Wordle](#), and [IA\\_Codeur\\_Mastermind](#).

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

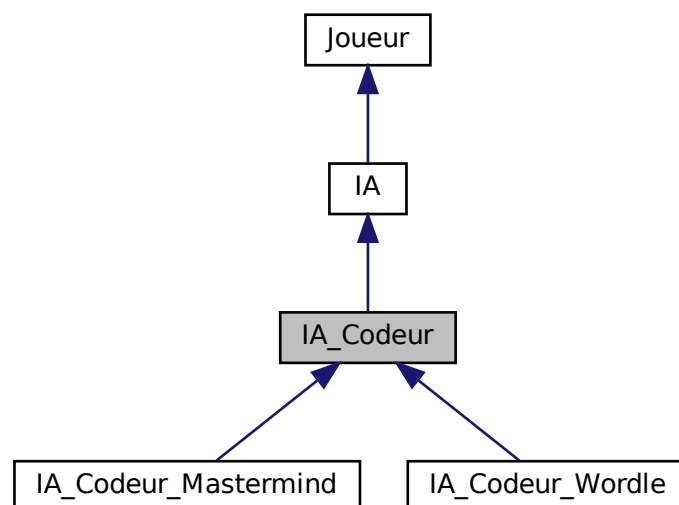
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA.cpp

## 6.13 IA\_Codeur Class Reference

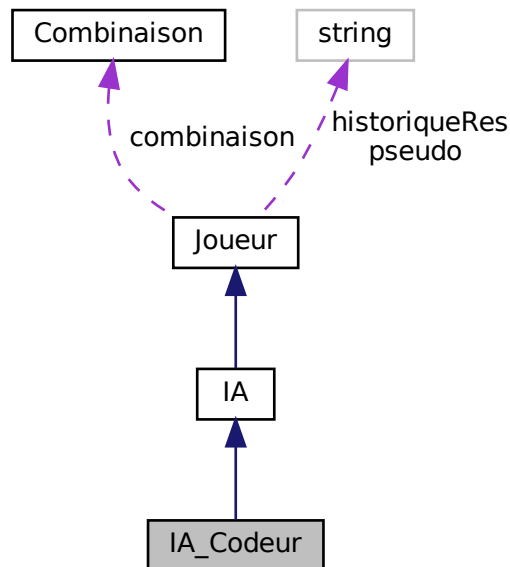
Cette classe abstraite permet de décrire l'intelligence artificielle en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#include <IA_Codeur.hpp>
```

Inheritance diagram for IA\_Codeur:



Collaboration diagram for IA\_Codeur:



## Public Member Functions

- [IA\\_Codeur](#) ()  
*Constructeur de la classe [IA\\_Codeur](#).*
- virtual [Combinaison genererCode](#) ()=0  
*methode abstraite permettant de générer le code que le décodeur devra deviner*
- int [genererMdP](#) ()  
*Méthode qui permet à la machine de générer un mot de passe.*
- int [getMdP](#) ()  
*accesseur de l'attribut MdP*

## Additional Inherited Members

### 6.13.1 Detailed Description

Cette classe abstraite permet de décrire l'intelligence artificielle en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

## 6.13.2 Member Function Documentation

### 6.13.2.1 genererCode()

```
Combinaison IA_Codeur::genererCode ( ) [pure virtual]
```

methode abstraite permettant de générer le code que le décodeur devra deviner

#### Returns

retourne une combinaison

Implemented in [IA\\_Codeur\\_Wordle](#), and [IA\\_Codeur\\_Mastermind](#).

### 6.13.2.2 genererMdP()

```
int IA_Codeur::genererMdP ( )
```

Méthode qui permet à la machine de générer un mot de passe.

#### Returns

retourne l'entier généré

### 6.13.2.3 getMdP()

```
int IA_Codeur::getMdP ( )
```

accesseur de l'attribut MdP

#### Returns

retourne l'attribut MdP de la classe [IA\\_Codeur](#)

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

- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Codeur.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Codeur.cpp

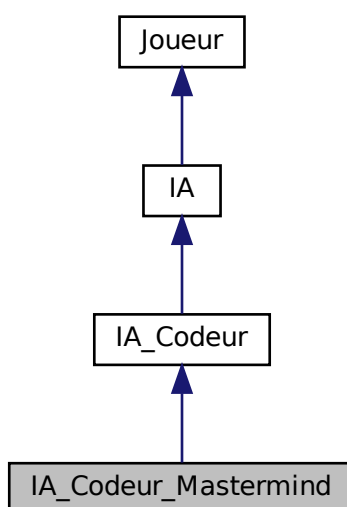


## 6.14 IA\_Codeur\_Mastermind Class Reference

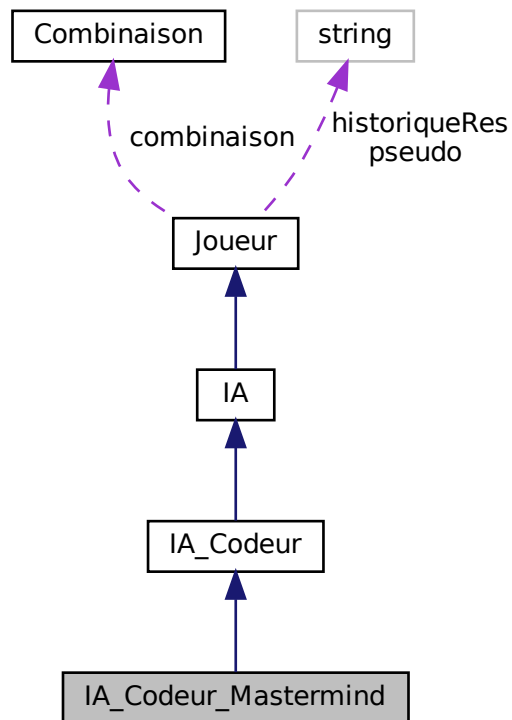
Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode codeur dans le mode de jeu [Mastermind](#).

```
#include <IA_Codeur_Mastermind.hpp>
```

Inheritance diagram for IA\_Codeur\_Mastermind:



Collaboration diagram for `IA_Codeur_Mastermind`:



## Public Member Functions

- [IA\\_Codeur\\_Mastermind](#) ()  
*Constructeur de la classe [IA\\_Codeur\\_Mastermind](#).*
- virtual [Combinaison genererCode](#) ()  
*méthode permettant à la machine de générer un code que l'intelligence artificielle devra ensuite trouver*
- virtual [Combinaison choisirCombinaison](#) ()  
*Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie mastermind avec [IA](#) codeur.*

## Additional Inherited Members

### 6.14.1 Detailed Description

Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode codeur dans le mode de jeu [Mastermind](#).

Author

Groupe A7

## Version

1.0

## Date

avril 2022

## 6.14.2 Member Function Documentation

### 6.14.2.1 choisirCombinaison()

```
IA_Codeur_Mastermind::choisirCombinaison ( ) [virtual]
```

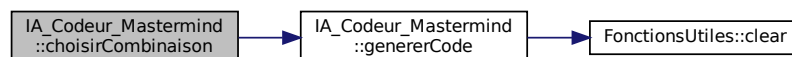
Cette méthode permet de a l'IA de choisir une combinaison parmi l'ensemble.

## Returns

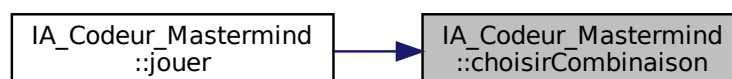
retourne la combinaison choisie

Implements [IA](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.14.2.2 genererCode()

