

Jeu de déduction

Generated by Doxygen 1.8.17

1	Projet Cpp S8	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	std Namespace Reference	11
6.1.1	Detailed Description	11
7	Class Documentation	13
7.1	CombiMastermind Class Reference	13
7.1.1	Detailed Description	14
7.1.2	Constructor & Destructor Documentation	14
7.1.2.1	CombiMastermind() [1/2]	14
7.1.2.2	CombiMastermind() [2/2]	14
7.1.3	Member Function Documentation	14
7.1.3.1	bienPlace()	14
7.1.3.2	malPlace()	15
7.1.3.3	toString()	15
7.2	Combinaison Class Reference	16
7.2.1	Detailed Description	16
7.2.2	Constructor & Destructor Documentation	17
7.2.2.1	Combinaison() [1/2]	17
7.2.2.2	Combinaison() [2/2]	17
7.2.3	Member Function Documentation	17
7.2.3.1	get() [1/2]	17
7.2.3.2	get() [2/2]	17
7.2.3.3	setCombinaison()	17
7.2.3.4	toString()	18
7.3	CombiWordle Class Reference	18
7.3.1	Detailed Description	19
7.3.2	Constructor & Destructor Documentation	19
7.3.2.1	CombiWordle() [1/2]	20
7.3.2.2	CombiWordle() [2/2]	20
7.3.3	Member Function Documentation	20

7.3.3.1 bienPlace()	20
7.3.3.2 malPlace()	20
7.3.3.3 toString()	21
7.4 Humain Class Reference	21
7.4.1 Detailed Description	22
7.4.2 Member Function Documentation	22
7.4.2.1 split()	22
7.4.2.2 verifierEntree()	23
7.5 Humain_Codeur Class Reference	23
7.5.1 Detailed Description	24
7.5.2 Constructor & Destructor Documentation	25
7.5.2.1 Humain_Codeur()	25
7.5.3 Member Function Documentation	25
7.5.3.1 entrerCode()	25
7.5.3.2 entrerMdP()	25
7.5.3.3 getMdP()	25
7.6 Humain_Codeur_Mastermind Class Reference	26
7.6.1 Detailed Description	27
7.7 Humain_Codeur_Wordle Class Reference	27
7.7.1 Detailed Description	28
7.7.2 Constructor & Destructor Documentation	29
7.7.2.1 Humain_Codeur_Wordle()	29
7.7.3 Member Function Documentation	29
7.7.3.1 entrerCode()	29
7.7.3.2 VerifierEntree()	29
7.8 Humain_Decodeur Class Reference	30
7.8.1 Detailed Description	31
7.8.2 Constructor & Destructor Documentation	31
7.8.2.1 Humain_Decodeur()	31
7.8.3 Member Function Documentation	31
7.8.3.1 entrerCombinaison()	31
7.9 Humain_Decodeur_Mastermind Class Reference	32
7.9.1 Detailed Description	33
7.9.2 Constructor & Destructor Documentation	33
7.9.2.1 Humain_Decodeur_Mastermind()	33
7.9.3 Member Function Documentation	33
7.9.3.1 entrerCombinaison()	34
7.9.3.2 VerifierEntree()	34
7.10 Humain_Decodeur_Wordle Class Reference	34
7.10.1 Detailed Description	35
7.10.2 Constructor & Destructor Documentation	36
7.10.2.1 Humain_Decodeur_Wordle()	36

7.10.3 Member Function Documentation	36
7.10.3.1 entrerCombinaison()	36
7.10.3.2 VerifierEntree()	36
7.11 IA Class Reference	37
7.11.1 Detailed Description	37
7.11.2 Member Function Documentation	38
7.11.2.1 choisirCombinaison()	38
7.12 IA_Codeur Class Reference	38
7.12.1 Detailed Description	39
7.12.2 Constructor & Destructor Documentation	39
7.12.2.1 IA_Codeur()	40
7.12.3 Member Function Documentation	40
7.12.3.1 genererCode()	40
7.12.3.2 genererMdP()	40
7.12.3.3 getMdP()	40
7.13 IA_Codeur_Mastermind Class Reference	41
7.13.1 Detailed Description	42
7.13.2 Constructor & Destructor Documentation	42
7.13.2.1 IA_Codeur_Mastermind()	42
7.13.3 Member Function Documentation	42
7.13.3.1 genererCode()	42
7.14 IA_Codeur_Wordle Class Reference	43
7.14.1 Detailed Description	44
7.14.2 Constructor & Destructor Documentation	44
7.14.2.1 IA_Codeur_Wordle()	44
7.14.3 Member Function Documentation	44
7.14.3.1 genererCode()	44
7.15 IA_Decodeur Class Reference	45
7.15.1 Detailed Description	46
7.15.2 Constructor & Destructor Documentation	46
7.15.2.1 IA_Decodeur()	46
7.15.3 Member Function Documentation	46
7.15.3.1 Choisir_combi()	47
7.15.3.2 Combi_possible()	47
7.15.3.3 Maj_ensemble()	47
7.16 IA_Decodeur_Mastermind Class Reference	48
7.16.1 Detailed Description	49
7.16.2 Constructor & Destructor Documentation	49
7.16.2.1 IA_Decodeur_Mastermind()	49
7.16.3 Member Function Documentation	49
7.16.3.1 Choisir_combi()	50
7.16.3.2 Combi_possible()	50

7.16.3.3 Maj_ensemble()	50
7.17 IA_Decodeur_Wordle Class Reference	51
7.17.1 Detailed Description	52
7.17.2 Constructor & Destructor Documentation	52
7.17.2.1 IA_Decodeur_Wordle()	52
7.17.3 Member Function Documentation	52
7.17.3.1 Choisir_combi()	53
7.17.3.2 Combi_possible()	53
7.17.3.3 Maj_ensemble()	53
7.18 JeuDeDeduction Class Reference	54
7.18.1 Detailed Description	55
7.18.2 Constructor & Destructor Documentation	55
7.18.2.1 JeuDeDeduction() [1/2]	55
7.18.2.2 JeuDeDeduction() [2/2]	55
7.18.3 Member Function Documentation	55
7.18.3.1 afficherCode()	55
7.18.3.2 afficherPartie()	56
7.18.3.3 detectionVictoire()	56
7.18.3.4 getTour()	56
7.18.3.5 incrTour()	56
7.18.3.6 main()	56
7.18.3.7 partie()	57
7.19 Joueur Class Reference	57
7.19.1 Detailed Description	58
7.19.2 Constructor & Destructor Documentation	58
7.19.2.1 Joueur() [1/2]	58
7.19.2.2 Joueur() [2/2]	58
7.19.3 Member Function Documentation	58
7.19.3.1 entrerPseudo()	58
7.19.3.2 getCombinaison()	59
7.19.3.3 getPseudo()	59
7.19.3.4 setCombinaison()	59
7.19.3.5 setPseudo()	59
7.19.3.6 verification()	60
7.20 Mastermind Class Reference	60
7.20.1 Detailed Description	61
7.21 Menu Class Reference	61
7.21.1 Detailed Description	62
7.21.2 Constructor & Destructor Documentation	62
7.21.2.1 Menu()	62
7.21.3 Member Function Documentation	63
7.21.3.1 afficherMenu()	63

7.21.3.2 appartientDico()	63
7.21.3.3 appartientEE()	63
7.21.3.4 choisirDifficulteIA()	63
7.21.3.5 choisirModeDeJeu()	64
7.21.3.6 parametreDeJeu()	64
7.21.4 Member Data Documentation	64
7.21.4.1 ENSEMBLE_ELEMENT	64
7.21.4.2 ModeDeJeu	64
7.21.4.3 NB_CASE	64
7.21.4.4 NB_ELEMENT	65
7.21.4.5 NB_TOUR	65
7.22 Wordle Class Reference	65
7.22.1 Detailed Description	66
8 File Documentation	67
8.1 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp File Reference	67
8.2 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.cpp File Reference	68
8.3 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp File Reference	68
8.4 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiWordle.hpp File Reference	69
8.5 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain.hpp File Reference	70
8.6 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur.hpp File Reference	72
8.7 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp File Reference	73
8.8 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Wordle.hpp File Reference	74
8.9 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur.hpp File Reference	75
8.10 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp File Reference	77
8.11 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp File Reference	78
8.12 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA.hpp File Reference	79
8.13 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur.hpp File Reference	80
8.14 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.hpp File Reference	82
8.15 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Wordle.hpp File Reference	84
8.16 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur.hpp File Reference	85
8.17 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Mastermind.hpp File Reference	86
8.18 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp File Reference	88
8.19 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp File Reference	89
8.20 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.cpp File Reference	90
8.21 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp File Reference	91
8.22 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.cpp File Reference	92
8.23 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.hpp File Reference	93
8.24 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.cpp File Reference	95
8.25 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp File Reference	95
8.26 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/README.md File Reference	96
8.27 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.hpp File Reference	96

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 namespaces with brief descriptions:

std	11
-------------------------------	--------------------

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Combinaison	16
CombiMastermind	13
CombiWordle	18
JeuDeDeduction	54
Mastermind	60
Wordle	65
Joueur	57
Humain	21
Humain_Codeur	23
Humain_Codeur_Mastermind	26
Humain_Codeur_Wordle	27
Humain_Decodeur	30
Humain_Decodeur_Mastermind	32
Humain_Decodeur_Wordle	34
IA	37
IA_Codeur	38
IA_Codeur_Mastermind	41
IA_Codeur_Wordle	43
IA_Decodeur	45
IA_Decodeur_Mastermind	48
IA_Decodeur_Wordle	51
Menu	61

Chapter 4

Class Index

4.1 Class List

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

CombiMastermind	
Tableau d'entier représentant une combinaison de couleur	13
Combinaison	
Tableau d'entier représentant une combinaison de couleur ou de lettre	16
CombiWordle	
Tableau d'entier représentant une combinaison de lettre	18
Humain	
Si le joueur est un humain	21
Humain_Codeur	
Cette classe abstraite permet de décrire les joueurs humains en mode codeur peu importe le mode de jeu (Mastermind ou wordle)	23
Humain_Codeur_Mastermind	
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu Mastermind	26
Humain_Codeur_Wordle	
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu Wordle	27
Humain_Decodeur	
Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu (Mastermind ou wordle)	30
Humain_Decodeur_Mastermind	
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu Mastermind	32
Humain_Decodeur_Wordle	
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu Wordle	34
IA	
Si le joueur est un IA	37
IA_Codeur	
Cette classe abstraite permet de décrire l'intelligence artificielle en mode codeur peu importe le mode de jeu (Mastermind ou wordle)	38
IA_Codeur_Mastermind	
Cette classe est utilisée pour représenter un joueur artificiel (IA) en mode codeur dans le mode de jeu Mastermind	41

IA_Codeur_Wordle	Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode codeur dans le mode de jeu Wordle	43
IA_Decodeur	Cette classe abstraite permet de décrire l'intelligence artificielle en mode Decodeur peu importe le mode de jeu (Mastermind ou wordle)	45
IA_Decodeur_Mastermind	Cette classe est utilisée pour représenter un joueur artificiel (IA) en mode Decodeur dans le mode de jeu Mastermind	48
IA_Decodeur_Wordle	Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode Decodeur dans le mode de jeu Wordle	51
JeuDeDeduction	Classe principale permettant de lancement d'une partie. Elle comprend le main et	54
Joueur	Classe abstraite regroupant tout joueur	57
Mastermind	Classe principale permettant de lancement d'une partie. Elle comprend le main et	60
Menu	La classe Menu permet à l'utilisateur de choisir le mode et les parametres du jeu	61
Wordle	Classe principale permettant de lancement d'une partie de Wordle	65

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp	67
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.cpp	68
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp	68
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiWordle.hpp	69
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain.hpp	70
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur.hpp	72
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp	73
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Wordle.hpp	74
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur.hpp	75
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp	77
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp	78
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA.hpp	79
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur.hpp	80
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.hpp	82
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Wordle.hpp	84
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur.hpp	85
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Mastermind.hpp	86
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp	88
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp	89
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.cpp	90
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp	91
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/mastermind.cpp	92
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/mastermind.hpp	93
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.cpp	95
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp	95
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.hpp	96

Chapter 6

Namespace Documentation

6.1 std Namespace Reference

6.1.1 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. Elle possède un seul attribut.

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 l'objet [Combinaison](#). Elle transforme une chaîne de caractère (suite de couleur ou de lettre) séparée d'espace en tableau d'entier. Elle possède un seul attribut : le tableau d'entier.

Il s'agit d'une classe abstraite qui définit tout joueur du [Mastermind](#). On y retrouve deux attributs, le premier est un élément de la classe combinaison et le second est le nombre de tours écoulés qui est équivalent au nombre de combinaisons entrées par le joueur décodeur.

Cette classe permet de définir l'objet [Combinaison](#). Elle transforme une chaîne de caractère (suite de couleur) séparée d'espace en tableau d'entier.

Cette classe permet de définir l'objet [Combinaison](#). Elle transforme une chaîne de caractère (lettre) séparée d'espace en tableau d'entier.

Chapter 7

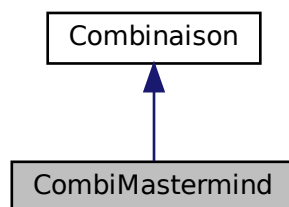
Class Documentation

7.1 CombiMastermind Class Reference

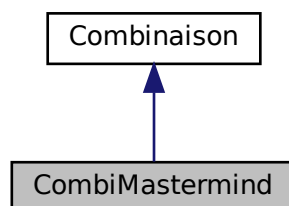
Tableau d'entier représentant une combinaison de couleur.

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

Inheritance diagram for CombiMastermind:



Collaboration diagram for CombiMastermind:



Public Member Functions

- [CombiMastermind](#) ()
Constructeur neutre de [CombiMastermind](#).
- [CombiMastermind](#) (string &chaine)
• int [bienPlace](#) (const [Combinaison](#) code)
renvoie le nombre de couleurs bien placées par rapport à la combinaison du codeur
- int [malPlace](#) (const [Combinaison](#) code)
renvoie le nombre de couleurs appartenant à la combinaison de couleur recherchée mais mal placées
- string [toString](#) ()
permet l'affichage de la combinaison de couleur

7.1.1 Detailed Description

Tableau d'entier représentant une combinaison de couleur.

Author

Groupe A7

Version

1.0

Date

avril 2022

7.1.2 Constructor & Destructor Documentation

7.1.2.1 [CombiMastermind\(\)](#) [1/2]

```
CombiMastermind::CombiMastermind ( )
```

Constructeur neutre de [CombiMastermind](#).

7.1.2.2 [CombiMastermind\(\)](#) [2/2]

```
CombiMastermind::CombiMastermind (
    string & chaine )
```

7.1.3 Member Function Documentation

7.1.3.1 [bienPlace\(\)](#)

```
int CombiMastermind::bienPlace (
    const Combinaison code )
```

renvoie le nombre de couleurs bien placées par rapport à la combinaison du codeur

Parameters

<i>code</i>	Combinaison : la combinaison de couleur valide recherchée par le décodeur
-------------	---

Returns

un entier

7.1.3.2 malPlace()

```
int CombiMastermind::malPlace (
    const Combinaison code )
```

renvoie le nombre de couleurs appartenant à la combinaison de couleur recherchée mais mal placées

Parameters

<i>code</i>	Combinaison : la combinaison de couleur valide recherchée par le décodeur
-------------	---

Returns

un entier

7.1.3.3 toString()

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

permet l'affichage de la combinaison de couleur

Returns

un string

Reimplemented from [Combinaison](#).

