

projet decodeur

Generated by Doxygen 1.8.17



<b>1</b>	<b>Projet Cpp S8</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Codeur Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	Codeur()	6
3.1.3	Member Function Documentation	6
3.1.3.1	getMdp()	6
3.1.3.2	setMdp()	6
3.2	Combinaison Class Reference	7
3.2.1	Detailed Description	7
3.2.2	Constructor & Destructor Documentation	7
3.2.2.1	Combinaison()	7
3.2.3	Member Function Documentation	8
3.2.3.1	bienPlace()	8
3.2.3.2	get() [1/2]	8
3.2.3.3	get() [2/2]	8
3.2.3.4	malPlace()	9
3.2.3.5	setCombinaison()	9
3.2.3.6	toString()	9
3.3	Decodeur Class Reference	10
3.3.1	Detailed Description	10
3.4	Joueur Class Reference	10
3.4.1	Detailed Description	11
3.4.2	Member Function Documentation	11
3.4.2.1	getCombinaison()	11
3.4.2.2	getnumeroTour()	12
3.5	Mastermind Class Reference	12
3.5.1	Detailed Description	13
3.5.2	Member Function Documentation	13
3.5.2.1	getCombinaison()	13
3.6	Menu Class Reference	13
3.6.1	Detailed Description	14
3.6.2	Member Function Documentation	14
3.6.2.1	parametreDeJeu()	14
<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

# Class Index

### 2.1 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 . . . . .	5
<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 . . . . .	10
<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 . . . . .	13





## Chapter 3

# Class Documentation

### 3.1 Codeur Class Reference

[Joueur](#) proposant la combinaison recherchée.

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

#### Public Member Functions

- [Codeur](#) (string mdp)  
*Constructeur de la classe [Codeur](#).*
- void [setMdp](#) (string mdp)  
*setteur de l'attribut motDePasse*
- string [getMdp](#) ()  
*getteur de l'attribut motDePasse*

#### 3.1.1 Detailed Description

[Joueur](#) proposant la combinaison recherchée.

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 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.

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 Codeur()

```
Codeur::Codeur (
    string mdp )
```

Constructeur de la classe [Codeur](#).

##### Parameters

<i>mdp</i>	le mot de passe du joueur codeur
------------	----------------------------------

### 3.1.3 Member Function Documentation

#### 3.1.3.1 getMdp()

```
string Codeur::getMdp ( )
```

getteur de l'attribut motDePasse

##### Returns

une chaine de caractère

#### 3.1.3.2 setMdp()

```
void Codeur::setMdp (
    string mdp )
```

setteur de l'attribut motDePasse

##### Parameters

<i>mdp</i>	string
------------	--------

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

- /home/thibaut/Documents/GM4/Projets/Projet\_Cpp\_S8/Codeur.hpp

## 3.2 Combinaison Class Reference

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

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

### Public Member Functions

- [Combinaison](#) (string chaîne)  
*Constructeur de la classe [Combinaison](#).*
- void [setCombinaison](#) (string chaîne)  
*setteur de l'attribut combinaison*
- int \* [get](#) ()  
*getteur de l'attribut combinaison*
- int [get](#) (int i)  
*getteur du ième élément de l'attribut combinaison*
- int [bienPlace](#) ([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*

### 3.2.1 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 tableau d'entier. Elle possède un seul attribut : le tableau d'entier.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 Combinaison()

```
Combinaison::Combinaison (  
    string chaîne )
```

Constructeur de la classe [Combinaison](#).

**Parameters**

<i>chaine</i>	est la chaine de caractère comprenant la suite de couleur ou de lettre séparée d'espace
---------------	---

### 3.2.3 Member Function Documentation

#### 3.2.3.1 bienPlace()

```
int Combinaison::bienPlace (
    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

#### 3.2.3.2 get() [1/2]

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

getteur de l'attribut combinaison

**Returns**

un pointeur sur un entier permettant de gérer un tableau d'entier,

#### 3.2.3.3 get() [2/2]

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

getteur du ième élément de l'attribut combinaison

**Parameters**

<i>i</i>	entier : l'indice
----------	-------------------

**Returns**

un entier

**3.2.3.4 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

**3.2.3.5 setCombinaison()**

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

setteur de l'attribut combinaison

**Parameters**

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

**3.2.3.6 toString()**

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

permet l'affichage de la combinaison

**Returns**

un string

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

- /home/thibaut/Documents/GM4/Projets/Projet\_Cpp\_S8/Combinaison.hpp

### 3.3 Decodeur Class Reference

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

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

#### Public Member Functions

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

#### 3.3.1 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/thibaut/Documents/GM4/Projets/Projet\_Cpp\_S8/Decodeur.hpp

### 3.4 Joueur Class Reference

classe abstraite regroupant tout joueur

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

## Public Member Functions

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

### 3.4.1 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.

### 3.4.2 Member Function Documentation

#### 3.4.2.1 [getCombinaison\(\)](#)

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

Pour accéder à l'attribut combinaison.

#### Returns

retourne la combinaison en attribut

### 3.4.2.2 getnumeroTour()

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

Pour accéder à l'attribut numeroTour.

#### Returns

un entier

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

- /home/thibaut/Documents/GM4/Projets/Projet\_Cpp\_S8/Joueur.hpp

## 3.5 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](#) ()



### 3.5.1 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 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.

### 3.5.2 Member Function Documentation

#### 3.5.2.1 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 file:

- /home/thibaut/Documents/GM4/Projets/Projet\_Cpp\_S8/Mastermind.hpp

## 3.6 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

- 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**

### 3.6.1 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

### 3.6.2 Member Function Documentation

#### 3.6.2.1 [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/thibaut/Documents/GM4/Projets/Projet_Cpp_S8/Menu.hpp`



# Index

- bienPlace
  - Combinaison, [8](#)
- Codeur, [5](#)
  - Codeur, [6](#)
  - getMdp, [6](#)
  - setMdp, [6](#)
- Combinaison, [7](#)
  - bienPlace, [8](#)
  - Combinaison, [7](#)
  - get, [8](#)
  - malPlace, [9](#)
  - setCombinaison, [9](#)
  - toString, [9](#)
- Decodeur, [10](#)
- get
  - Combinaison, [8](#)
- getCombinaison
  - Joueur, [11](#)
  - Mastermind, [13](#)
- getMdp
  - Codeur, [6](#)
- getnumeroTour
  - Joueur, [11](#)
- Joueur, [10](#)
  - getCombinaison, [11](#)
  - getnumeroTour, [11](#)
- malPlace
  - Combinaison, [9](#)
- Mastermind, [12](#)
  - getCombinaison, [13](#)
- Menu, [13](#)
  - parametreDeJeu, [14](#)
- parametreDeJeu
  - Menu, [14](#)
- setCombinaison
  - Combinaison, [9](#)
- setMdp
  - Codeur, [6](#)
- toString
  - Combinaison, [9](#)