```
IA_Codeur_Mastermind::genererCode ( ) [virtual]
```

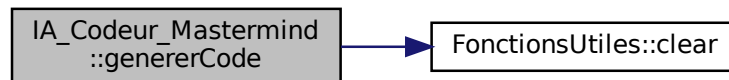
méthode permettant à la machine de générer un code que l'intelligence artificielle devra ensuite trouver

#### Returns

retourne le code créé par la machine

Implements [IA\\_Codeur](#).

Here is the call graph for this function:



Here is the caller graph for this function:



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

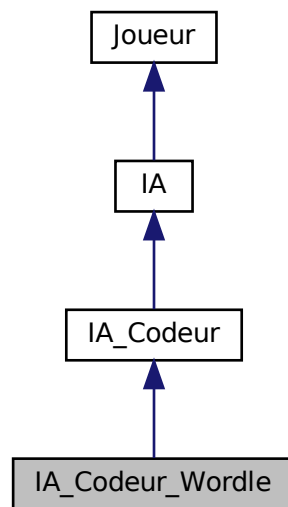
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.cpp`

## 6.15 IA\_Codeur\_Wordle Class Reference

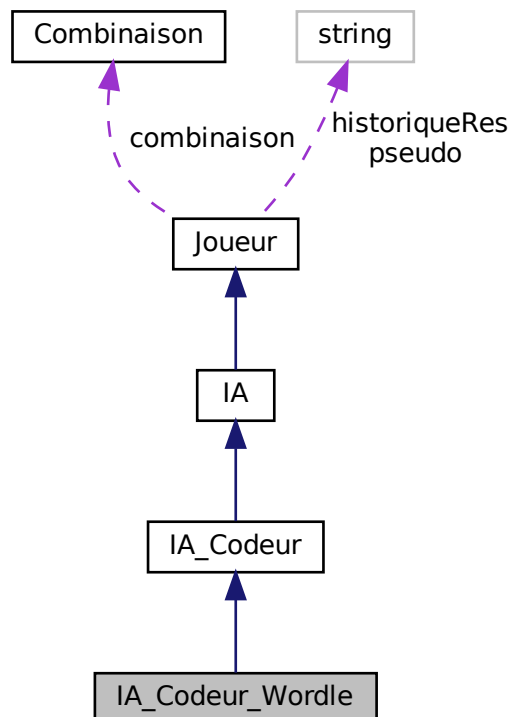
Cette classe est utilisée pour représenter un joueur virtuel ([IA](#)) en mode codeur dans le mode de jeu [Wordle](#).

```
#include <IA_Codeur_Wordle.hpp>
```

Inheritance diagram for IA\_Codeur\_Wordle:



Collaboration diagram for IA\_Codeur\_Wordle:



## Public Member Functions

- [IA\\_Codeur\\_Wordle](#) ()  
*Constructeur de la classe [IA\\_Codeur\\_Wordle](#).*
- virtual [Combinaison genererCode](#) ()  
*méthode permettant à la machine de générer un code que le joueur artificiel décodeur devra ensuite trouver*
- virtual [Combinaison choisirCombinaison](#) ()  
*Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie wordle avec [IA](#) codeur.*

## Additional Inherited Members

### 6.15.1 Detailed Description

Cette classe est utilisée pour représenter un joueur virtuel ([IA](#)) en mode codeur dans le mode de jeu [Wordle](#).

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.15.2 Member Function Documentation

#### 6.15.2.1 choisirCombinaison()

```
IA_Codeur_Wordle::choisirCombinaison ( ) [virtual]
```

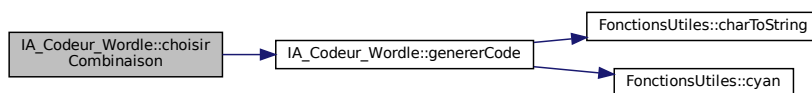
Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.

**Returns**

retourne la combinaison choisie

Implements [IA](#).

Here is the call graph for this function:



Here is the caller graph for this function:

**6.15.2.2 genererCode()**

```
IA_Codeur_Wordle::genererCode ( ) [virtual]
```

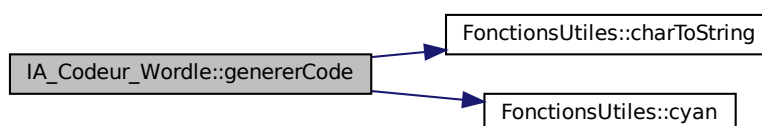
méthode permettant à la machine de générer un code que le joueur artificiel décodeur devra ensuite trouver

**Returns**

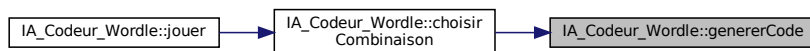
retourne le code créé par la machine

Implements [IA\\_Codeur](#).

Here is the call graph for this function:



Here is the caller graph for this function:



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

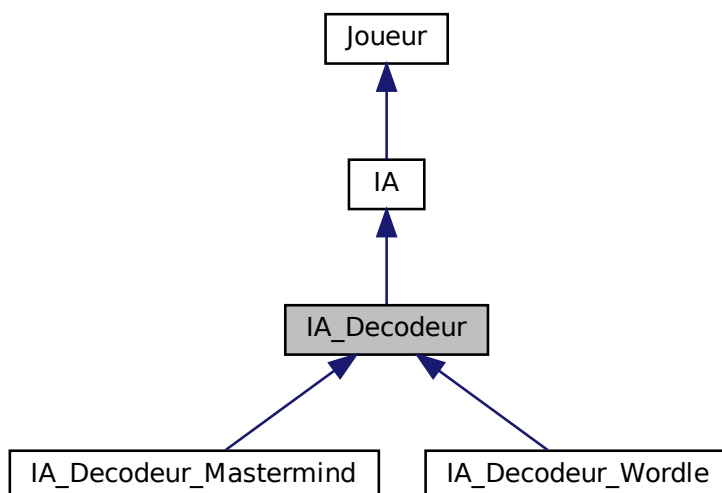
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Codeur\_Wordle.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Codeur\_Wordle.cpp

## 6.16 IA\_Decodeur Class Reference

Cette classe abstraite permet de décrire l'intelligence artificielle en mode Decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

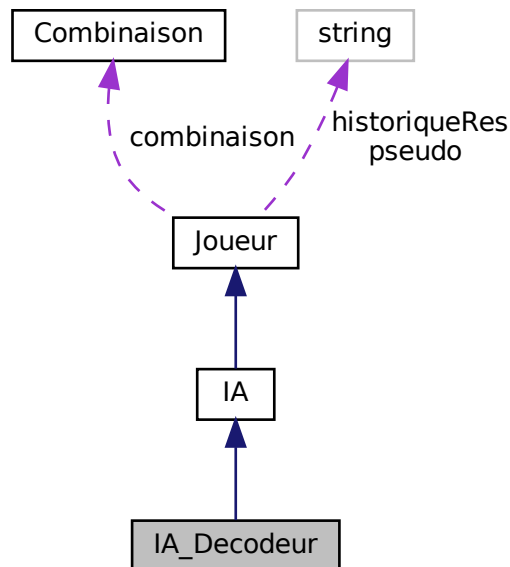
```
#include <IA_Decodeur.hpp>
```

Inheritance diagram for IA\_Decodeur:





Collaboration diagram for IA\_Decodeur:



## Public Member Functions

- [IA\\_Decodeur](#) ()  
*Constructeur de la classe [IA\\_Decodeur](#).*
- virtual vector< [Combinaison](#) > [Combi\\_possible](#) ()=0  
*Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons possibles dans le cas du [Mastermind](#) ou prendre l'ensemble des mots présents sur le dictionnaire pour le [Wordle](#).*
- virtual void [Maj\\_ensemble](#) ([Combinaison](#) combi, string resultat)=0  
*Si la proposition est incorrecte, cette méthode permet retirer de S tout code qui ne donnerait pas la même réponse si la supposition actuelle était le code.*

## Additional Inherited Members

### 6.16.1 Detailed Description

Cette classe abstraite permet de décrire l'intelligence artificielle en mode Decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

Author

Groupe A7

Version

1.0

Date

avril 2022

## 6.16.2 Member Function Documentation

### 6.16.2.1 Combi\_possible()