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

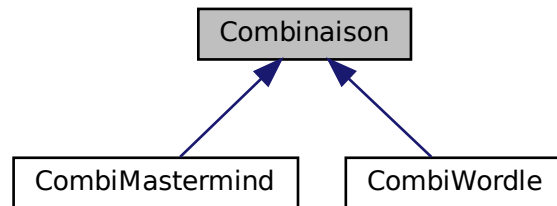
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp](#)

7.2 Combinaison Class Reference

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

```
#include <Combinaison.hpp>
```

Inheritance diagram for Combinaison:



Public Member Functions

- [Combinaison](#) ()
Constructeur neutre de [Combinaison](#).
- [Combinaison](#) (string &chaine)
- void [setCombinaison](#) (const string chaine)
setteur de l'attribut combinaison
- vector< string > [get](#) () const
getteur de l'attribut combinaison
- string [get](#) (const int i) const
- virtual string [toString](#) ()
permet l'affichage de la combinaison

7.2.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

7.2.2 Constructor & Destructor Documentation

7.2.2.1 Combinaison() [1/2]

```
Combinaison::Combinaison ( )
```

Constructeur neutre de [Combinaison](#).

7.2.2.2 Combinaison() [2/2]

```
Combinaison::Combinaison (
    string & chaine )
```

7.2.3 Member Function Documentation

7.2.3.1 get() [1/2]

```
Combinaison::get ( ) const
```

getteur de l'attribut combinaison

Returns

retourne un vecteur

7.2.3.2 get() [2/2]

```
string Combinaison::get (
    const int i ) const
```

7.2.3.3 setCombinaison()

```
void Combinaison::setCombinaison (
    const string chaine )
```

setteur de l'attribut combinaison

Parameters

<i>chaine</i>	string
---------------	--------

7.2.3.4 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:

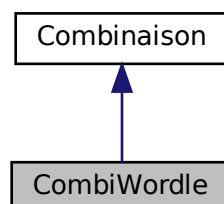
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp](#)
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.cpp](#)

7.3 CombiWordle Class Reference

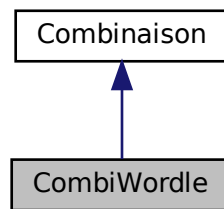
Tableau d'entier représentant une combinaison de lettre.

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

Inheritance diagram for CombiWordle:



Collaboration diagram for CombiWordle:



Public Member Functions

- `CombiWordle ()`
Constructeur neutre de `CombiWordle`.
- `CombiWordle (string &chaine)`
- `vector< int > bienPlace (const Combinaison code)`
renvoie les indices des lettres placées
- `vector< int > malPlace (const Combinaison code)`
renvoie les indices des lettres appartenant au mot recherché mais mal placées
- `string toString ()`
permet l'affichage du mot

7.3.1 Detailed Description

Tableau d'entier représentant une combinaison de lettre.

Author

Groupe A7

Version

1.0

Date

avril 2022

7.3.2 Constructor & Destructor Documentation

7.3.2.1 CombiWordle() [1/2]

```
CombiWordle::CombiWordle ( )
```

Constructeur neutre de [CombiWordle](#).

7.3.2.2 CombiWordle() [2/2]

```
CombiWordle::CombiWordle (
    string & chaine )
```

7.3.3 Member Function Documentation

7.3.3.1 bienPlace()

```
int CombiWordle::bienPlace (
    const Combinaison code )
```

renvoie les indices des lettres placées

Parameters

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

Returns

un tableau d'entier

7.3.3.2 malPlace()

```
int CombiWordle::malPlace (
    const Combinaison code )
```

renvoie les indices des lettres appartenant au mot recherché mais mal placées

Parameters

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

Returns

un tableau d'entier

7.3.3.3 toString()

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

permet l'affichage du mot

Returns

un string

Reimplemented from [Combinaison](#).

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

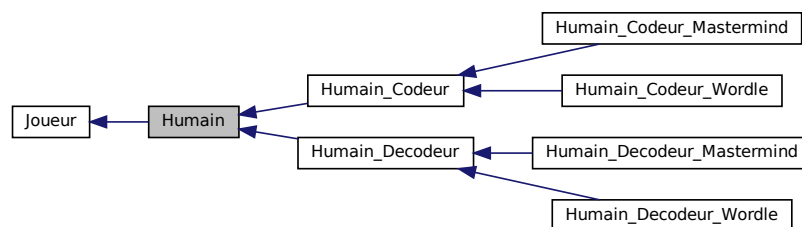
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiWordle.hpp](#)

7.4 Humain Class Reference

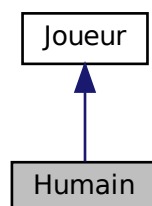
Si le joueur est un humain.

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

Inheritance diagram for Humain:



Collaboration diagram for Humain:



Public Member Functions

- virtual bool [verifierEntree](#) (string entree)
une méthode abstraite qui permettra de vérifier la syntaxe de l'entrée selon le mode de jeu
- vector< string > [split](#) (const string &chaine, char delimiteur)
permet de séparer une chaîne de caractère avec un caractère delimiteur donné

7.4.1 Detailed Description

Si le joueur est un humain.

Author

Groupe A7

Version

1.0

Date

avril 2022

7.4.2 Member Function Documentation

7.4.2.1 split()

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

permet de séparer une chaîne de caractère avec un caractère delimiteur donné

Parameters

<i>chaine</i>	string : la chaîne à découper
<i>char</i>	delimiteur caractère précisant où séparer la chaîne

Returns

retourne un vecteur

7.4.2.2 verifierEntree()

```
bool Humain::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)

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

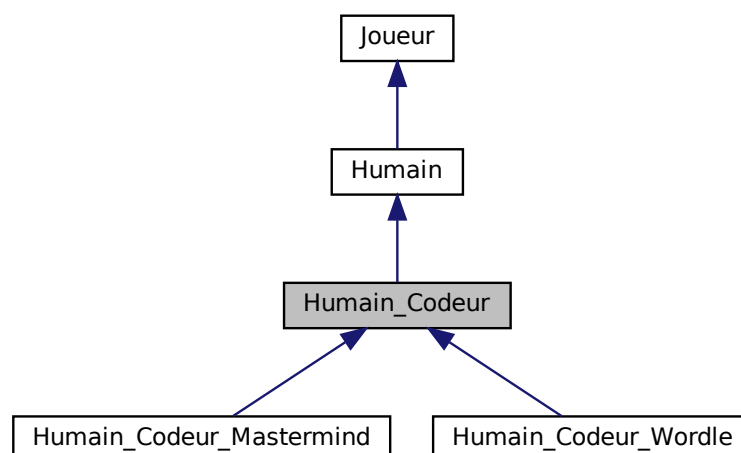
- /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain.hpp

7.5 Humain_Codeur Class Reference

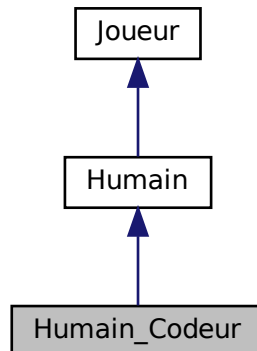
Cette classe abstraite permet de décrire les joueurs humains en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#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](#) ()
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

7.5.1 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

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Humain_Codeur()

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

Constructeur de la classe [Humain_Codeur](#).

7.5.3 Member Function Documentation

7.5.3.1 entrerCode()

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

methode abstraite utilisée pour demander un code au joueur codeur. Cette méthode devra être redefini pour les différents modes de jeu

Reimplemented in [Humain_Codeur_Wordle](#).

7.5.3.2 entrerMdP()

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

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

Returns

retourne l'entier entré par l'utilisateur

7.5.3.3 getMdP()

```
int 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 file:

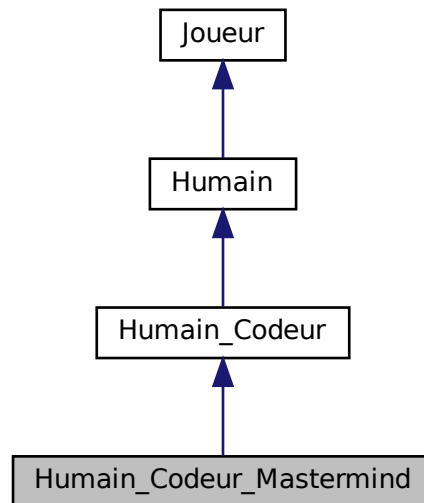
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur.hpp](#)

7.6 Humain_Codeur_Mastermind Class Reference

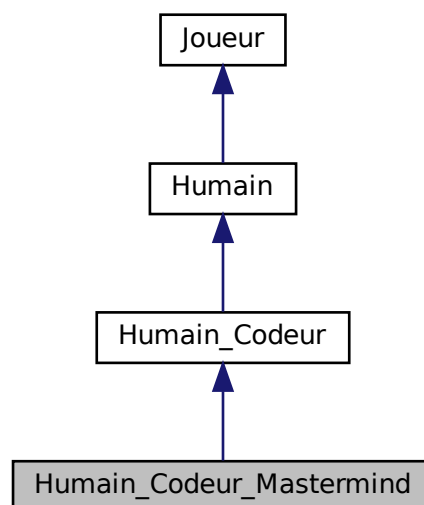
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Mastermind](#).

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

Inheritance diagram for Humain_Codeur_Mastermind:



Collaboration diagram for Humain_Codeur_Mastermind:



Additional Inherited Members

7.6.1 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

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

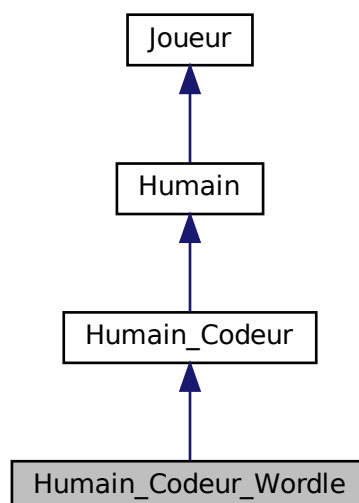
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp](#)

7.7 Humain_Codeur_Wordle Class Reference

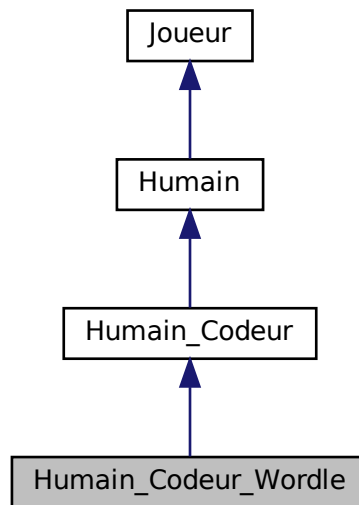
Cette classe est utilisée pour représenter un joueur humain en mode codeur dans le mode de jeu [Wordle](#).

```
#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](#).
- [Combinaison entrerCode](#) ()
méthode permettant à l'utilisateur de saisir le code que le joueur décodeur devra ensuite deviner
- 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](#).

7.7.1 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

7.7.2 Constructor & Destructor Documentation

7.7.2.1 Humain_Codeur_Wordle()

```
Humain_Codeur_Wordle::Humain_Codeur_Wordle ( )
```

Constructeur de la classe [Humain_Codeur_Wordle](#).

7.7.3 Member Function Documentation

7.7.3.1 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

Reimplemented from [Humain_Codeur](#).

7.7.3.2 VerifierEntree()

```
bool Humain_Codeur_Wordle::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](#).

Returns

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

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

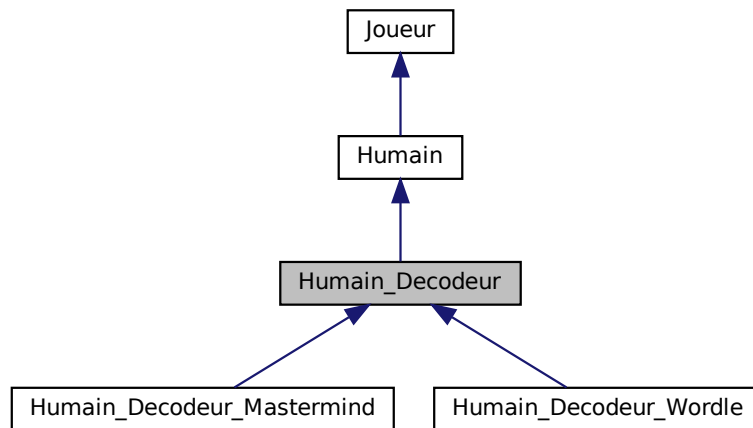
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Wordle.hpp](#)

7.8 Humain_Decodeur Class Reference

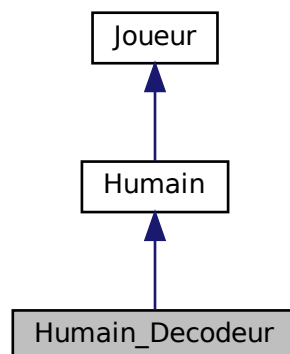
Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#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](#) ()
methode abstraite à redefinir pour les différents modes de jeu

7.8.1 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

7.8.2 Constructor & Destructor Documentation

7.8.2.1 Humain_Decodeur()

```
Humain_Decodeur::Humain_Decodeur ( )
```

Constructeur de la classe [Humain_Decodeur](#).

7.8.3 Member Function Documentation

7.8.3.1 entrerCombinaison()

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

methode abstraite à redefinir pour les différents modes de jeu

Reimplemented in [Humain_Decodeur_Mastermind](#), and [Humain_Decodeur_Wordle](#).

