

Jeu de déduction

Generated by Doxygen 1.8.17



---

<b>1</b>	<b>Projet Cpp S8</b>	<b>1</b>
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Codeur Class Reference	7
4.2	Combinaison Class Reference	7
4.2.1	Detailed Description	8
4.2.2	Member Function Documentation	8
4.2.2.1	bienPlace()	8
4.2.2.2	get()	8
4.2.2.3	malPlace()	9
4.2.2.4	setCombinaison()	9
4.2.2.5	split()	9
4.2.2.6	toString()	10
4.3	Decodeur Class Reference	10
4.3.1	Detailed Description	10
4.4	Joueur Class Reference	11
4.4.1	Detailed Description	11
4.4.2	Constructor & Destructor Documentation	11
4.4.2.1	Joueur()	11
4.4.3	Member Function Documentation	12
4.4.3.1	getCombinaison()	12
4.4.3.2	getnumeroTour()	12
4.5	Mastermind Class Reference	12
4.5.1	Detailed Description	13
4.5.2	Member Function Documentation	13
4.5.2.1	getCombinaison()	14
4.6	Menu Class Reference	14
4.6.1	Detailed Description	14
4.6.2	Member Function Documentation	15
4.6.2.1	parametreDeJeu()	15
	<b>Index</b>	<b>17</b>



## Chapter 1

# Projet Cpp S8

Réalisation d'un jeu de déduction type [Mastermind](#) avec implémentation d'une IA.



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Combinaison . . . . .	7
Decodeur . . . . .	10
Joueur . . . . .	11
Codeur . . . . .	7
Mastermind . . . . .	12
Menu . . . . .	14





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Codeur</a>		
	<a href="#">Joueur</a> proposant la combinaison recherchée . . . . .	7
<a href="#">Combinaison</a>		
	Tableau d'entier représentant une combinaison de couleur ou de lettre . . . . .	7
<a href="#">Decodeur</a>		
	<a href="#">Joueur</a> cherchant la combinaison du codeur . . . . .	10
<a href="#">Joueur</a>		
	Classe abstraite regroupant tout joueur . . . . .	11
<a href="#">Mastermind</a>		
	Classe principale permettant de lancement d'une partie. Elle comprend le mail et . . . . .	12
<a href="#">Menu</a>		
	La classe <a href="#">Menu</a> permet à l'utilisateur de choisir le mode et les parametres du jeu . . . . .	14



## Chapter 4

# Class Documentation

### 4.1 Codeur Class Reference

[Joueur](#) proposant la combinaison recherchée.

```
#include <Codeur.hpp>
```

Inheritance diagram for Codeur:

### 4.2 Combinaison Class Reference

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

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

#### 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
- int [bienPlace](#) (const [Combinaison](#) code)  
*renvoie le nombre d'élément (couleur ou lettre) bien placé par rapport à la combinaison du codeur*
- int [malPlace](#) ([Combinaison](#) code)  
*renvoie le nombre d'élément (couleur ou lettre) appartenant à la combinaison du codeur mais mal placé*
- string [toString](#) ()  
*permet l'affichage de la combinaison*
- vector< string > [split](#) (const string &chaine, char delimiteur)  
*permet de séparer une chaine de caractère*

### 4.2.1 Detailed Description

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

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

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.

### 4.2.2 Member Function Documentation

#### 4.2.2.1 bienPlace()

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

renvoie le nombre d'élément (couleur ou lettre) bien placé par rapport à la combinaison du codeur

#### Parameters

<i>code</i>	<a href="#">Combinaison</a> : la combinaison valide recherchée par le décodeur
-------------	--

#### Returns

un entier

#### 4.2.2.2 get()

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

getteur de l'attribut combinaison

#### Returns

retourne un vecteur

#### 4.2.2.3 malPlace()

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

renvoie le nombre d'élément (couleur ou lettre) appartenant à la combinaison du codeur mais mal placé

##### Parameters

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

##### Returns

un entier

#### 4.2.2.4 setCombinaison()

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

setteur de l'attribut combinaison

##### Parameters

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

#### 4.2.2.5 split()

```
vector< string > Combinaison::split (
    const string & chaine,
    char delimitateur )
```

permet de séparer une chaine de caractère

##### Parameters

<i>chaine</i>	string : la chaine à découper
<i>delimiteur</i>	char : le caractère qui va servir à séparer les éléments de la chaine (souvent un ' ')

##### Returns

Renvoie un vecteur de string

#### 4.2.2.6 toString()

```
string Combinaison::toString ( )
```

permet l'affichage de la combinaison

##### Returns

un string

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

### 4.3 Decodeur Class Reference

[Joueur](#) cherchant la combinaison du codeur.

```
#include <Decodeur.hpp>
```

#### Public Member Functions

- [Decodeur](#) ()  
*Constructeur de la classe [Decodeur](#).*

#### 4.3.1 Detailed Description

[Joueur](#) cherchant la combinaison du codeur.