```
vector< Combinaison > IA_Decodeur::Combi_possible ( ) [pure virtual]
```

Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons possibles dans le cas du [Mastermind](#) ou prendre l'ensemble des mots présents sur le dictionnaire pour le [Wordle](#).

#### Returns

retourne une liste de combinaison

Implemented in [IA\\_Decodeur\\_Wordle](#), and [IA\\_Decodeur\\_Mastermind](#).

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

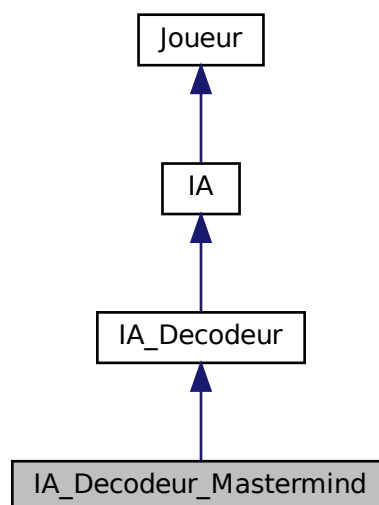
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Decodeur.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Decodeur.cpp

## 6.17 IA\_Decodeur\_Mastermind Class Reference

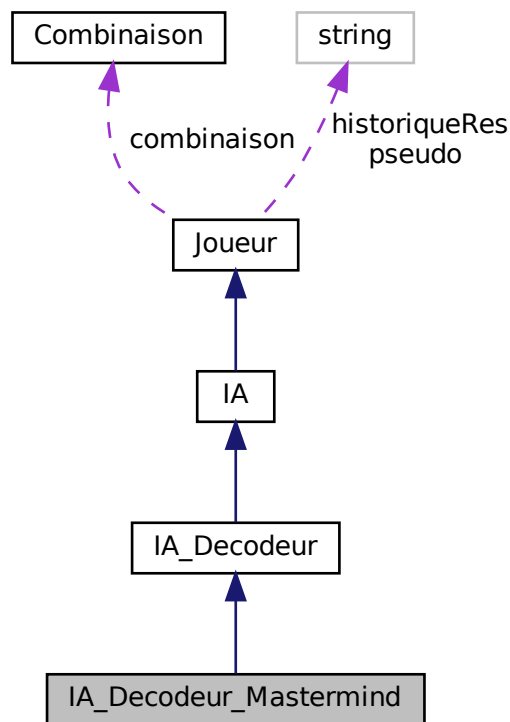
Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode Decodeur dans le mode de jeu [Mastermind](#).

```
#include <IA_Decodeur_Mastermind.hpp>
```

Inheritance diagram for [IA\\_Decodeur\\_Mastermind](#):



Collaboration diagram for IA\_Decodeur\_Mastermind:



## Public Member Functions

- [IA\\_Decodeur\\_Mastermind](#) ()  
*Constructeur de la classe [IA\\_Decodeur\\_Mastermind](#).*
- virtual vector< [Combinaison](#) > [Combi\\_possible](#) ()  
*Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons possibles dans le cas du [Mastermind](#) ou prendre l'ensemble des mots présents sur le dictionnaire pour le [Wordle](#).*
- virtual void [Maj\\_ensemble](#) ([Combinaison](#) combi, string resultat)  
*Si la proposition est incorrecte, cette méthode permet retirer de S tout code qui ne donnerait pas la même réponse si la supposition actuelle était le code.*
- virtual [Combinaison](#) [choisirCombinaison](#) ()  
*Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie mastermind avec [IA](#) décodeur.*

## Additional Inherited Members

### 6.17.1 Detailed Description

Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode Decodeur dans le mode de jeu [Mastermind](#).

**Author**

Groupe A7

**Version**

1.0

**Date**

avril 2022

## 6.17.2 Member Function Documentation

### 6.17.2.1 choisirCombinaison()

```
IA_Decodeur_Mastermind::choisirCombinaison ( ) [virtual]
```

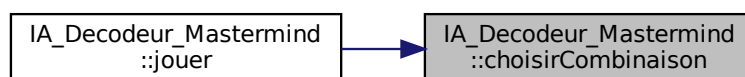
Cette méthode permet de a l'IA de choisir une combinaison parmi l'ensemble.

**Returns**

retourne la combinaison choisie

Implements [IA](#).

Here is the caller graph for this function:



### 6.17.2.2 Combi\_possible()

```
vector< Combinaison > IA_Decodeur_Mastermind::Combi_possible ( ) [virtual]
```

Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons possibles dans le cas du [Mastermind](#) ou prendre l'ensemble des mots présents sur le dictionnaire pour le [Wordle](#).

#### Returns

retourne une liste de combinaison

Implements [IA\\_Decodeur](#).

Here is the caller graph for this function:



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

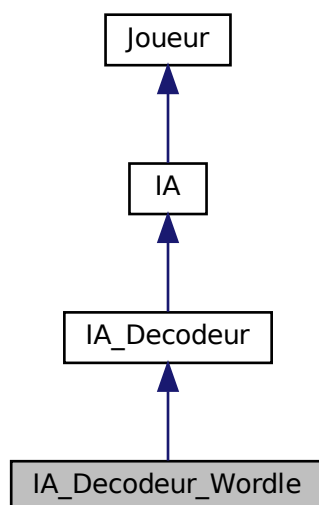
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Decodeur\_Mastermind.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/IA\_Decodeur\_Mastermind.cpp

## 6.18 IA\_Decodeur\_Wordle Class Reference

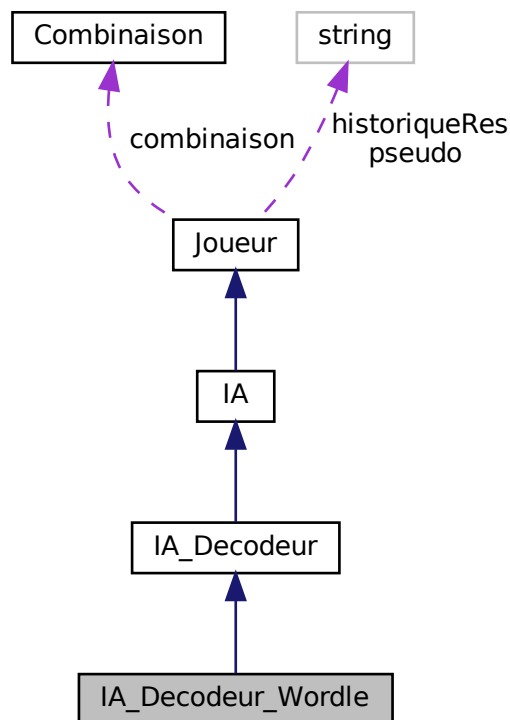
Cette classe est utilisée pour représenter un joueur virtuel ([IA](#)) en mode Decodeur dans le mode de jeu [Wordle](#).