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

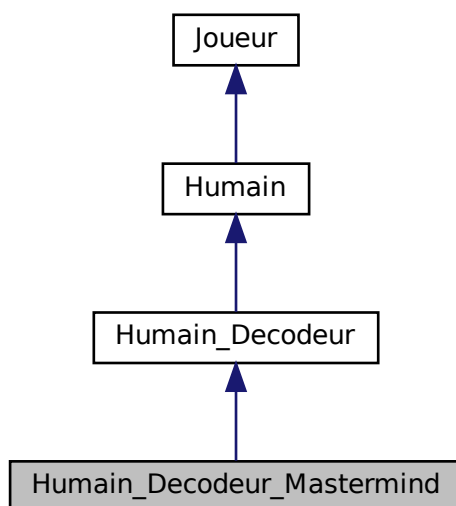
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur.hpp](#)

7.9 Humain_Decodeur_Mastermind Class Reference

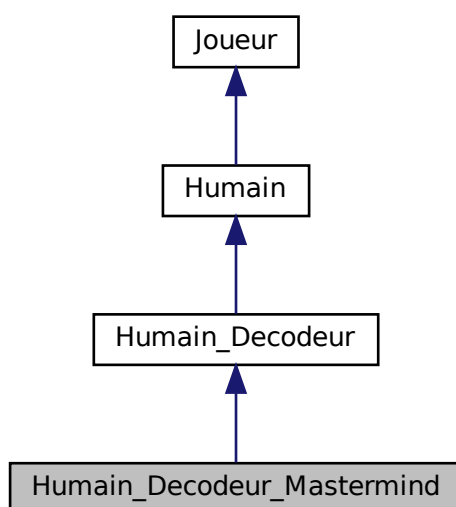
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Mastermind](#).

```
#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](#).
- [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)
Méthode qui permet de vérifier si l'entrée du joueur decodeur peut être interpréter en une combinaison du [Mastermind](#).

7.9.1 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

7.9.2 Constructor & Destructor Documentation

7.9.2.1 Humain_Decodeur_Mastermind()

```
Humain_Decodeur_Mastermind::Humain_Decodeur_Mastermind ( )
```

Constructeur de la classe [Humain_Decodeur_Mastermind](#).

7.9.3 Member Function Documentation

7.9.3.1 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

Reimplemented from [Humain_Decodeur](#).

7.9.3.2 VerifierEntree()

`bool Humain_Decodeur_Mastermind::VerifierEntree (`
 `string entree)`

Méthode qui permet de vérifier si l'entrée du joueur decodeur peut être interpréter en une combinaison du [Mastermind](#).

Returns

retourne un booleen, true si la chaîne peut être considérée comme une combinaison et false sinon

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

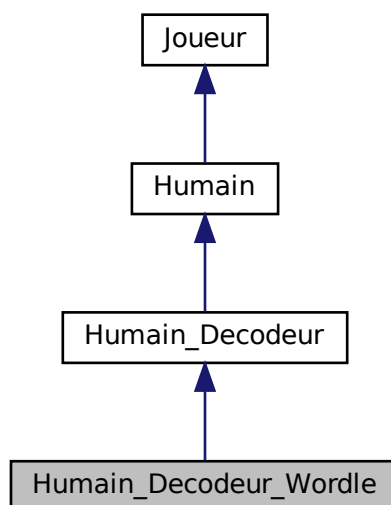
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp](#)

7.10 Humain_Decodeur_Wordle Class Reference

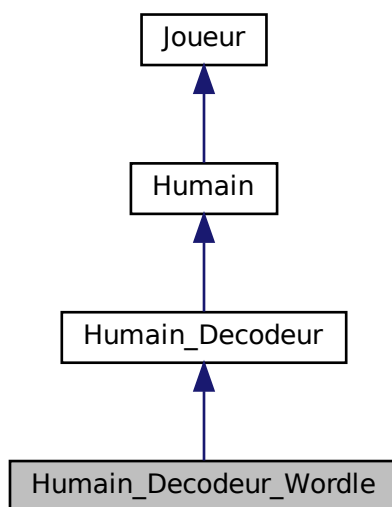
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Wordle](#).

```
#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_Decodeur_Wordle](#).
- [Combinaison entrerCombinaison](#) ()
méthode permettant à l'utilisateur de saisir une combinaison qu'il propose pour deviner le mot donnée par le joueur codeur
- bool [VerifierEntree](#) (string entree)
Méthode qui Permet de vérifier si l'entrée du joueur decodeur peut être interpréter en un mot de [Wordle](#).

7.10.1 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

7.10.2 Constructor & Destructor Documentation

7.10.2.1 Humain_Decodeur_Wordle()

```
Humain_Decodeur_Wordle::Humain_Decodeur_Wordle ( )
```

Constructeur de la classe [Humain_Decodeur_Wordle](#).

7.10.3 Member Function Documentation

7.10.3.1 entrerCombinaison()

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

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

Returns

retourne la combinaison saisi par le joueur decodeur

Reimplemented from [Humain_Decodeur](#).

7.10.3.2 VerifierEntree()

```
bool Humain_Decodeur_Wordle::VerifierEntree (
    string entree )
```

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

Returns

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

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

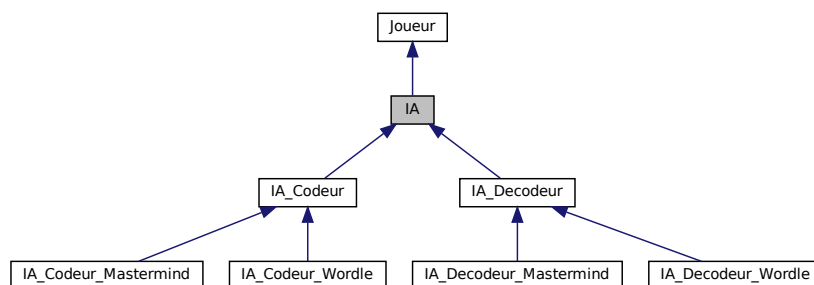
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp](#)

7.11 IA Class Reference

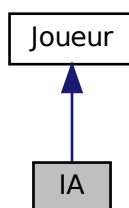
Si le joueur est un [IA](#).

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

Inheritance diagram for IA:



Collaboration diagram for IA:



Public Member Functions

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

7.11.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

7.11.2 Member Function Documentation

7.11.2.1 choisirCombinaison()

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

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

Returns

une combinaison

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

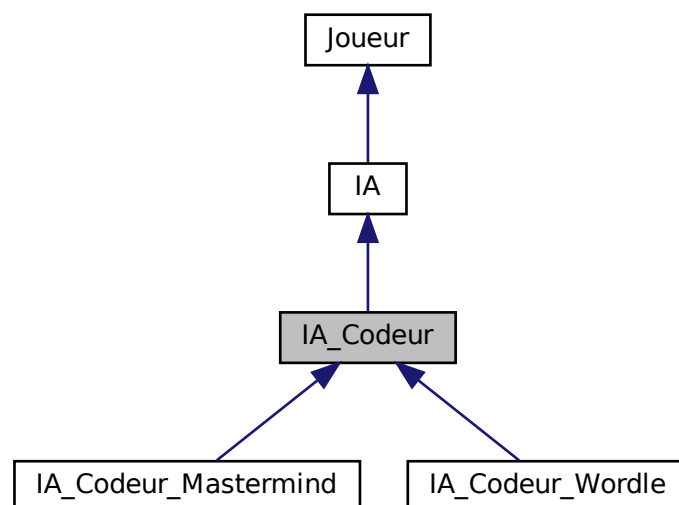
- /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA.hpp

7.12 IA_Codeur Class Reference

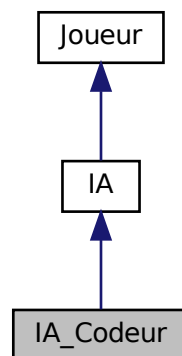
Cette classe abstraite permet de décrire l'intelligence artificielle en mode codeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#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](#) ()
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

7.12.1 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

7.12.2 Constructor & Destructor Documentation

7.12.2.1 IA_Codeur()

```
IA_Codeur::IA_Codeur ( )
```

Constructeur de la classe [IA_Codeur](#).

7.12.3 Member Function Documentation

7.12.3.1 genererCode()

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

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

Returns

retourne une combinaison

Reimplemented in [IA_Codeur_Wordle](#), and [IA_Codeur_Mastermind](#).

7.12.3.2 genererMdP()

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

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

Returns

retourne l'entier généré

7.12.3.3 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 file:

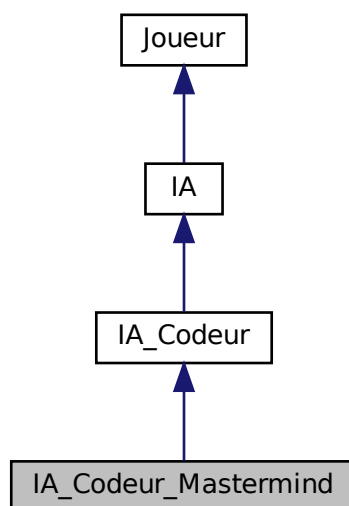
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur.hpp](#)

7.13 IA_Codeur_Mastermind Class Reference

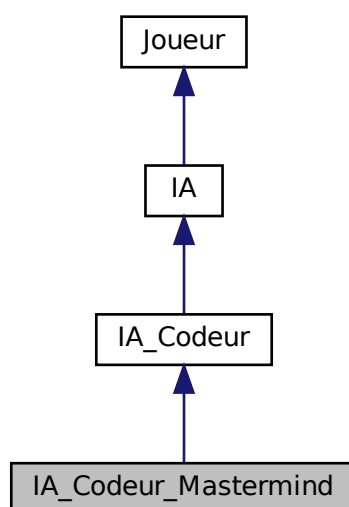
Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode codeur dans le mode de jeu [Mastermind](#).

```
#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](#).
- [Combinaison genererCode](#) ()
méthode permettant à la machine de générer un code que l'intelligence artificielle devra ensuite trouver

7.13.1 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

7.13.2 Constructor & Destructor Documentation

7.13.2.1 [IA_Codeur_Mastermind](#)()

```
IA_Codeur_Mastermind::IA_Codeur_Mastermind ( )
```

Constructeur de la classe [IA_Codeur_Mastermind](#).

7.13.3 Member Function Documentation

7.13.3.1 [genererCode](#)()

```
Combinaison 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

Reimplemented from [IA_Codeur](#).

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

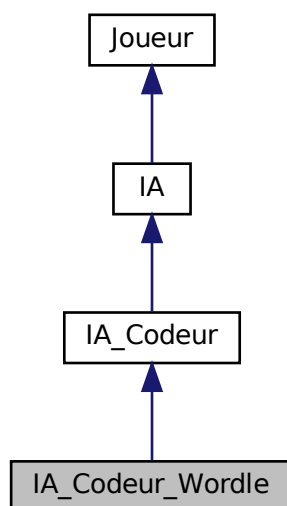
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.hpp](#)

7.14 IA_Codeur_Wordle Class Reference

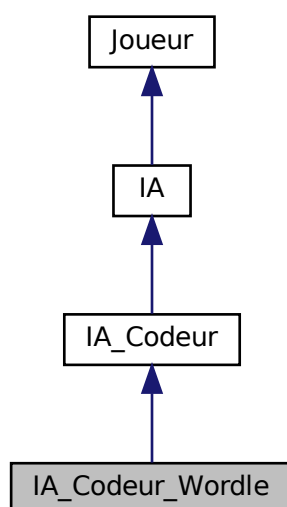
Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode codeur dans le mode de jeu [Wordle](#).

```
#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](#).
- [Combinaison genererCode](#) ()
méthode permettant à la machine de générer un code que le joueur artificiel décodeur devra ensuite trouver

7.14.1 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

7.14.2 Constructor & Destructor Documentation

7.14.2.1 [IA_Codeur_Wordle](#)()

```
IA_Codeur_Wordle::IA_Codeur_Wordle ( )
```

Constructeur de la classe [IA_Codeur_Wordle](#).

7.14.3 Member Function Documentation

7.14.3.1 [genererCode](#)()

```
Combinaison 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

Reimplemented from [IA_Codeur](#).

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

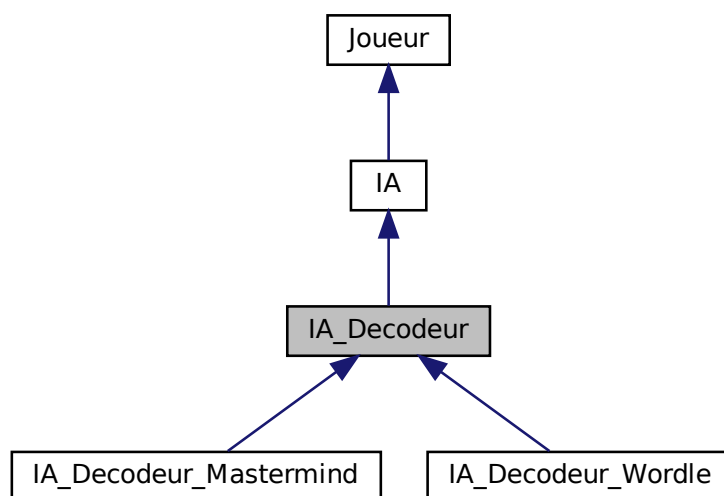
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Wordle.hpp](#)

7.15 IA_Decodeur Class Reference

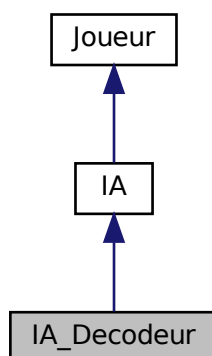
Cette classe abstraite permet de décrire l'intelligence artificielle en mode Decodeur peu importe le mode de jeu ([Mastermind](#) ou wordle).

```
#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](#) ()
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](#) ()
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](#) [Choisir_combi](#) ()
Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

7.15.1 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

7.15.2 Constructor & Destructor Documentation

7.15.2.1 [IA_Decodeur](#)()

```
IA_Decodeur::IA_Decodeur ( )
```

Constructeur de la classe [IA_Decodeur](#).

7.15.3 Member Function Documentation

7.15.3.1 Choisir_combi()

```
Combinaison IA_Decodeur::Choisir_combi ( ) [virtual]
```

Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

Returns

retourne une combinaison

Reimplemented in [IA_Decodeur_Wordle](#), and [IA_Decodeur_Mastermind](#).

7.15.3.2 Combi_possible()

```
vector< Combinaison > IA_Decodeur::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

Reimplemented in [IA_Decodeur_Wordle](#), and [IA_Decodeur_Mastermind](#).

7.15.3.3 Maj_ensemble()

```
void IA_Decodeur::Maj_ensemble ( ) [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.

Reimplemented in [IA_Decodeur_Wordle](#), and [IA_Decodeur_Mastermind](#).

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

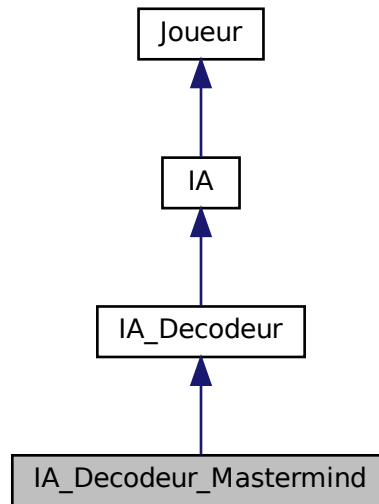
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur.hpp](#)

7.16 IA_Decodeur_Mastermind Class Reference

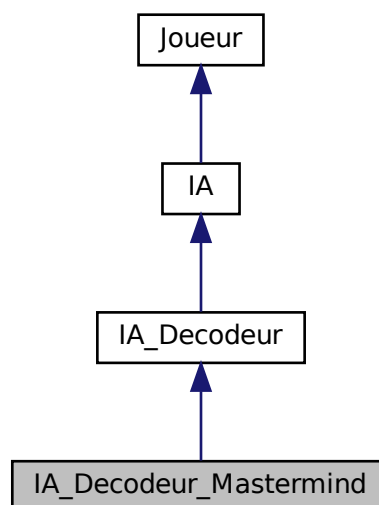
Cette classe est utilisée pour représenter un joueur artificiel (IA) en mode Decodeur dans le mode de jeu [Mastermind](#).