##### Author

Groupe A7

##### Version

1.0

##### Date

avril 2022

Cette classe hérite de la classe [Joueur](#), elle représente l'humain ou l'ordi/IA qui devra chercher la combinaison entrée par le joueur codeur. Elle possède les mêmes attributs que [Joueur](#).

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

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

## 4.4 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, int nb)  
*Constructeur de la classe joueur avec paramètre.*
- [Combinaison](#) [getCombinaison](#) ()  
*Pour accéder à l'attribut combinaison.*
- int [getnumeroTour](#) ()  
*Pour accéder à l'attribut numeroTour.*
- void [entrerCombinaison](#) ()  
*Méthode demandant la saisie d'une combinaison.*

### 4.4.1 Detailed Description

classe abstraite regroupant tout joueur

#### Author

Groupe A7

#### Version

1.0

#### Date

avril 2022

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.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 Joueur()

```
Joueur::Joueur (
    Combinaison comb,
    int nb )
```

Constructeur de la classe joueur avec paramètre.

**Parameters**

<i>comb</i>	la combinaison
<i>nb</i>	le nombre de tour

## 4.4.3 Member Function Documentation

### 4.4.3.1 getCombinaison()

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

Pour accéder à l'attribut combinaison.

**Returns**

retourne la combinaison en attribut

### 4.4.3.2 getnumeroTour()

```
int Joueur::getnumeroTour ( )
```

Pour accéder à l'attribut numeroTour.

**Returns**

un entier

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

## 4.5 Mastermind Class Reference

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

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



## Public Member Functions

- **Mastermind** (const string chaine)
- int **main** ()  
*méthode principale, elle permet de lancement du jeu*
- void **partie** ()  
*Elle 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.*
- void **genererCode** ()  
*Selon le mode de jeu, si le codeur est une IA/ordi alors le code sera générer aléatoirement, sinon on demandera au joueur codeur d'entrée la combinaison qui sera recherchée par le décodeur.*
- **Combinaison** **getCombinaison** ()  
*Cette méthode demandera au décodeur d'entrer une combinaison afin de deviner celle cherchée qui sera retournée en sortie.*
- void **afficherPartie** ()  
*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 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** ()

### 4.5.1 Detailed Description

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

Author

Groupe A7

Version

1.0

Date

avril 2022

Cette classe permet de définir l'objet **Combinaison**. Elle transforme une chaine de caractère (suite de couleur ou de lettre) séparée d'espace en tableau d'entier. Elle possède un seul attribut : le tableau d'entier.

### 4.5.2 Member Function Documentation

#### 4.5.2.1 getCombinaison()

`Combinaison` Mastermind::getCombinaison ( )

Cette méthode demandera au décodeur d'entrer une combinaison afin de deviner celle cherchée qui sera retournée en sortie.

##### Returns

Elle retourne la combinaison du joueur décodeur.

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

## 4.6 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 `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.*

### Public Attributes

- const int `NB_ELEMENT`
- const int `ENSEMBLE_ELEMENT`
- const int `NB_CASE`
- const int `NB_TOUR`
- const int `ModeDeJeu`

#### 4.6.1 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

## 4.6.2 Member Function Documentation

### 4.6.2.1 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

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

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



# Index

- bienPlace
  - Combinaison, [8](#)
- Codeur, [7](#)
- Combinaison, [7](#)
  - bienPlace, [8](#)
  - get, [8](#)
  - malPlace, [8](#)
  - setCombinaison, [9](#)
  - split, [9](#)
  - toString, [9](#)
- Decodeur, [10](#)
- get
  - Combinaison, [8](#)
- getCombinaison
  - Joueur, [12](#)
  - Mastermind, [13](#)
- getnumeroTour
  - Joueur, [12](#)
- Joueur, [11](#)
  - getCombinaison, [12](#)
  - getnumeroTour, [12](#)
  - Joueur, [11](#)
- malPlace
  - Combinaison, [8](#)
- Mastermind, [12](#)
  - getCombinaison, [13](#)
- Menu, [14](#)
  - parametreDeJeu, [15](#)
- parametreDeJeu
  - Menu, [15](#)
- setCombinaison
  - Combinaison, [9](#)
- split
  - Combinaison, [9](#)
- toString
  - Combinaison, [9](#)