```
#include <IA_Decodeur_Wordle.hpp>
```

Inheritance diagram for `IA_Decodeur_Wordle`:



Collaboration diagram for `IA_Decodeur_Wordle`:



## Public Member Functions

- [IA\\_Decodeur\\_Wordle](#) ()  
*Constructeur de la classe [IA\\_Decodeur\\_Wordle](#).*
- virtual vector< [Combinaison](#) > [Combi\\_possible](#) ()  
*Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons de couleurs possibles.*
- virtual void [Maj\\_ensemble](#) ([Combinaison](#) combi, string resultat)  
*Si la proposition est incorrecte, cette méthode permet retirer de S tout code qui ne donnerait pas la même réponse si la supposition actuelle était le code.*
- [Combinaison](#) [choisirCombinaison](#) ()  
*Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.*
- virtual void [jouer](#) ()  
*Méthode qui Permet de lancer la partie wordle avec [IA](#) décodeur.*

## Additional Inherited Members

### 6.18.1 Detailed Description

Cette classe est utilisée pour représenter un joueur virtuel ([IA](#)) en mode Decodeur dans le mode de jeu [Wordle](#).

**Author**

Groupe A7

**Version**

1.0

**Date**

avril 2022

## 6.18.2 Member Function Documentation

### 6.18.2.1 choisirCombinaison()

```
IA_Decodeur_Wordle::choisirCombinaison ( ) [virtual]
```

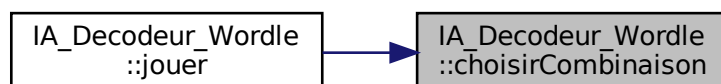
Cette méthode permet de a l'[IA](#) de choisir une combinaison parmi l'ensemble.

**Returns**

retourne la combinaison choisie

Implements [IA](#).

Here is the caller graph for this function:



### 6.18.2.2 Combi\_possible()

```
IA_Decodeur_Wordle::Combi_possible ( ) [virtual]
```

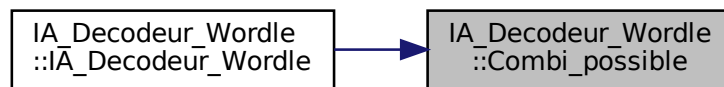
Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons de couleurs possibles.

#### Returns

retourne une liste de combinaison

Implements [IA\\_Decodeur](#).

Here is the caller graph for this function:



### 6.18.2.3 Maj\_ensemble()

```
IA_Decodeur_Wordle::Maj_ensemble (
    Combinaison combi,
    string resultat ) [virtual]
```

Si la proposition est incorrecte, cette méthode permet retirer de S tout code qui ne donnerait pas la même réponse si la supposition actuelle était le code.

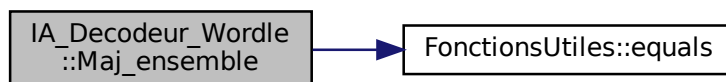
#### Parameters

<i>combi</i>	la combinaison saisie
<i>resultat</i>	string

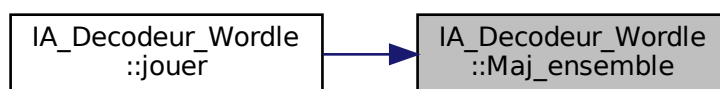
Implements [IA\\_Decodeur](#).



Here is the call graph for this function:



Here is the caller graph for this function:



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

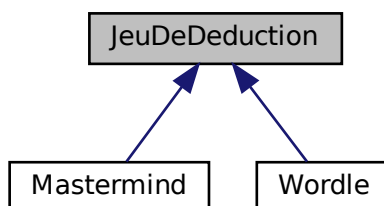
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.cpp`

## 6.19 JeuDeDeduction Class Reference

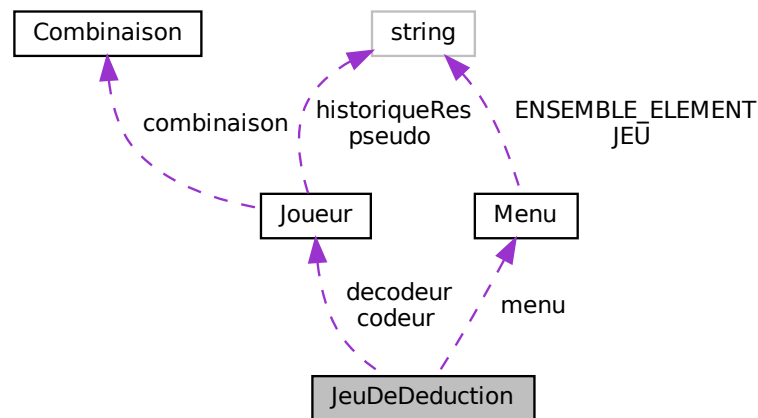
Classe principale permettant de lancement d'une partie. Elle comprend le main et.

```
#include <JeuDeDeduction.hpp>
```

Inheritance diagram for JeuDeDeduction:



Collaboration diagram for JeuDeDeduction:



## Public Member Functions

- [JeuDeDeduction](#) ()  
*Constructeur neutre de la classe [JeuDeDeduction](#).*
- **JeuDeDeduction** ([Joueur](#) \*codeur1, [Joueur](#) \*decodeur1)
- virtual [~JeuDeDeduction](#) ()  
*Destructeur de la classe [JeuDeDeduction](#).*
- virtual void [partie](#) ()=0  
*Cette méthode abstraite permet de lancer une partie une fois que les paramètres et le mode de jeu ont bien été choisis par le joueur s'il le souhaite.*
- virtual void **afficherHistorique** ()=0
- [Joueur](#) \* [detectionVictoire](#) ()  
*Cette méthode permet de savoir s'il y a un gagnant au cours de la partie ou non.*
- int [getTour](#) ()  
*Cette méthode permet de savoir le numéro du tour sur lequel le joueur est en train de jouer.*
- void [incrTour](#) ()  
*Cette méthode permet d'incrémenter de 1 l'attribut numeroTour.*

## Protected Attributes

- [Joueur](#) \* **codeur**
- [Joueur](#) \* **decodeur**
- vector< [Combinaison](#) > **historiqueCombinaison**
- int **numeroTour**
- [Menu](#) **menu**
- vector< string > **historiqueResultat**

### 6.19.1 Detailed Description

Classe principale permettant de lancement d'une partie. Elle comprend le main et.

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.19.2 Constructor & Destructor Documentation

#### 6.19.2.1 JeuDeDeduction()

```
JeuDeDeduction::JeuDeDeduction ( )
```

Constructeur neutre de la classe [JeuDeDeduction](#).

Constructeur neutre de la classe [JeuDeDeduction](#) avec paramètres.

#### Parameters

<i>codeur</i>	: joueur codeur qui propose une combinaison
<i>decodeur</i>	: joueur décodeur qui va deviner la combinaison

### 6.19.3 Member Function Documentation

#### 6.19.3.1 detectionVictoire()

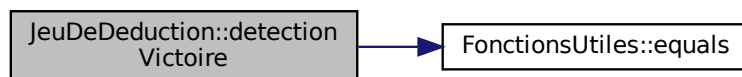
```
Joueur JeuDeDeduction::detectionVictoire ( )
```

Cette méthode permet de savoir s'il y a un gagnant au cours de la partie ou non.

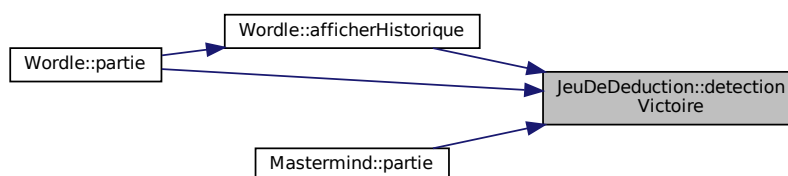
**Returns**

Renvoie le joueur gagnant s'il en existe un, sinon elle retourne null

Here is the call graph for this function:



Here is the caller graph for this function:

**6.19.3.2 getTour()**