```
#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](#).
- `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](#).*
- `void Maj_ensemble ()`
*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 Choisir_combi ()`
Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

7.16.1 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

7.16.2 Constructor & Destructor Documentation

7.16.2.1 [IA_Decodeur_Mastermind](#)()

```
IA_Decodeur_Mastermind::IA_Decodeur_Mastermind ( )
```

Constructeur de la classe [IA_Decodeur_Mastermind](#).

7.16.3 Member Function Documentation

7.16.3.1 Choisir_combi()

```
Combinaison IA_Decodeur_Mastermind::Choisir_combi ( ) [virtual]
```

Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

Returns

retourne une combinaison

Reimplemented from [IA_Decodeur](#).

7.16.3.2 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

Reimplemented from [IA_Decodeur](#).

7.16.3.3 Maj_ensemble()

```
void IA_Decodeur_Mastermind::Maj_ensemble ( ) [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.

Reimplemented from [IA_Decodeur](#).

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

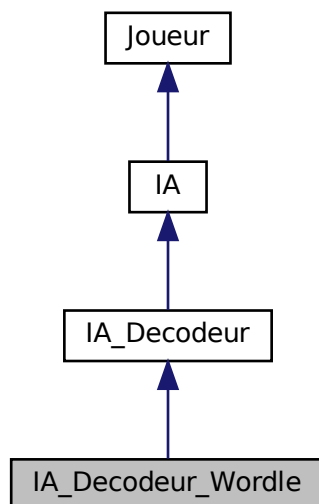
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Mastermind.hpp](#)

7.17 IA_Decodeur_Wordle Class Reference

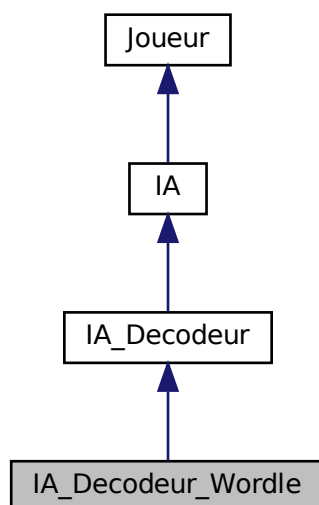
Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode Decodeur dans le mode de jeu [Wordle](#).

```
#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](#).
- `vector< Combinaison > Combi_possible ()`
Cette méthode abstraite permet de créer l'ensemble S de toutes les combinaisons de couleurs possibles.
- `void Maj_ensemble ()`
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 Choisir_combi](#) ()
Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

7.17.1 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

7.17.2 Constructor & Destructor Documentation

7.17.2.1 [IA_Decodeur_Wordle](#)()

```
IA_Decodeur_Wordle::IA_Decodeur_Wordle ( )
```

Constructeur de la classe [IA_Decodeur_Wordle](#).

7.17.3 Member Function Documentation

7.17.3.1 Choisir_combi()

```
Combinaison IA_Decodeur_Wordle::Choisir_combi ( ) [virtual]
```

Cette méthode permet de choisir la prochaine proposition soit dans l'ordre de son apparition dans l'ensemble, soit en choisissant une de manière aléatoire, soit en utilisant l'algorithme MinMax.

Returns

retourne une combinaison

Reimplemented from [IA_Decodeur](#).

7.17.3.2 Combi_possible()

```
vector< Combinaison > 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

Reimplemented from [IA_Decodeur](#).

7.17.3.3 Maj_ensemble()

```
void IA_Decodeur_Wordle::Maj_ensemble ( ) [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.

Reimplemented from [IA_Decodeur](#).

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

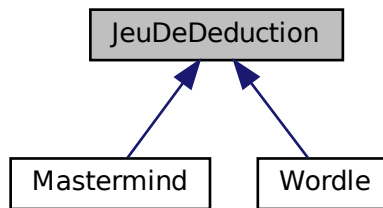
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp](#)

7.18 JeuDeDeduction Class Reference

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

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

Inheritance diagram for JeuDeDeduction:



Public Member Functions

- [JeuDeDeduction](#) ()
Constructeur neutre de la classe [JeuDeDeduction](#).
- [JeuDeDeduction](#) (const string chaine)
- int [main](#) ()
méthode principale, elle permet de lancement du jeu
- virtual void [partie](#) ()
Cette méthode abstraite permet de lancer une partie une fois que les paramètres et le mode de jeu ont bien été choisi par le joueur s'il le souhaite.
- virtual void [afficherPartie](#) ()
Cette méthode abstraite 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 car nous sommes du point de vue décodeur.
- void [afficherCode](#) ()
Cette méthode permet au joueur codeur de regarder son code discrètement. Pour activer cette méthode il faudra rentrer un mot de passe. Si le mot de passe rentré est le même que l'attribut du codeur alors le code sera affiché quelques secondes.
- [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 laquelle le joueur est en train de jouer.
- void [incrTour](#) ()
Cette méthode permet d'incrémenter de 1 l'attribut numeroTour.

7.18.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

7.18.2 Constructor & Destructor Documentation

7.18.2.1 JeuDeDeduction() [1/2]

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

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

7.18.2.2 JeuDeDeduction() [2/2]

```
JeuDeDeduction::JeuDeDeduction (
    const string chaine )
```

7.18.3 Member Function Documentation

7.18.3.1 afficherCode()

```
void JeuDeDeduction::afficherCode ( )
```

Cette méthode permet au joueur codeur de regarder son code discrètement. Pour activer cette méthode il faudra rentrer un mot de passe. Si le mot de passe rentré est le même que l'attribut du codeur alors le code sera affiché quelques secondes.

7.18.3.2 afficherPartie()

```
void JeuDeDeduction::afficherPartie ( ) [virtual]
```

Cette méthode abstraite 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 car nous sommes du point de vue décodeur.

7.18.3.3 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

7.18.3.4 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

7.18.3.5 incrTour()

```
void JeuDeDeduction::incrTour ( )
```

Cette méthode permet d'incrémenter de 1 l'attribut numeroTour.

7.18.3.6 main()

```
int JeuDeDeduction::main ( )
```

méthode principale, elle permet de lancement du jeu

7.18.3.7 partie()

```
void JeuDeDeduction::partie ( ) [virtual]
```

Cette méthode abstraite permet de lancer une partie une fois que les paramètres et le mode de jeu ont bien été choisi par le joueur s'il le souhaite.

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

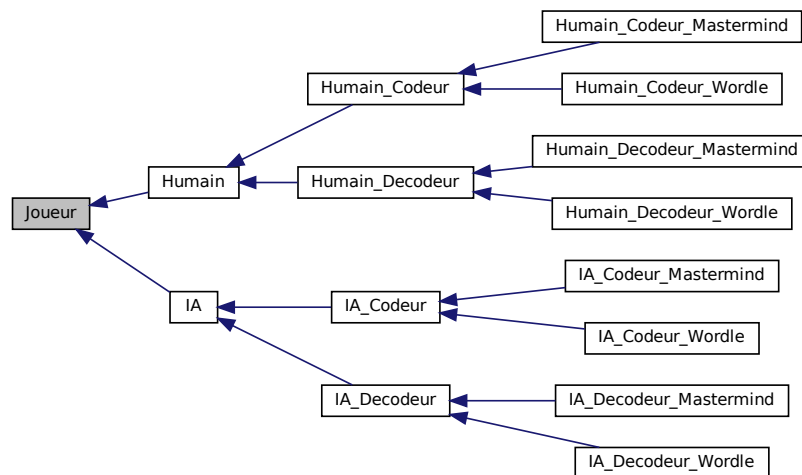
- /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp

7.19 Joueur Class Reference

classe abstraite regroupant tout joueur

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

Inheritance diagram for Joueur:



Public Member Functions

- [Joueur \(\)](#)
Constructeur de la classe joueur.
- [Joueur \(Combinaison comb, string nom\)](#)
- [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
- bool [verification \(const string chaine\)](#)
permet de verifier une chaine de caractère pour savoir si elle peut être transformée en combinaison

7.19.1 Detailed Description

classe abstraite regroupant tout joueur

Author

Groupe A7

Version

1.0

Date

avril 2022

7.19.2 Constructor & Destructor Documentation

7.19.2.1 Joueur() [1/2]

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

Constructeur de la classe joueur.

7.19.2.2 Joueur() [2/2]

```
Joueur::Joueur (
    Combinaison comb,
    string nom )
```

7.19.3 Member Function Documentation

7.19.3.1 entrerPseudo()

```
void Joueur::entrerPseudo ( )
```

Pour permettre au joueur de choisir un pseudo.

7.19.3.2 getCombinaison()

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

Pour accéder à l'attribut combinaison.

Returns

retourne la combinaison en attribut

7.19.3.3 getPseudo()

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

getteur de l'attribut Pseudo

Returns

retourne le pseudo en attribut

7.19.3.4 setCombinaison()

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

setteur de l'attribut combinaison

Parameters

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

7.19.3.5 setPseudo()

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

setteur de l'attribut Pseudo

Parameters

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

7.19.3.6 verification()

```
bool Joueur::verification (
    const string chaine )
```

permet de verifier une chaine de caractère pour savoir si elle peut être transformée en combinaison

Parameters

<i>chaine</i>	string : la chaine à vérifier
---------------	-------------------------------

Returns

un booleen

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

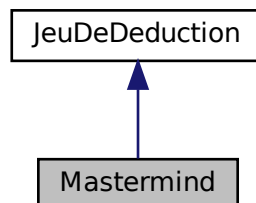
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp](#)
- [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.cpp](#)

7.20 Mastermind Class Reference

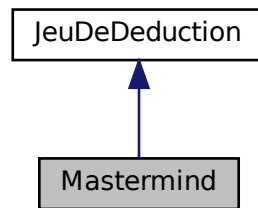
Classe principale permettant de lancement d'une partie. Elle comprend le main et.

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

Inheritance diagram for Mastermind:



Collaboration diagram for Mastermind:



Additional Inherited Members

7.20.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

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

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

7.21 Menu Class Reference

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

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

Public Member Functions

- [Menu](#) ()
Constructeur de la classe [Menu](#).
- void [afficherMenu](#) ()
getteur de l'attribut motDePasse
- void [choisirModeDeJeu](#) (int [ModeDeJeu](#))
- void [choisirDifficulteIA](#) ()
Fonction qui permet à l'utilisateur de choisir la difficulté de l'IA.
- void [parametreDeJeu](#) (int [NB_ELEMENT](#), int [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 int [NB_ELEMENT](#)
- static vector< string > [ENSEMBLE_ELEMENT](#)
- static int [NB_CASE](#)
- static int [NB_TOUR](#)
- static int [ModeDeJeu](#)

7.21.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

7.21.2 Constructor & Destructor Documentation

7.21.2.1 [Menu](#)()

```
Menu::Menu ( )
```

Constructeur de la classe [Menu](#).

7.21.3 Member Function Documentation

7.21.3.1 afficherMenu()

```
Menu::afficherMenu ( )
```

getteur de l'attribut motDePasse

7.21.3.2 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
------------	--

7.21.3.3 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
----------------	---

7.21.3.4 choisirDifficulteIA()

```
Menu::choisirDifficulteIA ( )
```

Fonction qui permet à l'utilisateur de choisir la difficulté de l'[IA](#).

7.21.3.5 choisirModeDeJeu()

```
void Menu::choisirModeDeJeu (
    int ModeDeJeu )
```

7.21.3.6 parametreDeJeu()

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

7.21.4 Member Data Documentation

7.21.4.1 ENSEMBLE_ELEMENT

```
vector<string> Menu::ENSEMBLE_ELEMENT [static]
```

7.21.4.2 ModeDeJeu

```
int Menu::ModeDeJeu [static]
```

7.21.4.3 NB_CASE

```
int Menu::NB_CASE [static]
```


7.21.4.4 NB_ELEMENT

```
int Menu::NB_ELEMENT [static]
```

7.21.4.5 NB_TOUR

```
int Menu::NB_TOUR [static]
```

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

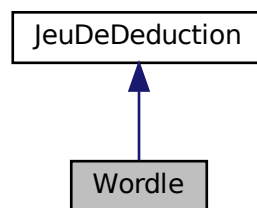
- /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp

7.22 Wordle Class Reference

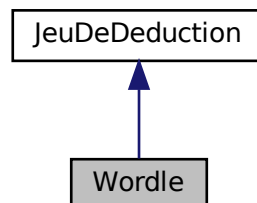
Classe principale permettant de lancement d'une partie de [Wordle](#).

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

Inheritance diagram for Wordle:



Collaboration diagram for Wordle:



Additional Inherited Members

7.22.1 Detailed Description

Classe principale permettant de lancement d'une partie de [Wordle](#).

Author

Groupe A7

Version

1.0

Date

avril 2022

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

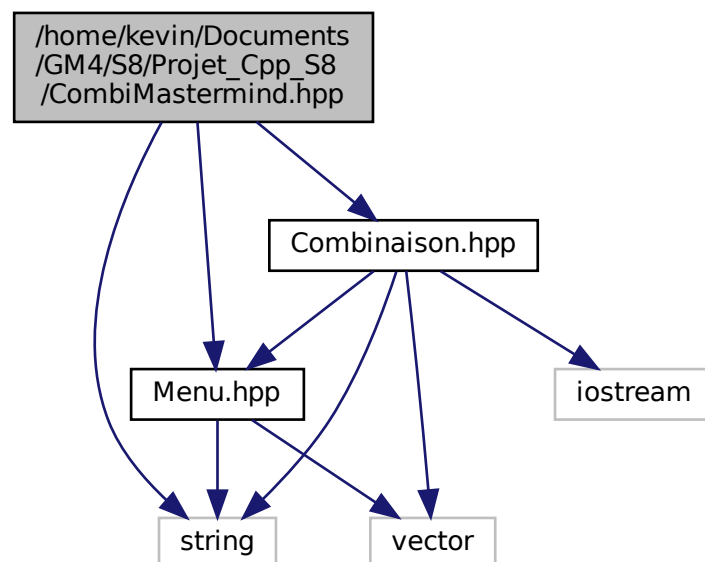
- /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.hpp

Chapter 8

File Documentation

8.1 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp File Reference

```
#include <string>
#include "Menu.hpp"
#include "Combinaison.hpp"
Include dependency graph for CombiMastermind.hpp:
```



Classes

- class [CombiMastermind](#)

Tableau d'entier représentant une combinaison de couleur.