```
int JeuDeDeduction::getTour ( )
```

Cette méthode permet de savoir le numéro du tour sur laquelle le joueur est en train de jouer.

**Returns**

renvoie un entier

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

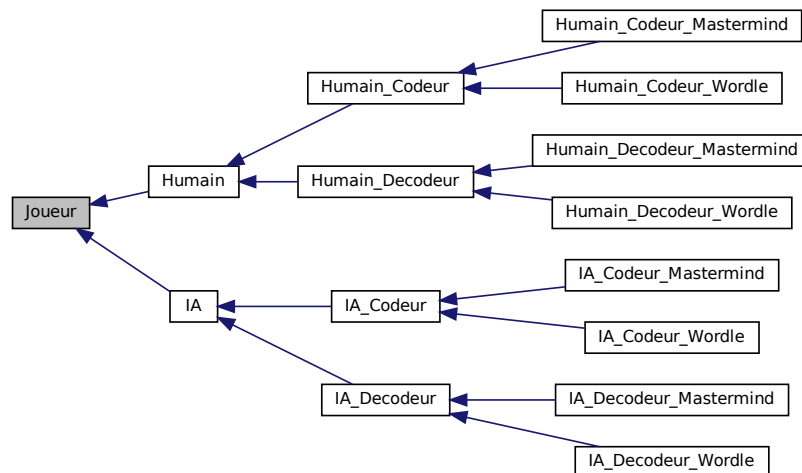
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.cpp`

## 6.20 Joueur Class Reference

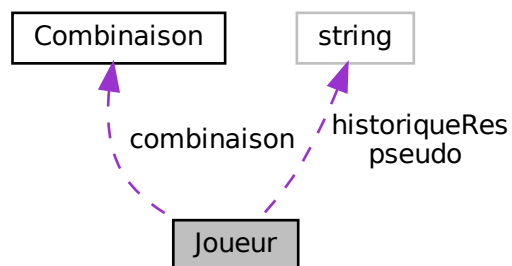
classe abstraite regroupant tous les joueurs (humain et IA)

```
#include <Joueur.hpp>
```

Inheritance diagram for Joueur:



Collaboration diagram for Joueur:



### Public Member Functions

- `Joueur()`  
*Constructeur de la classe joueur.*
- `Joueur(string pseudo1)`
- `Combinaison getCombinaison()`  
*Pour accéder à l'attribut combinaison.*

- void `setCombinaison` (`Combinaison` comb)  
*setteur de l'attribut combinaison*
- void `entrerPseudo` ()  
*Pour permettre au joueur de choisir un pseudo.*
- string `getPseudo` ()  
*getteur de l'attribut Pseudo*
- void `setPseudo` (string nom)  
*setteur de l'attribut Pseudo*
- string `toString` ()  
*permet l'affichage du pseudo*
- void `set_historiqueRes` (string res)  
*setteur de l'attribut historiqueRes*
- virtual void `jouer` ()=0  
*méthode abstraite qui sera définie dans les classes filles*

## Protected Attributes

- `Combinaison` **combinaison**
- string **pseudo**
- string **historiqueRes**

### 6.20.1 Detailed Description

classe abstraite regroupant tous les joueurs (humain et [IA](#))

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.20.2 Constructor & Destructor Documentation

#### 6.20.2.1 Joueur()

```
Joueur::Joueur ( )
```

Constructeur de la classe joueur.

Constructeur de la classe joueur avec paramètre.

**Parameters**

<i>pseudo</i>	du joueur
---------------	-----------

## 6.20.3 Member Function Documentation

### 6.20.3.1 getCombinaison()

```
Joueur::getCombinaison ( )
```

Pour accéder à l'attribut combinaison.

**Returns**

retourne la combinaison en attribut

### 6.20.3.2 getPseudo()

```
Joueur::getPseudo ( )
```

getteur de l'attribut Pseudo

**Returns**

retourne le pseudo en attribut

### 6.20.3.3 set\_historiqueRes()

```
Joueur::set_historiqueRes (
    string res )
```

setteur de l'attribut historiqueRes

**Parameters**

<i>res</i>	un string
------------	-----------

#### 6.20.3.4 setCombinaison()

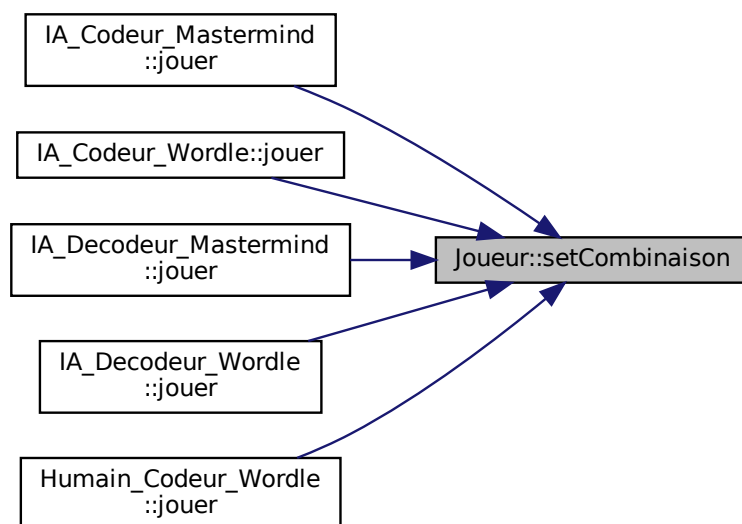
```
Joueur::setCombinaison (
    Combinaison comb )
```

setteur de l'attribut combinaison

##### Parameters

<i>comb</i>	une combinaison
-------------	-----------------

Here is the caller graph for this function:



#### 6.20.3.5 setPseudo()

```
Joueur::setPseudo (
    string nom )
```

setteur de l'attribut Pseudo

##### Parameters

<i>nom</i>	un string
------------	-----------



### 6.20.3.6 toString()

```
Joueur::toString ( )
```

permet l'affichage du pseudo

#### Returns

un string

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

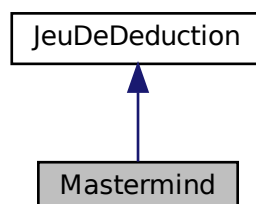
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Joueur.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Joueur.cpp

## 6.21 Mastermind Class Reference

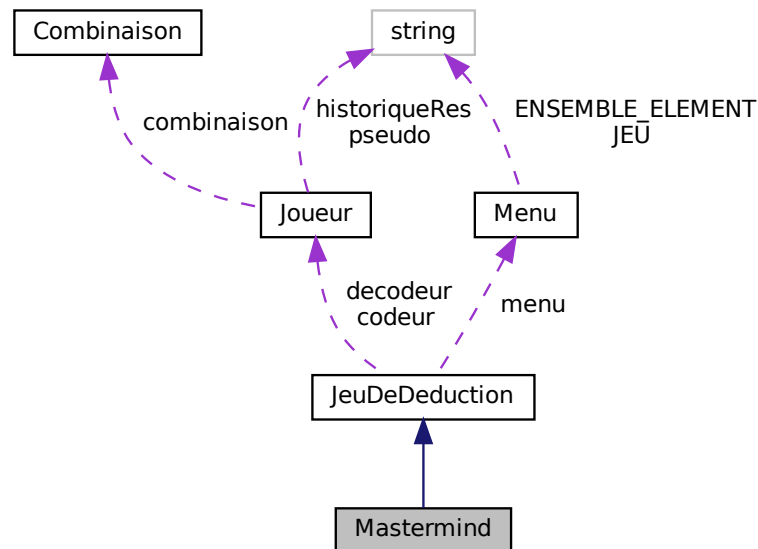
Classe principale permettant de lancement d'une partie de [Mastermind](#). Elle définit les méthodes abstraites présente dans [JeuDeDeduction](#).

```
#include <Mastermind.hpp>
```

Inheritance diagram for Mastermind:



Collaboration diagram for Mastermind:



## Public Member Functions

- [Mastermind](#) ()  
*Constructeur neutre de la classe [Mastermind](#).*
- **Mastermind** ([Joueur](#) \*codeur1, [Joueur](#) \*decodeur1)
- void [partie](#) ()  
*Elle permet de lancer une partie de [Mastermind](#).*
- virtual void **afficherHistorique** ()

## Additional Inherited Members

### 6.21.1 Detailed Description

Classe principale permettant de lancement d'une partie de [Mastermind](#). Elle définit les méthodes abstraites présente dans [JeuDeDeduction](#).

Author

Groupe A7

Version

1.0

Date

avril 2022

## 6.21.2 Constructor & Destructor Documentation

### 6.21.2.1 Mastermind()

```
Mastermind::Mastermind ( )
```

Constructeur neutre de la classe [Mastermind](#).

Constructeur neutre de la classe [Mastermind](#) avec paramètres.

#### Parameters

<i>codeur</i>	: joueur codeur qui propose une combinaison
<i>decodeur</i>	: joueur décodeur qui va deviner la combinaison

## 6.21.3 Member Function Documentation

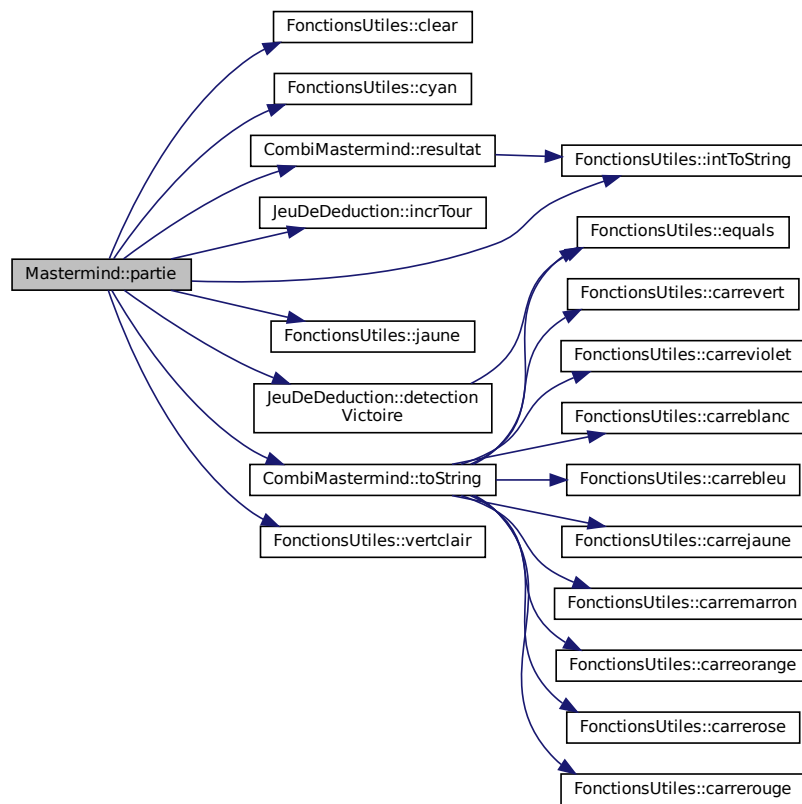
### 6.21.3.1 partie()

```
Mastermind::partie ( ) [virtual]
```

Elle permet de lancer une partie de [Mastermind](#).

Implements [JeuDeDeduction](#).

Here is the call graph for this function:



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

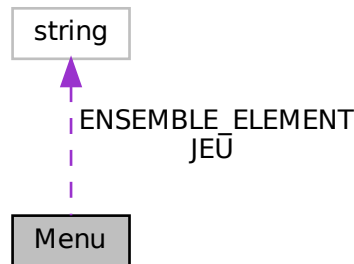
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.cpp`

## 6.22 Menu Class Reference

La classe [Menu](#) permet à l'utilisateur de choisir le mode et les paramètres du jeu.

```
#include <Menu.hpp>
```

Collaboration diagram for Menu:



## Public Member Functions

- [Menu](#) ()  
*Constructeur de la classe [Menu](#).*
- [Menu](#) (int nbcase, int nbtour, int langue)  
*Constructeur de la classe [Menu](#) avec paramètres.*
- void [afficherMenu](#) ()  
*affiche le menu.*
- void [choisirJeu](#) (const string &jeu)  
*Fonction qui permet à l'utilisateur de choisir le jeu.*
- void [choisirModeDeJeu](#) (int mdj)
- void [choisirDifficulteA](#) ()  
*Fonction qui permet à l'utilisateur de choisir la difficulté de l'IA.*
- void [parametreDeJeu](#) (int NB\_ELEMENT, string ENSEMBLE\_ELEMENT, int NB\_CASE, int NB\_TOUR)  
*Fonction qui permet à l'utilisateur de choisir le parametre de jeu.*
- bool [appartientEE](#) (string couleur)  
*Fonction qui permet de vérifier si une combinaison de couleurs est valide syntaxiquement ou non.*
- bool [appartientDico](#) (string mot)  
*Fonction qui permet de vérifier si un mot est reconnu par le dictionnaire [Wordle](#) ou non.*

## Static Public Attributes

- static string **JEU** = "Mastermind"
- static int **NB\_ELEMENT** =9
- static string **ENSEMBLE\_ELEMENT** ="ressources"
- static int **NB\_CASE** =4
- static int **NB\_TOUR** =12
- static int **Langue** =2
- static int **ModeDeJeu** = 1

### 6.22.1 Detailed Description

La classe [Menu](#) permet à l'utilisateur de choisir le mode et les paramètres du jeu.

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

### 6.22.2 Constructor & Destructor Documentation

#### 6.22.2.1 Menu()

```
Menu::Menu (
    int nbcase,
    int nbtour,
    int langue )
```

Constructeur de la classe [Menu](#) avec paramètres.

#### Parameters

<i>nbcase</i>	un entier pour définir le nombre de case
<i>nbtour</i>	un entier pour définir le nombre de tour de la partie
<i>langue</i>	un entier (1 ou 2) pour définir la langue de jeu dans le cas de wordle

### 6.22.3 Member Function Documentation

#### 6.22.3.1 appartientDico()

```
Menu::appartientDico (
    string mot )
```

Fonction qui permet de vérifier si un mot est reconnu par le dictionnaire [Wordle](#) ou non.

## Parameters

<i>mot</i>	est une combinaison qui correspond à un mot de la langue française
------------	--

### 6.22.3.2 appartientEE()