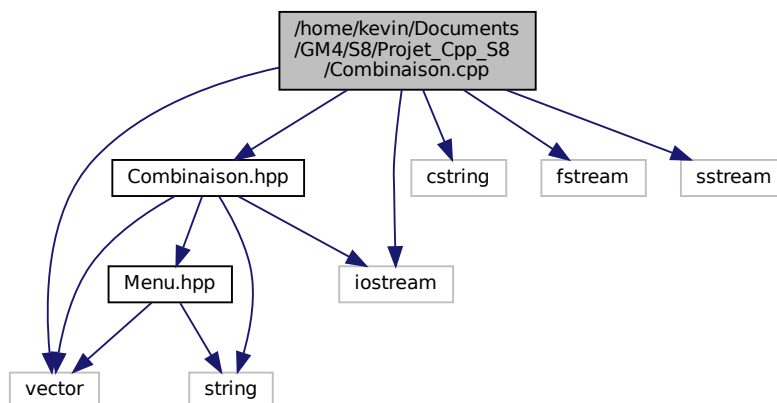
Namespaces

- [std](#)

8.2 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.cpp File Reference

```
#include <vector>
#include "Combinaison.hpp"
#include <cstring>
#include <iostream>
#include <fstream>
#include <sstream>
```

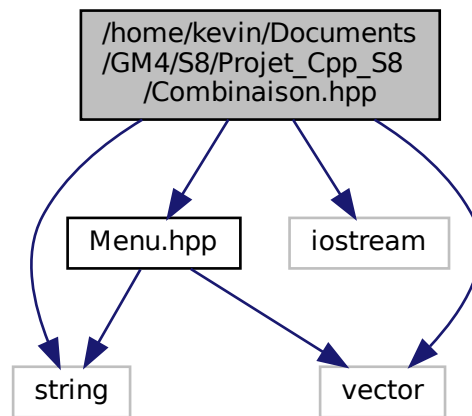
Include dependency graph for Combinaison.cpp:



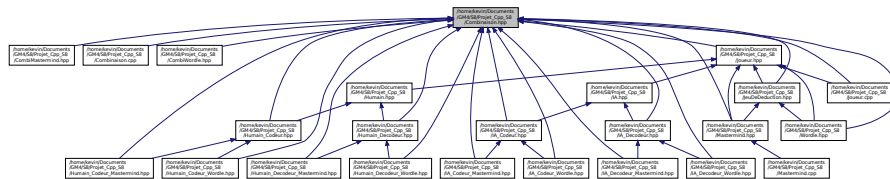
8.3 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include "Menu.hpp"
```

Include dependency graph for Combinaison.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Combinaison](#)

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

Namespaces

- `std`

8.4 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiWordle.hpp File Reference

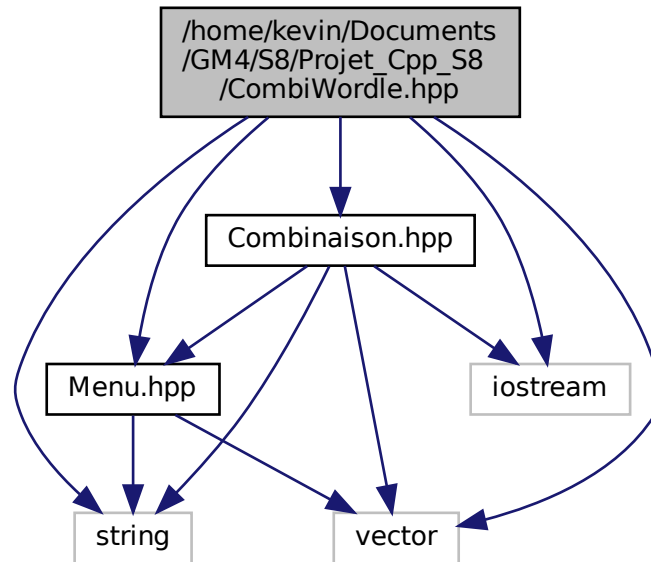
```

#include <string>
#include <vector>
#include <iostream>
#include "Menu.hpp"

```

```
#include "Combinaison.hpp"
```

Include dependency graph for CombiWordle.hpp:



Classes

- class [CombiWordle](#)

Tableau d'entier représentant une combinaison de lettre.

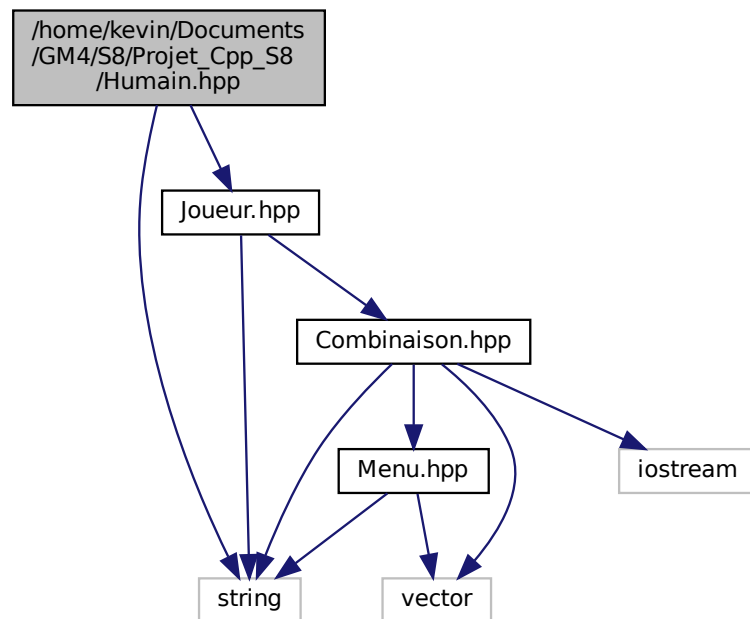
Namespaces

- [std](#)

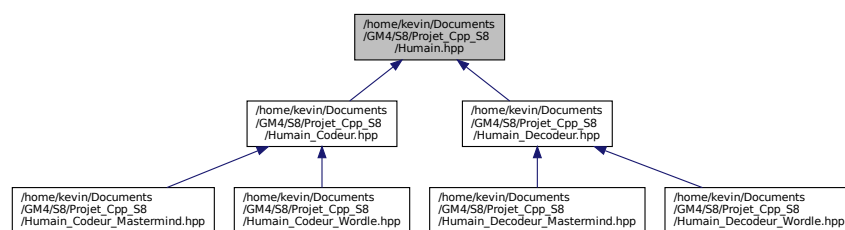
8.5 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain.hpp File Reference

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

Include dependency graph for Humain.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Humain](#)

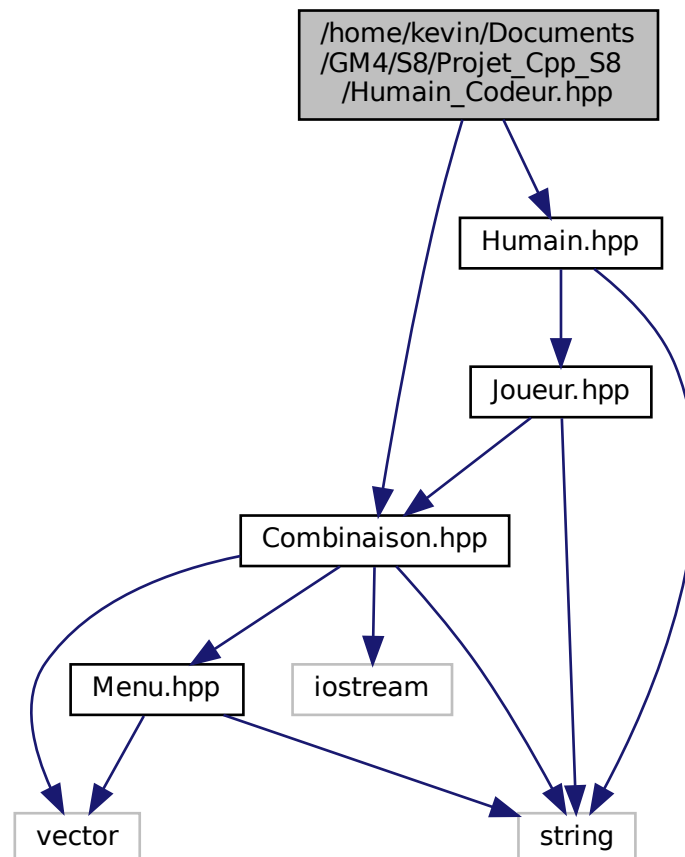
Si le joueur est un humain.

Namespaces

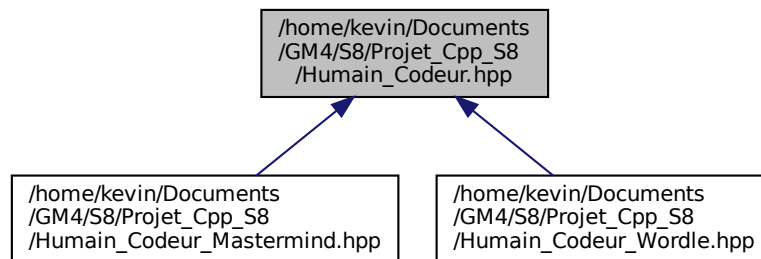
- [std](#)

8.6 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_↔ Codeur.hpp File Reference

```
#include "Humain.hpp"  
#include "Combinaison.hpp"  
Include dependency graph for Humain_Codeur.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Humain_Codeur](#)

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

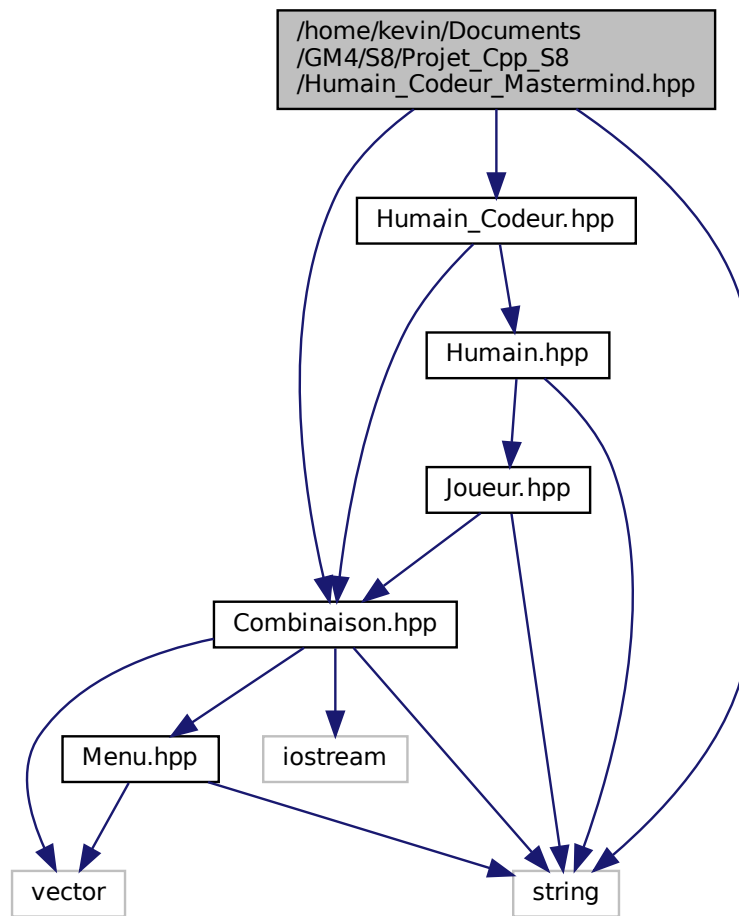
Namespaces

- [std](#)

8.7 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "Humain_Codeur.hpp"
```

Include dependency graph for Humain_Codeur_Mastermind.hpp:



Classes

- class [Humain_Codeur_Mastermind](#)

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

Namespaces

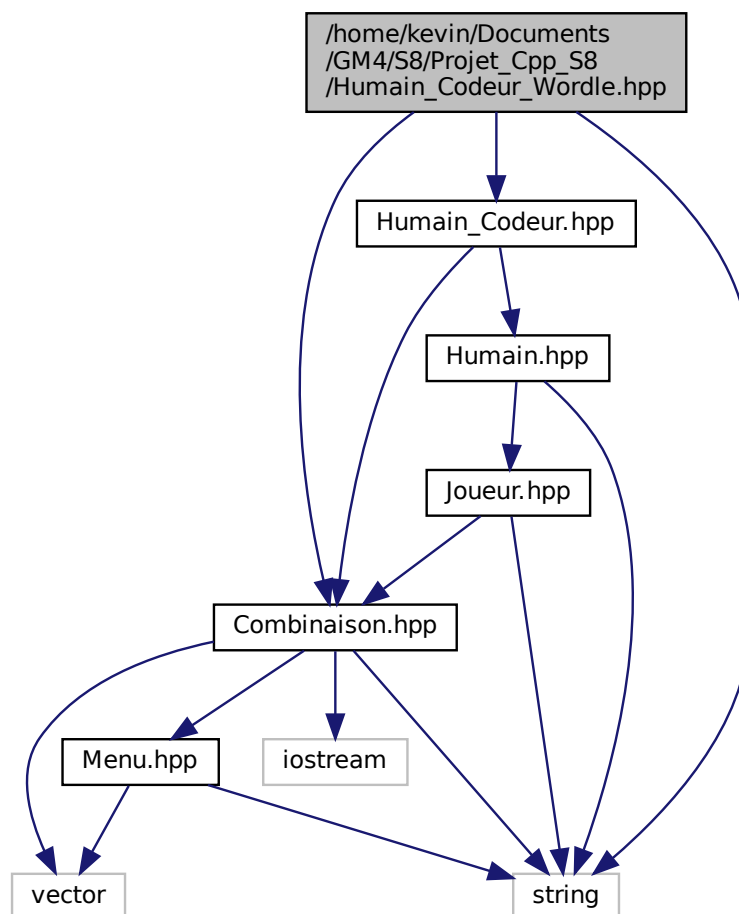
- [std](#)

8.8 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Wordle.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
```

```
#include "Humain_Codeur.hpp"
```

Include dependency graph for Humain_Codeur_Wordle.hpp:



Classes

- class [Humain_Codeur_Wordle](#)

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

Namespaces

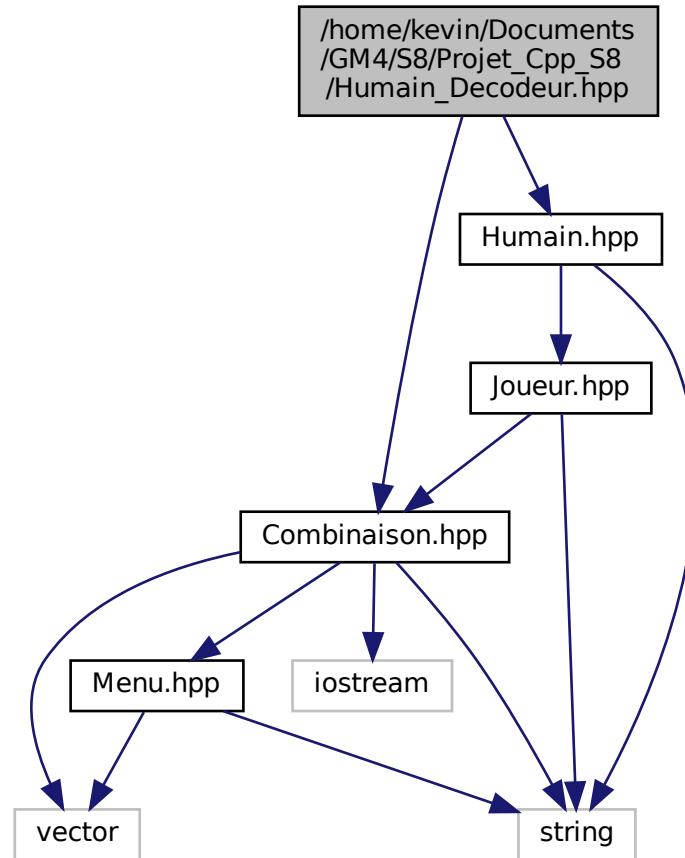
- [std](#)

8.9 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur.hpp File Reference

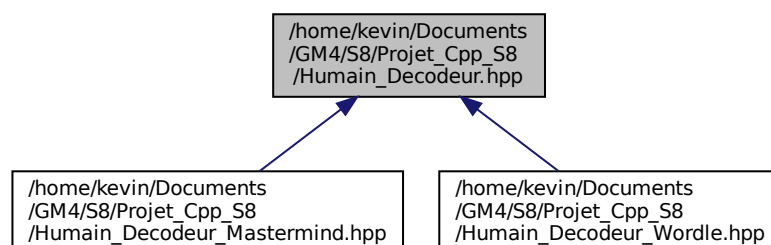
```
#include "Humain.hpp"
```

```
#include "Combinaison.hpp"
```

Include dependency graph for Humain_Decodeur.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Humain_Decodeur](#)

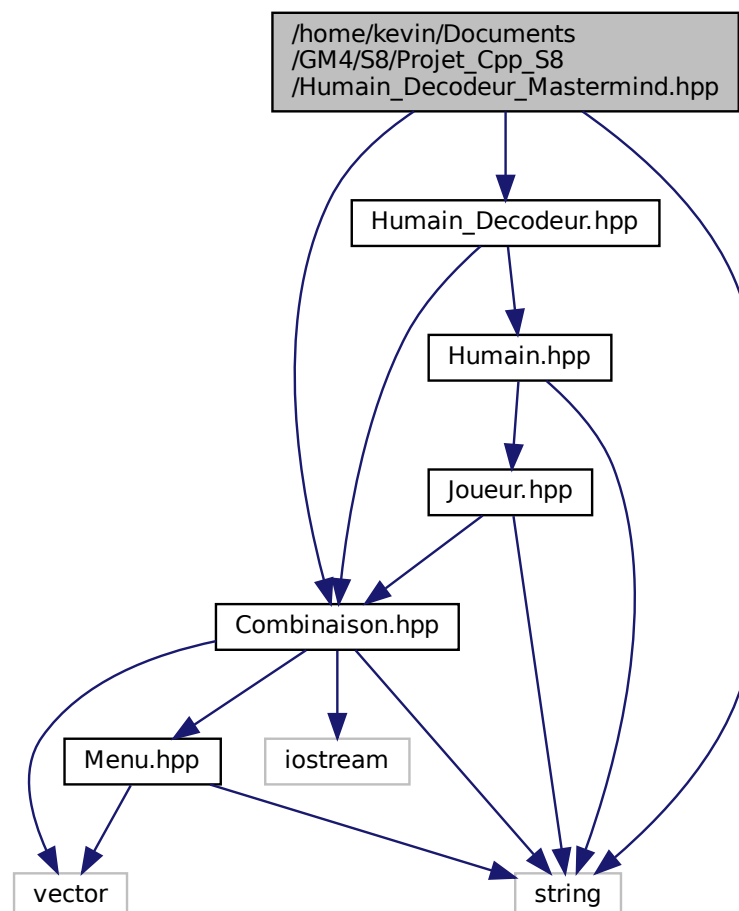
Cette classe abstraite permet de décrire les joueurs humains en mode decodeur peu importe le mode de jeu (*Mastermind* ou *wordle*).

Namespaces

- `std`

8.10 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "Humain_Decodeur.hpp"
Include dependency graph for Humain_Decodeur_Mastermind.hpp:
```



Classes

- class [Humain_Decodeur_Mastermind](#)

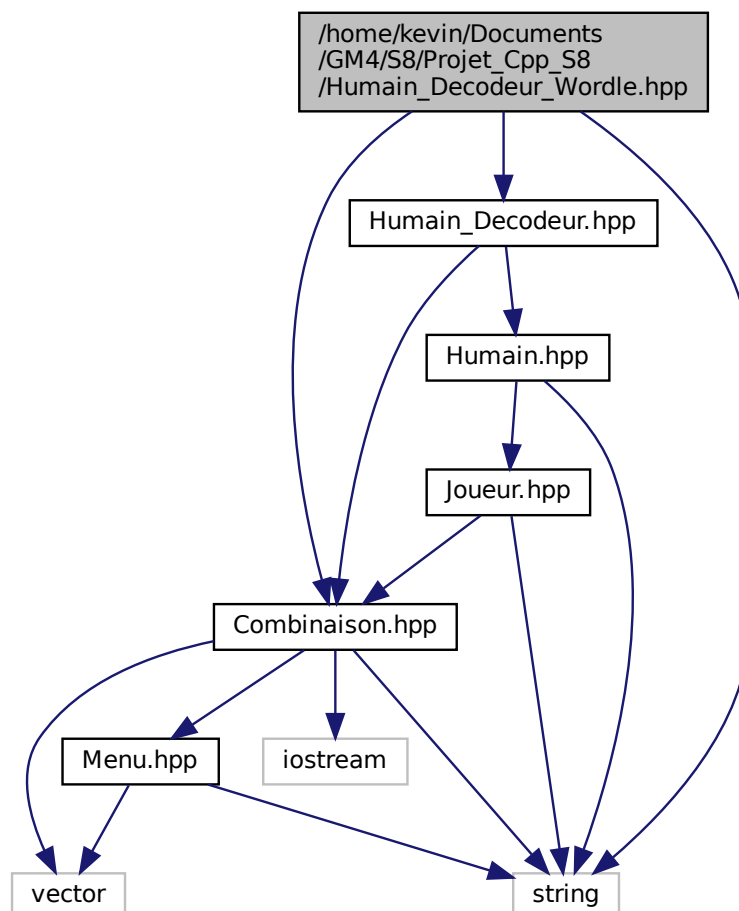
Cette classe est utilisée pour représenter un joueur humain en mode decodeur dans le mode de jeu [Mastermind](#).

Namespaces

- [std](#)

8.11 [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp](#) File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "Humain_Decodeur.hpp"
Include dependency graph for Humain_Decodeur_Wordle.hpp:
```



Classes

- class [Humain_Decodeur_Wordle](#)

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

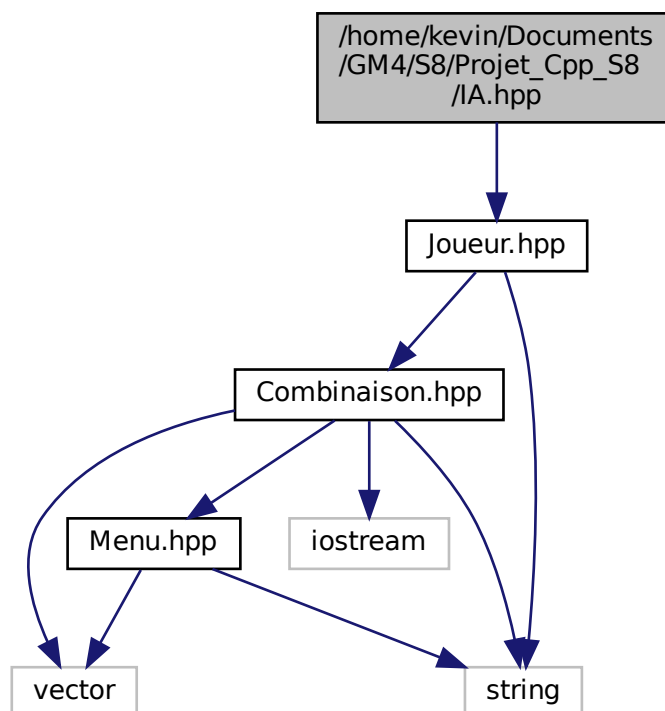
Namespaces

- [std](#)

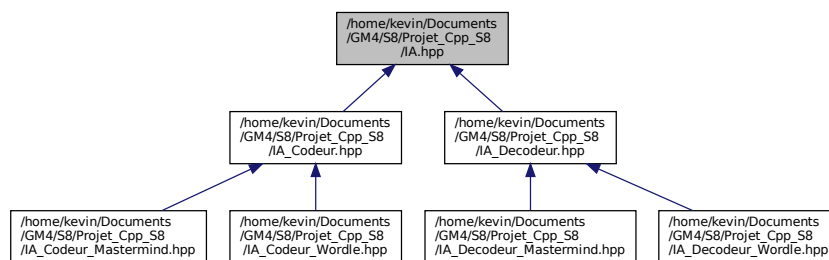
8.12 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA.hpp File Reference

```
#include "Joueur.hpp"
```

Include dependency graph for IA.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [IA](#)

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

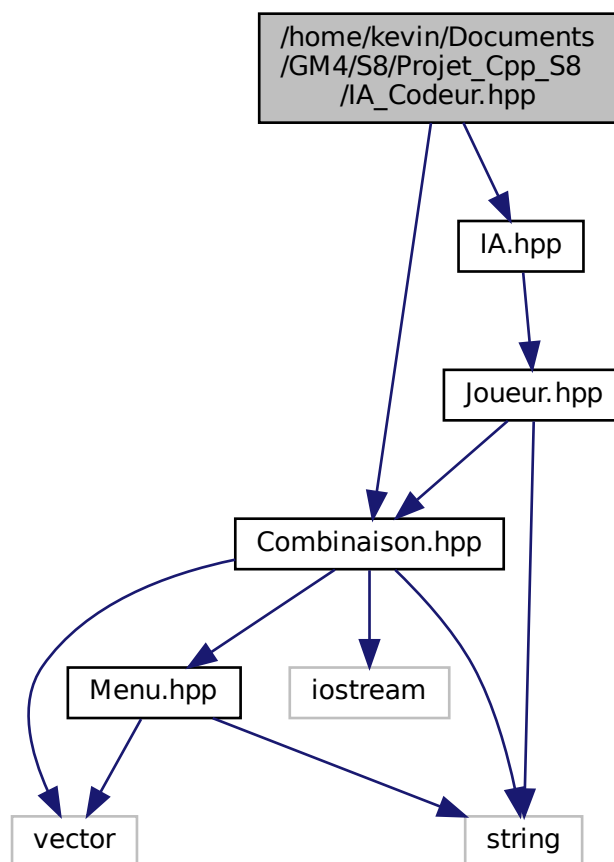
Namespaces

- [std](#)

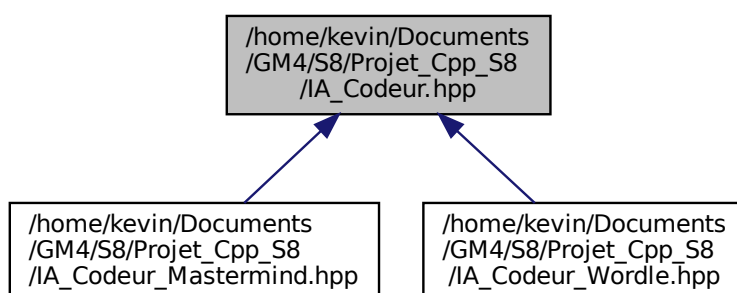
8.13 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur.hpp File Reference

```
#include "IA.hpp"
#include "Combinaison.hpp"
```


Include dependency graph for IA_Codeur.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [IA_Codeur](#)

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

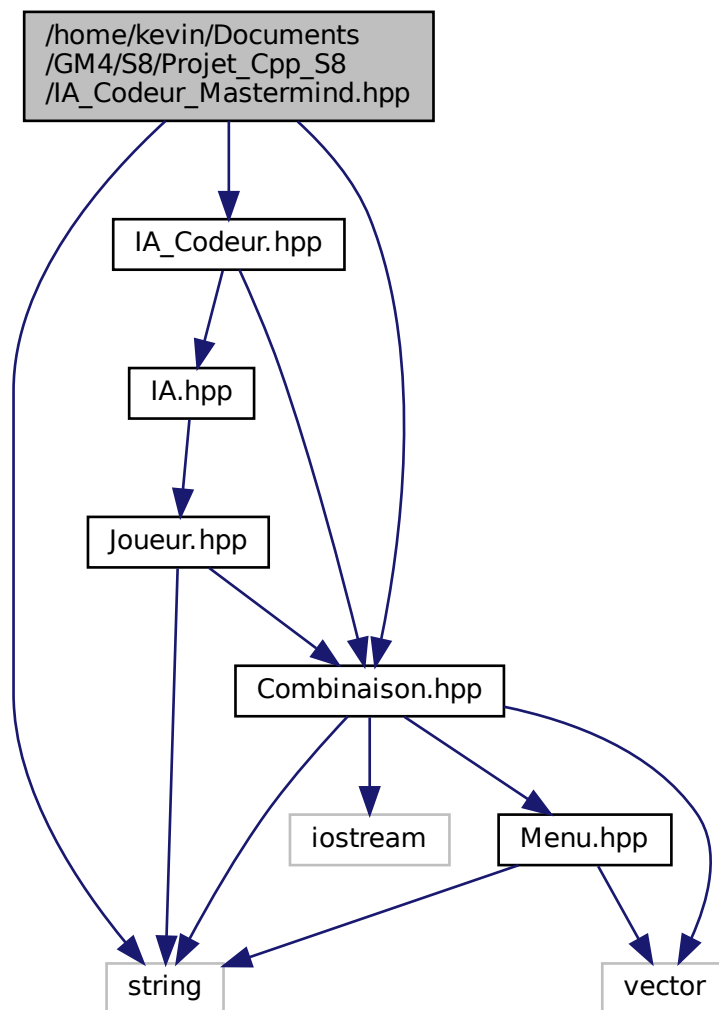
Namespaces

- [std](#)

8.14 [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_↔](#) Mastermind.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "IA_Codeur.hpp"
```

Include dependency graph for IA_Codeur_Mastermind.hpp:



Classes

- class [IA_Codeur_Mastermind](#)

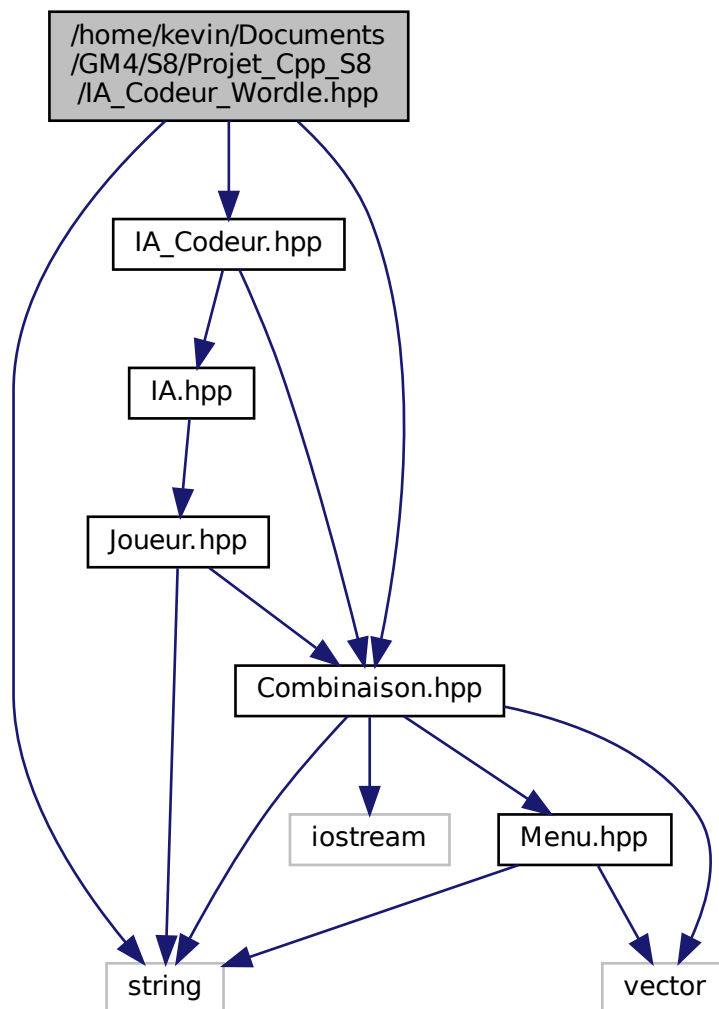
Cette classe est utilisée pour représenter un joueur artificiel (IA) en mode codeur dans le mode de jeu [Mastermind](#).

Namespaces

- [std](#)

8.15 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Wordle.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "IA_Codeur.hpp"
Include dependency graph for IA_Codeur_Wordle.hpp:
```