```
Menu::appartientEE (
    string couleur )
```

Fonction qui permet de vérifier si une combinaison de couleurs est valide syntaxiquement ou non.

## Parameters

<i>couleur</i>	est un string qui correspond à une succession de couleurs différentes définissant une combinaison
----------------	---

### 6.22.3.3 choisirJeu()

```
Menu::choisirJeu (
    const string & jeu )
```

Fonction qui permet à l'utilisateur de choisir le jeu.

## Parameters

<i>jeu</i>	est une chaîne de caractère qui indique le jeu choisi
------------	---

Here is the call graph for this function:



### 6.22.3.4 parametreDeJeu()

```
Menu::parametreDeJeu (
    int NB_ELEMENT,
```

```

    string ENSEMBLE_ELEMENT,
    int NB_CASE,
    int NB_TOUR )

```

Fonction qui permet à l'utilisateur de choisir le parametre de jeu.

#### Parameters

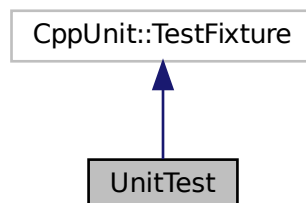
<i>NB_ELEMENT</i>	indique le nombre d'éléments constitutifs parmi lesquels on choisit pour la combinaison à faire deviner
<i>NB_CASE</i>	indique la taille du tableau d'entier combinaison
<i>ENSEMBLE_ELEMENT</i>	définit si on choisit de jouer avec des couleurs ou des lettres
<i>NB_TOUR</i>	le nombre de tours maximale pour faire deviner la combinaison

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

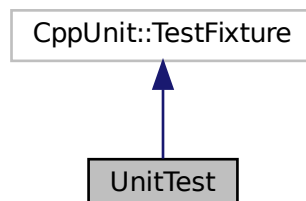
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Menu.hpp
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/Menu.cpp

## 6.23 UnitTest Class Reference

Inheritance diagram for UnitTest:



Collaboration diagram for UnitTest:





## Public Member Functions

- void **setUp** (void)
- void **tearDown** (void)

## Protected Member Functions

- void **testGetI** (void)
- void **testResultatMastermind** (void)
- void **testResultatWordle** (void)
- void **testSplit** (void)
- void **testCheckMotFichier** (void)
- void **testIsNumber** (void)
- void **testIntToString** (void)
- void **testCharToString** (void)
- void **testEquals** (void)

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

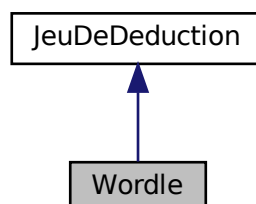
- /home/kevin/Documents/GM4/S8/Projet\_Cpp\_S8/UnitTest.cpp

## 6.24 Wordle Class Reference

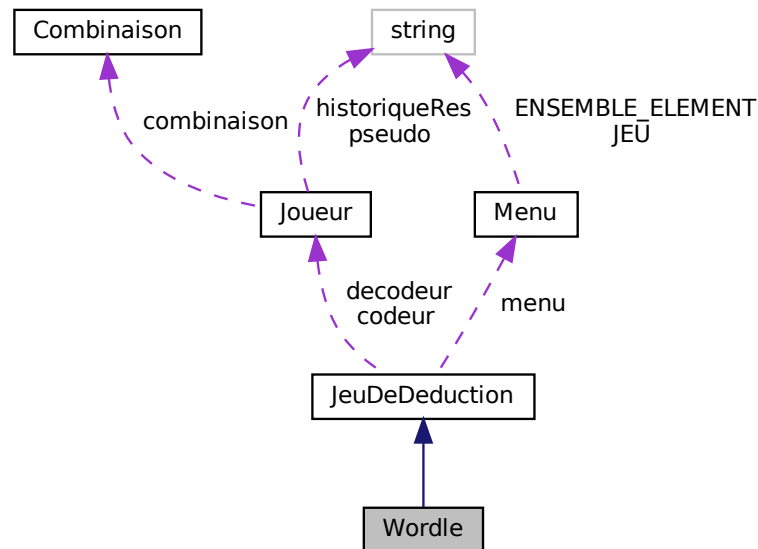
Classe principale permettant de lancement d'une partie de [Wordle](#). Elle définit les méthodes abstraites présente dans [JeuDeDeduction](#).

```
#include <Wordle.hpp>
```

Inheritance diagram for Wordle:



Collaboration diagram for Wordle:



## Public Member Functions

- [Wordle](#) ()  
*Constructeur neutre de la classe [Wordle](#).*
- **Wordle** ([Joueur](#) \*codeur1, [Joueur](#) \*decodeur1)
- virtual void [partie](#) ()  
*Elle permet de lancer une partie de [Wordle](#).*
- virtual void [afficherHistorique](#) ()  
*Cette méthode permet d'afficher le jeu entier comprenant les historiques des combinaisons rentrées avec le résultat de comparaison avec le code associé (nombre de couleur bien placé et mal placé).*
- void [choisirFichierDeMot](#) ()  
*Cette méthode permet de choisir le fichier de mot qui sera alors le dictionnaire pour le [Wordle](#).*

## Additional Inherited Members

### 6.24.1 Detailed Description

Classe principale permettant de lancement d'une partie de [Wordle](#). Elle définit les méthodes abstraites présente dans [JeuDeDeduction](#).

Author

Groupe A7

Version

1.0

Date

avril 2022

## 6.24.2 Constructor & Destructor Documentation

### 6.24.2.1 Wordle()