Classes

- class [IA_Codeur_Wordle](#)

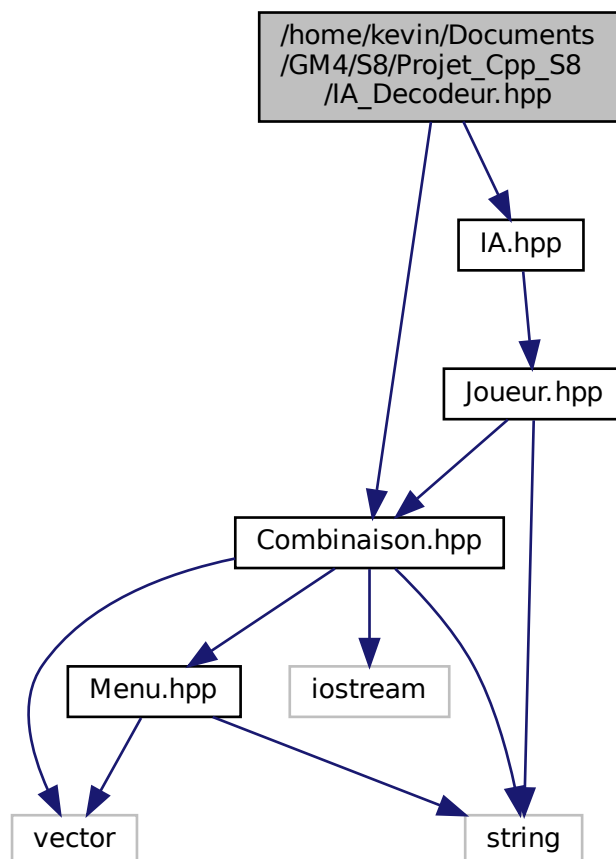
Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode codeur dans le mode de jeu [Wordle](#).

Namespaces

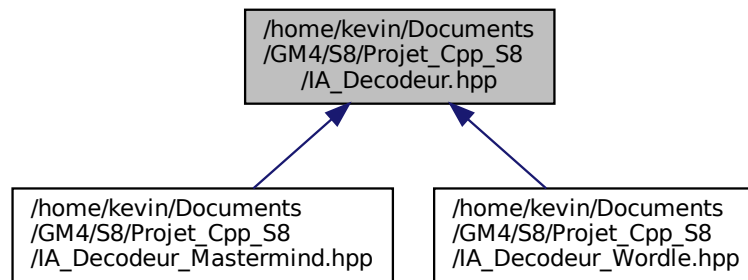
- [std](#)

8.16 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur.hpp File Reference

```
#include "IA.hpp"  
#include "Combinaison.hpp"  
Include dependency graph for IA_Decodeur.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [IA_Decodeur](#)

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

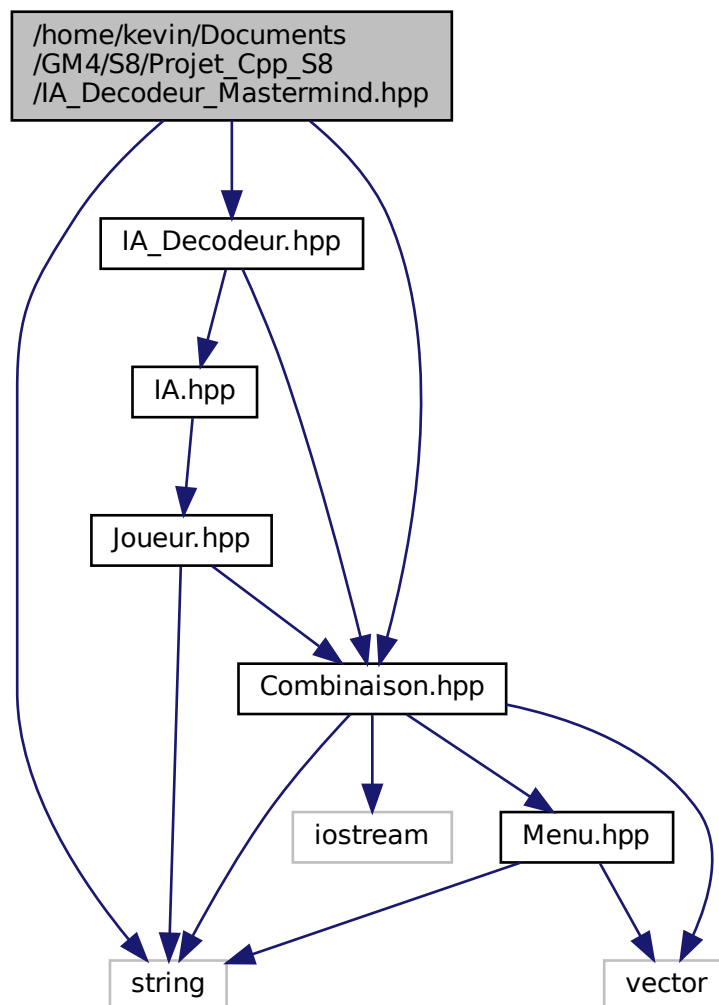
Namespaces

- [std](#)

8.17 [/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_](#) **Mastermind.hpp File** **Reference**

```
#include <string>
#include "Combinaison.hpp"
#include "IA_Decodeur.hpp"
```

Include dependency graph for IA_Decodeur_Mastermind.hpp:



Classes

- class [IA_Decodeur_Mastermind](#)

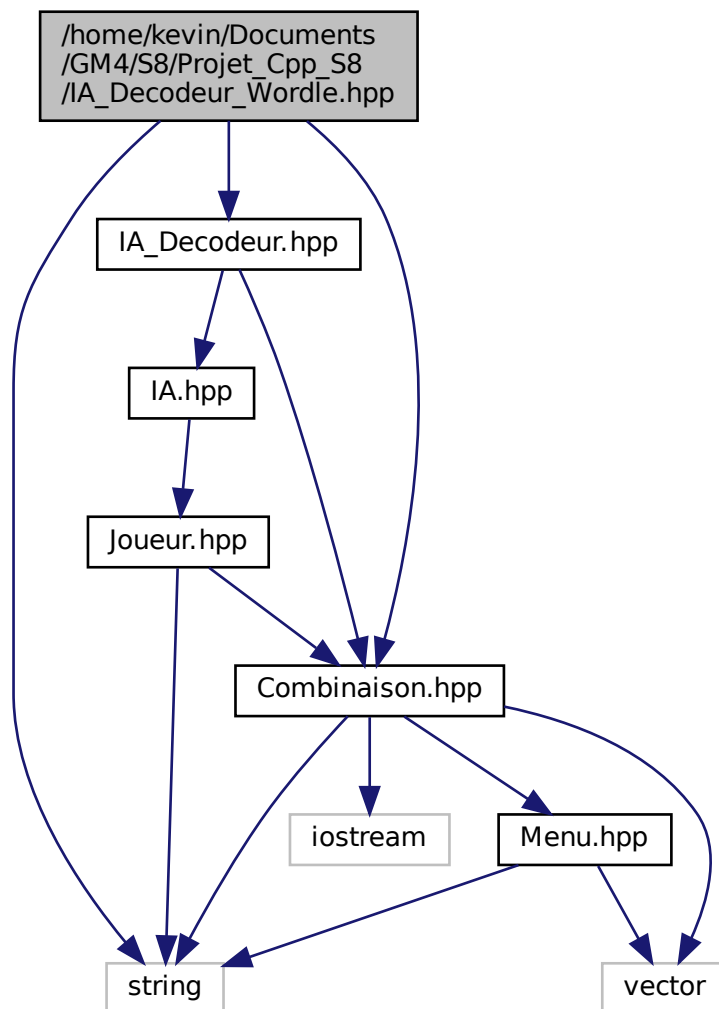
Cette classe est utilisée pour représenter un joueur artificiel ([IA](#)) en mode Decodeur dans le mode de jeu [Mastermind](#).

Namespaces

- [std](#)

8.18 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
#include "IA_Decodeur.hpp"
Include dependency graph for IA_Decodeur_Wordle.hpp:
```



Classes

- class [IA_Decodeur_Wordle](#)

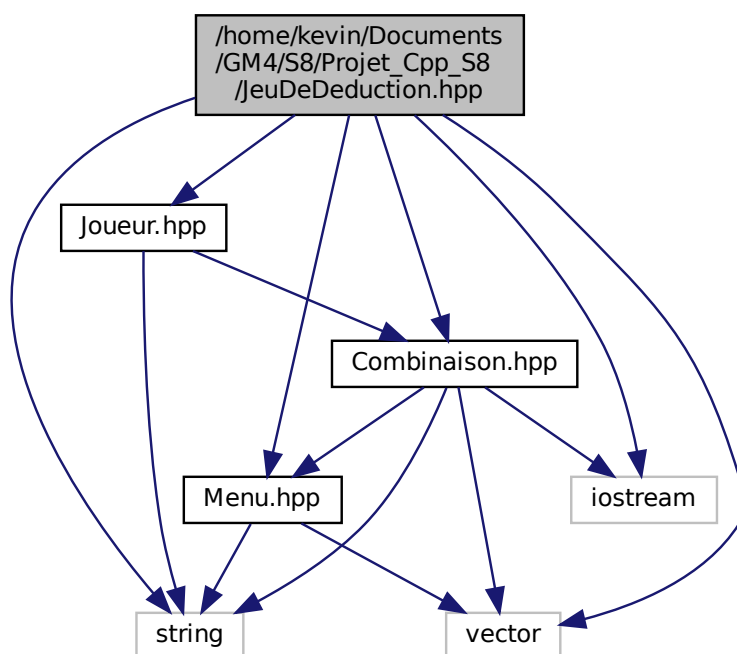
Cette classe est utilisée pour représenter un joueur virtuel (IA) en mode Decodeur dans le mode de jeu [Wordle](#).

Namespaces

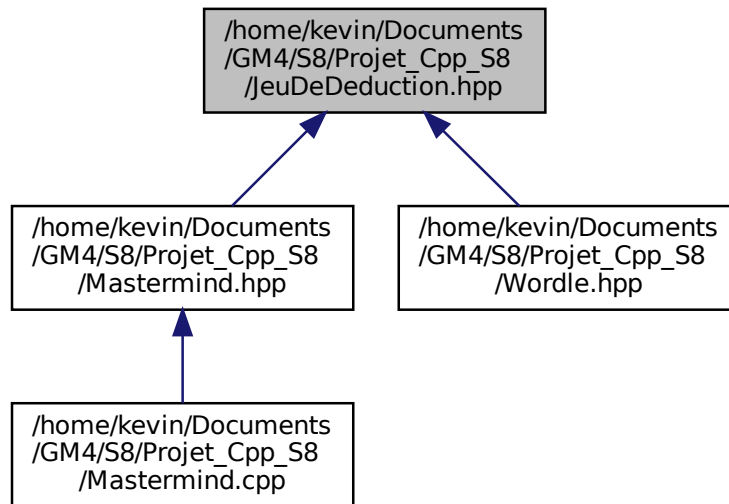
- [std](#)

8.19 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp File Reference

```
#include <string>
#include <iostream>
#include <vector>
#include "Menu.hpp"
#include "Joueur.hpp"
#include "Combinaison.hpp"
Include dependency graph for JeuDeDeduction.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [JeuDeDeduction](#)

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

Namespaces

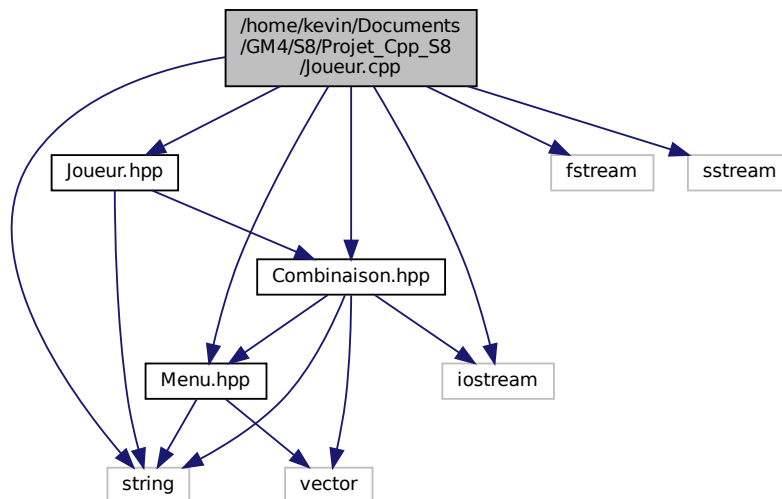
- [std](#)

8.20 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.cpp File Reference

```
#include "Joueur.hpp"
#include "Combinaison.hpp"
#include "Menu.hpp"
#include <iostream>
#include <string>
#include <fstream>
```

```
#include <sstream>
```

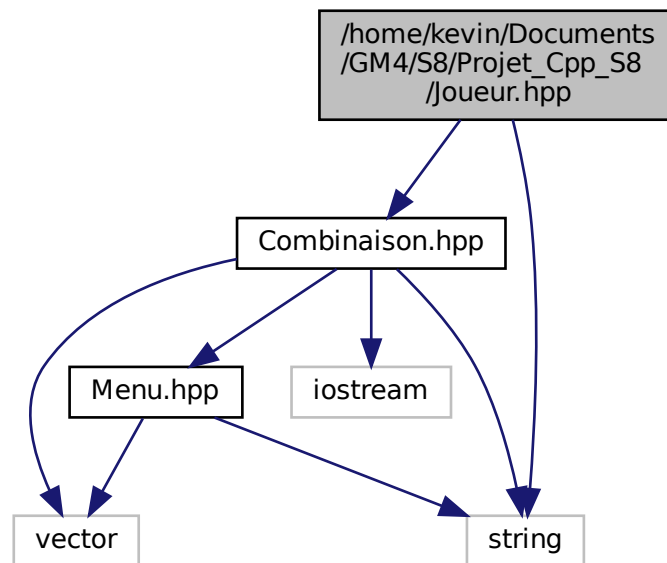
Include dependency graph for Joueur.cpp:



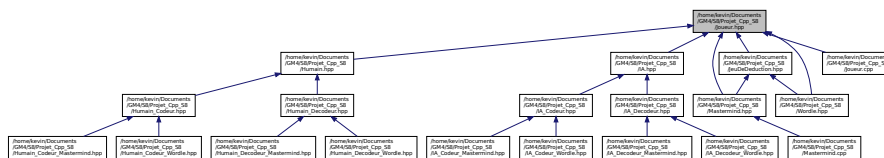
8.21 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp File Reference