```
Wordle::Wordle ( )
```

Constructeur neutre de la classe [Wordle](#).

Constructeur de la classe [Wordle](#) avec les paramètres.

#### Parameters

<i>codeur</i>	: joueur codeur qui propose une combinaison
<i>decodateur</i>	: joueur décodeur qui va deviner la combinaison

## 6.24.3 Member Function Documentation

### 6.24.3.1 afficherHistorique()

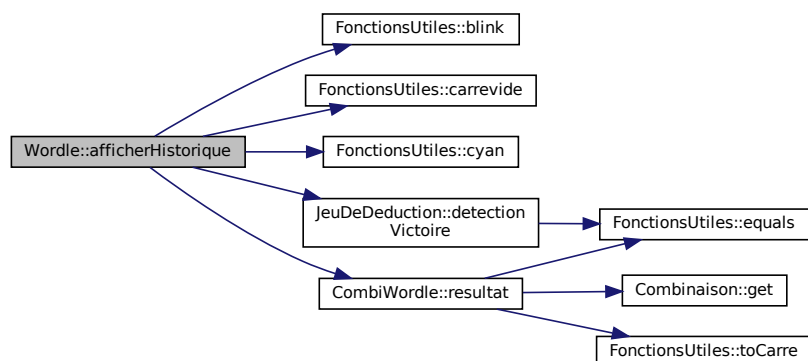
```
void Wordle::afficherHistorique ( ) [virtual]
```

Cette méthode permet d'afficher le jeu entier comprenant les historiques des combinaisons rentrées avec le résultat de comparaison avec le code associé (nombre de couleur bien placé et mal placé).

Cette méthode ne permet pas d'afficher le code quand nous sommes du point de vue décodeur.

Implements [JeuDeDeduction](#).

Here is the call graph for this function:



Here is the caller graph for this function:



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

- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.hpp`
- `/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.cpp`

# Index

afficherHistorique  
    Wordle, [99](#)  
afficherResultat  
    CombiWordle, [20](#)  
appartientDico  
    Menu, [94](#)  
appartientEE  
    Menu, [95](#)  
  
blanc  
    FonctionsUtiles, [24](#)  
bleu  
    FonctionsUtiles, [25](#)  
blink  
    FonctionsUtiles, [25](#)  
  
carreblanc  
    FonctionsUtiles, [26](#)  
carrebleu  
    FonctionsUtiles, [26](#)  
carrejaune  
    FonctionsUtiles, [27](#)  
carremarron  
    FonctionsUtiles, [27](#)  
carreorange  
    FonctionsUtiles, [28](#)  
carrerose  
    FonctionsUtiles, [28](#)  
carrerouge  
    FonctionsUtiles, [29](#)  
carrevert  
    FonctionsUtiles, [29](#)  
carrevide  
    FonctionsUtiles, [30](#)  
carreviolet  
    FonctionsUtiles, [30](#)  
charToString  
    FonctionsUtiles, [30](#)  
checkMotFichier  
    FonctionsUtiles, [31](#)  
choisirCombinaison  
    IA, [61](#)  
    IA\_Codeur\_Mastermind, [67](#)  
    IA\_Codeur\_Wordle, [70](#)  
    IA\_Decodeur\_Mastermind, [76](#)  
    IA\_Decodeur\_Wordle, [79](#)  
choisirJeu  
    Menu, [95](#)  
Combi\_possible  
    IA\_Decodeur, [74](#)  
  
    IA\_Decodeur\_Mastermind, [76](#)  
    IA\_Decodeur\_Wordle, [79](#)  
CombiMastermind, [11](#)  
    CombiMastermind, [12](#)  
    resultat, [13](#)  
    toString, [13](#)  
Combinaison, [15](#)  
    Combinaison, [16](#)  
    get, [16](#)  
    resultat, [16](#)  
    setCombinaison, [17](#)  
    toString, [17](#)  
CombiWordle, [18](#)  
    afficherResultat, [20](#)  
    CombiWordle, [19](#)  
    resultat, [20](#)  
    resultat\_couleur, [21](#)  
    toString, [22](#)  
cyan  
    FonctionsUtiles, [32](#)  
  
detectionVictoire  
    JeuDeDeduction, [83](#)  
  
entrerCode  
    Humain\_Codeur\_Mastermind, [46](#)  
    Humain\_Codeur\_Wordle, [50](#)  
entrerCombinaison  
    Humain\_Decodeur\_Mastermind, [56](#)  
entrerMdP  
    Humain\_Codeur, [44](#)  
equals  
    FonctionsUtiles, [32](#)  
  
FonctionsUtiles, [23](#)  
    blanc, [24](#)  
    bleu, [25](#)  
    blink, [25](#)  
    carreblanc, [26](#)  
    carrebleu, [26](#)  
    carrejaune, [27](#)  
    carremarron, [27](#)  
    carreorange, [28](#)  
    carrerose, [28](#)  
    carrerouge, [29](#)  
    carrevert, [29](#)  
    carrevide, [30](#)  
    carreviolet, [30](#)  
    charToString, [30](#)  
    checkMotFichier, [31](#)

- cyan, 32
- equals, 32
- intToString, 33
- isNumber, 34
- jaune, 34
- rouge, 35
- split, 36
- toCarre, 37
- vert, 37
- vertclair, 38
- violet, 38
- violetclair, 39
- genererCode
  - IA\_Codeur, 64
  - IA\_Codeur\_Mastermind, 67
  - IA\_Codeur\_Wordle, 71
- genererMdP
  - IA\_Codeur, 64
- get
  - Combinaison, 16
- getCombinaison
  - Joueur, 87
- getMdP
  - Humain\_Codeur, 44
  - IA\_Codeur, 64
- getPseudo
  - Joueur, 87
- getTour
  - JeuDeDeduction, 84
- Humain, 39
  - Humain, 41
  - verifierEntree, 41
- Humain\_Codeur, 42
  - entrerMdP, 44
  - getMdP, 44
- Humain\_Codeur\_Mastermind, 44
  - entrerCode, 46
  - verifierEntree, 47
- Humain\_Codeur\_Wordle, 48
  - entrerCode, 50
  - verifierEntree, 51
- Humain\_Decodeur, 52
- Humain\_Decodeur\_Mastermind, 54
  - entrerCombinaison, 56
  - verifierEntree, 56
- Humain\_Decodeur\_Wordle, 57
  - verifierEntree, 59
- IA, 60
  - choisirCombinaison, 61
  - IA, 61
- IA\_Codeur, 62
  - genererCode, 64
  - genererMdP, 64
  - getMdP, 64
- IA\_Codeur\_Mastermind, 65
  - choisirCombinaison, 67
  - genererCode, 67
- IA\_Codeur\_Wordle, 68
  - choisirCombinaison, 70
  - genererCode, 71
- IA\_Decodeur, 72
  - Combi\_possible, 74
- IA\_Decodeur\_Mastermind, 74
  - choisirCombinaison, 76
  - Combi\_possible, 76
- IA\_Decodeur\_Wordle, 77
  - choisirCombinaison, 79
  - Combi\_possible, 79
  - Maj\_ensemble, 80
- intToString
  - FonctionsUtiles, 33
- isNumber
  - FonctionsUtiles, 34
- jaune
  - FonctionsUtiles, 34
- JeuDeDeduction, 81
  - detectionVictoire, 83
  - getTour, 84
  - JeuDeDeduction, 83
- Joueur, 85
  - getCombinaison, 87
  - getPseudo, 87
  - Joueur, 86
  - set\_historiqueRes, 87
  - setCombinaison, 87
  - setPseudo, 88
  - toString, 88
- Maj\_ensemble
  - IA\_Decodeur\_Wordle, 80
- Mastermind, 89
  - Mastermind, 91
  - partie, 91
- Menu, 92
  - appartientDico, 94
  - appartientEE, 95
  - choisirJeu, 95
  - Menu, 94
  - parametreDeJeu, 95
- parametreDeJeu
  - Menu, 95
- partie
  - Mastermind, 91
- resultat
  - CombiMastermind, 13
  - Combinaison, 16
  - CombiWordle, 20
- resultat\_couleur
  - CombiWordle, 21
- rouge
  - FonctionsUtiles, 35
- set\_historiqueRes

- Joueur, [87](#)
- setCombinaison
  - Combinaison, [17](#)
  - Joueur, [87](#)
- setPseudo
  - Joueur, [88](#)
- split
  - FonctionsUtiles, [36](#)
- std, [9](#)
- toCarre
  - FonctionsUtiles, [37](#)
- toString
  - CombiMastermind, [13](#)
  - Combinaison, [17](#)
  - CombiWordle, [22](#)
  - Joueur, [88](#)
- UnitTest, [96](#)
- verifierEntree
  - Humain, [41](#)
  - Humain\_Codeur\_Mastermind, [47](#)
  - Humain\_Codeur\_Wordle, [51](#)
  - Humain\_Decodeur\_Mastermind, [56](#)
  - Humain\_Decodeur\_Wordle, [59](#)
- vert
  - FonctionsUtiles, [37](#)
- vertclair
  - FonctionsUtiles, [38](#)
- violet
  - FonctionsUtiles, [38](#)
- violetclair
  - FonctionsUtiles, [39](#)
- Wordle, [97](#)
  - afficherHistorique, [99](#)
  - Wordle, [99](#)