```
#include <string>
#include "Combinaison.hpp"
```

Include dependency graph for `Joueur.hpp`:



This graph shows which files directly or indirectly include this file:



Classes

- class `Joueur`
classe abstraite regroupant tout joueur

Namespaces

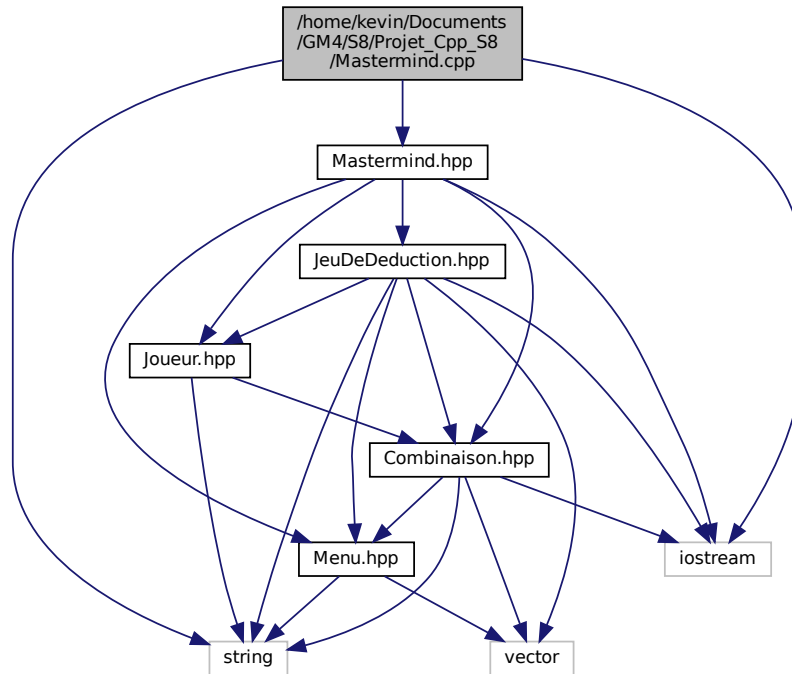
- `std`

8.22 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.cpp File Reference

```
#include <iostream>
#include <string>
```

```
#include "Mastermind.hpp"
```

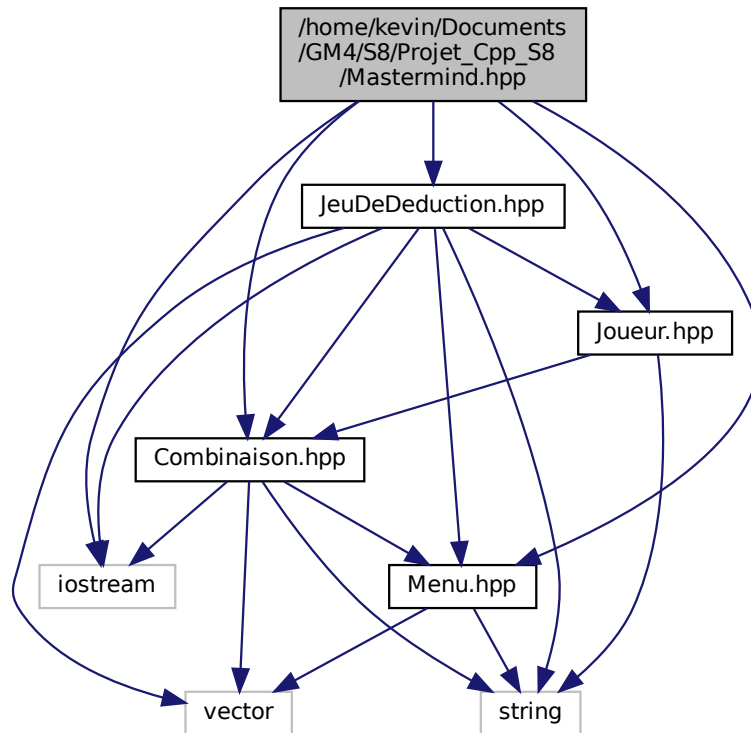
Include dependency graph for Mastermind.cpp:



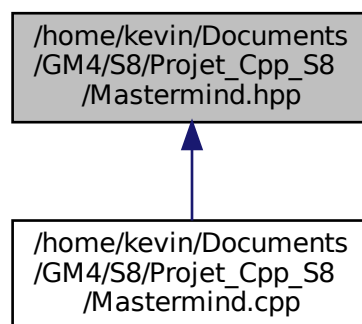
8.23 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.hpp File Reference

```
#include <iostream>
#include "Menu.hpp"
#include "Joueur.hpp"
#include "Combinaison.hpp"
#include "JeuDeDeduction.hpp"
```

Include dependency graph for Mastermind.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Mastermind](#)

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

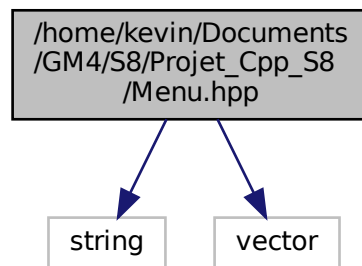
Namespaces

- [std](#)

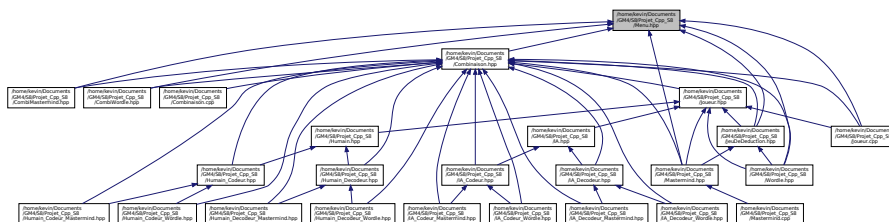
8.24 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.cpp File Reference

8.25 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp File Reference

```
#include <string>
#include <vector>
Include dependency graph for Menu.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Menu](#)

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

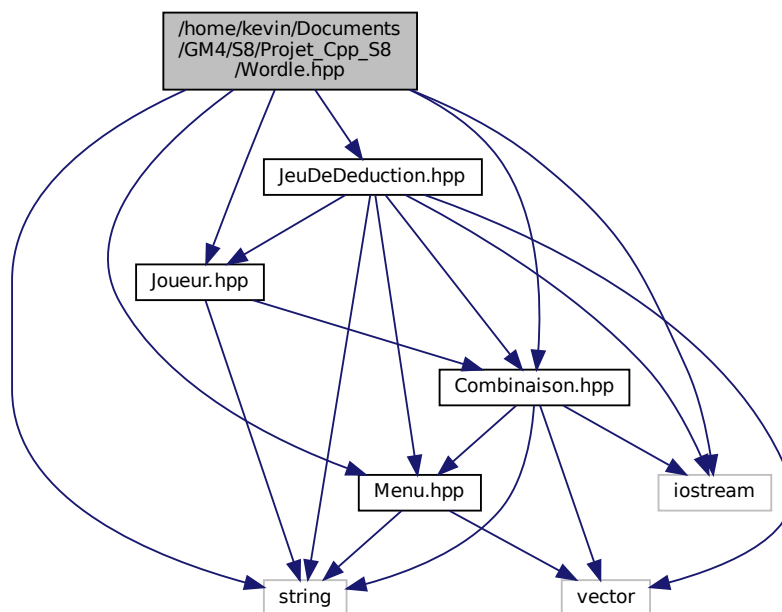
Namespaces

- [std](#)

8.26 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/README.md File Reference

8.27 /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Wordle.hpp File Reference

```
#include <string>
#include <iostream>
#include "Menu.hpp"
#include "Joueur.hpp"
#include "Combinaison.hpp"
#include "JeuDeDeduction.hpp"
Include dependency graph for Wordle.hpp:
```



Classes

- class [Wordle](#)

Classe principale permettant de lancement d'une partie de [Wordle](#).

Namespaces

- [std](#)

Index

/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiMastermind.hpp, 67
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/CombiWordle.hpp, 69
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.cpp, 68
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp, 68
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain.hpp, 70
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur.hpp, 72
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Mastermind.hpp, 73
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Codeur_Wordle.hpp, 74
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur.hpp, 75
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Mastermind.hpp, 77
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Humain_Decodeur_Wordle.hpp, 78
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA.hpp, 79
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur.hpp, 80
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Mastermind.hpp, 82
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Codeur_Wordle.hpp, 84
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur.hpp, 85
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Mastermind.hpp, 86
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/IA_Decodeur_Wordle.hpp, 88
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/JeuDeDeduction.hpp, 89
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.cpp, 90
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp, 91
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.cpp, 92
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.hpp, 93
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.cpp, 95
/home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp, 95

afficherCode, 55
JeuDeDeduction, 55
afficherMenu, 63
Menu, 63
afficherPartie, 55
JeuDeDeduction, 55
appartientDico, 63
Menu, 63
appartientFF, 63
Menu, 63
bienPlace, 14
CombiMastermind, 14
CombiWordle, 20
Choisir_combi, 46
IA_Decodeur, 49
IA_Decodeur_Mastermind, 52
IA_Decodeur_Wordle, 52
choisirCombinaison, 38
IA, 38
choisirDifficulteIA, 63
Menu, 63
choisirModeDeJeu, 63
Menu, 63
combi_possible, 47
IA_Decodeur, 50
IA_Decodeur_Mastermind, 53
IA_Decodeur_Wordle, 13
CombiMastermind, 14
bienPlace, 15
malPlace, 15
toString, 17
CombiMastermind, 17
CombiWordle, 18
bienPlace, 19, 20
CombiWordle, 19, 20
malPlace, 20
toString, 21

- detectionVictoire
 - JeuDeDeduction, 56
- ENSEMBLE_ELEMENT
 - Menu, 64
- entrerCode
 - Humain_Codeur, 25
 - Humain_Codeur_Wordle, 29
- entrerCombinaison
 - Humain_Decodeur, 31
 - Humain_Decodeur_Mastermind, 33
 - Humain_Decodeur_Wordle, 36
- entrerMdP
 - Humain_Codeur, 25
- entrerPseudo
 - Joueur, 58
- genererCode
 - IA_Codeur, 40
 - IA_Codeur_Mastermind, 42
 - IA_Codeur_Wordle, 44
- genererMdP
 - IA_Codeur, 40
- get
 - Combinaison, 17
- getCombinaison
 - Joueur, 58
- getMdP
 - Humain_Codeur, 25
 - IA_Codeur, 40
- getPseudo
 - Joueur, 59
- getTour
 - JeuDeDeduction, 56
- Humain, 21
 - split, 22
 - verifierEntree, 22
- Humain_Codeur, 23
 - entrerCode, 25
 - entrerMdP, 25
 - getMdP, 25
 - Humain_Codeur, 25
- Humain_Codeur_Mastermind, 26
- Humain_Codeur_Wordle, 27
 - entrerCode, 29
 - Humain_Codeur_Wordle, 29
 - VerifierEntree, 29
- Humain_Decodeur, 30
 - entrerCombinaison, 31
 - Humain_Decodeur, 31
- Humain_Decodeur_Mastermind, 32
 - entrerCombinaison, 33
 - Humain_Decodeur_Mastermind, 33
 - VerifierEntree, 34
- Humain_Decodeur_Wordle, 34
 - entrerCombinaison, 36
 - Humain_Decodeur_Wordle, 36
 - VerifierEntree, 36
- IA, 37
 - choisirCombinaison, 38
- IA_Codeur, 38
 - genererCode, 40
 - genererMdP, 40
 - getMdP, 40
 - IA_Codeur, 39
- IA_Codeur_Mastermind, 41
 - genererCode, 42
 - IA_Codeur_Mastermind, 42
- IA_Codeur_Wordle, 43
 - genererCode, 44
 - IA_Codeur_Wordle, 44
- IA_Decodeur, 45
 - Choisir_combi, 46
 - Combi_possible, 47
 - IA_Decodeur, 46
 - Maj_ensemble, 47
- IA_Decodeur_Mastermind, 48
 - Choisir_combi, 49
 - Combi_possible, 50
 - IA_Decodeur_Mastermind, 49
 - Maj_ensemble, 50
- IA_Decodeur_Wordle, 51
 - Choisir_combi, 52
 - Combi_possible, 53
 - IA_Decodeur_Wordle, 52
 - Maj_ensemble, 53
- incrTour
 - JeuDeDeduction, 56
- JeuDeDeduction, 54
 - afficherCode, 55
 - afficherPartie, 55
 - detectionVictoire, 56
 - getTour, 56
 - incrTour, 56
 - JeuDeDeduction, 55
 - main, 56
 - partie, 56
- Joueur, 57
 - entrerPseudo, 58
 - getCombinaison, 58
 - getPseudo, 59
 - Joueur, 58
 - setCombinaison, 59
 - setPseudo, 59
 - verification, 60
- main
 - JeuDeDeduction, 56
- Maj_ensemble
 - IA_Decodeur, 47
 - IA_Decodeur_Mastermind, 50
 - IA_Decodeur_Wordle, 53
- maPlace
 - CombiMastermind, 15
 - CombiWordle, 20
- Mastermind, 60

- Menu, [61](#)
 - [afficherMenu](#), [63](#)
 - [appartientDico](#), [63](#)
 - [appartientEE](#), [63](#)
 - [choisirDifficulteIA](#), [63](#)
 - [choisirModeDeJeu](#), [63](#)
 - [ENSEMBLE_ELEMENT](#), [64](#)
 - [Menu](#), [62](#)
 - [ModeDeJeu](#), [64](#)
 - [NB_CASE](#), [64](#)
 - [NB_ELEMENT](#), [64](#)
 - [NB_TOUR](#), [65](#)
 - [parametreDeJeu](#), [64](#)
- ModeDeJeu
 - [Menu](#), [64](#)
- [NB_CASE](#)
 - [Menu](#), [64](#)
- [NB_ELEMENT](#)
 - [Menu](#), [64](#)
- [NB_TOUR](#)
 - [Menu](#), [65](#)
- [parametreDeJeu](#)
 - [Menu](#), [64](#)
- [partie](#)
 - [JeuDeDeduction](#), [56](#)
- [setCombinaison](#)
 - [Combinaison](#), [17](#)
 - [Joueur](#), [59](#)
- [setPseudo](#)
 - [Joueur](#), [59](#)
- [split](#)
 - [Humain](#), [22](#)
- [std](#), [11](#)
- [toString](#)
 - [CombiMastermind](#), [15](#)
 - [Combinaison](#), [18](#)
 - [CombiWordle](#), [21](#)
- [verification](#)
 - [Joueur](#), [60](#)
- [VerifierEntree](#)
 - [Humain_Codeur_Wordle](#), [29](#)
 - [Humain_Decodeur_Mastermind](#), [34](#)
 - [Humain_Decodeur_Wordle](#), [36](#)
- [verifierEntree](#)
 - [Humain](#), [22](#)
- [Wordle](#), [65](#)