Jeu de déduction

Généré par Doxygen 1.8.17

Chapitre 1

Projet Cpp S8

Réalisation d'un jeu de déduction type **Mastermind** (p. 12) avec implémentation d'une IA.

Chapitre 2

Index des classes

2.1 Liste des classes

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

Chapitre 3

Documentation des classes

3.1 Référence de la classe Codeur

Joueur (p. 10) proposant la combinaison recherchée.

`#include <Codeur.hpp>`

Fonctions membres publiques

- **Codeur** (string mdp)
  *Constructeur de la classe **Codeur** (p. 5).*
- void **setMdp** (string mdp)
  setteur de l'attribut motDePasse
- string **getMdp** ()
  getteur de l'attribut motDePasse

3.1.1 Description détaillée

Joueur (p. 10) proposant la combinaison recherchée.

Auteur

Groupe A7

Version

1.0

Date

avril 2022

Cette classe hérite de la classe **Joueur** (p. 10), 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** (p. 10) 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 Documentation des constructeurs et destructeur

3.1.2.1 Codeur()

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

Constructeur de la classe **Codeur** (p. 5).

Paramètres

mdp	le mot de passe du joueur codeur

3.1.3 Documentation des fonctions membres

3.1.3.1 getMdp()

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

getteur de l'attribut motDePasse

Renvoie

une chaine de caractère

3.1.3.2 setMdp()

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

setteur de l'attribut motDePasse

Paramètres

mdp	string

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

— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Codeur.hpp

3.2 Référence de la classe Combinaison

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

`#include <Combinaison.hpp>`

Fonctions membres publiques

- **Combinaison** (string chaine)
  *Constructeur de la classe **Combinaison** (p. 7).*
- void **setCombinaison** (string chaine)
  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 Description détaillée

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

Auteur

Groupe A7

Version

1.0

Date

avril 2022

Cette classe permet de définir l'objet **Combinaison** (p. 7). 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.

3.2.2 Documentation des constructeurs et destructeur

3.2.2.1 Combinaison()

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

Constructeur de la classe **Combinaison** (p. 7).

Paramètres

chaine	est la chaine de caractère comprenant la suite de couleur ou de lettre séparée d'espace

3.2.3 Documentation des fonctions membres

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

Paramètres

code	**Combinaison** (p. 7) : la combinaison valide recherchée par le décodeur

Renvoie

un entier

3.2.3.2 get() [1/2]

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

getteur de l'attribut combinaison

Renvoie

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

Paramètres

i	entier : l'indice

Renvoie

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

Paramètres

code	**Combinaison** (p. 7) : la combinaison valide recherchée par le décodeur

Renvoie

un entier

3.2.3.5 setCombinaison()

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

setteur de l'attribut combinaison

Paramètres

chaine	string

3.2.3.6 toString()

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

permet l'affichage de la combinaison

Renvoie

un string

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

— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Combinaison.hpp

3.3 Référence de la classe Decodeur

Joueur (p. 10) cherchant la combinaison du codeur.

`#include <Decodeur.hpp>`

Fonctions membres publiques

— **Decodeur** ()
*Constructeur de la classe **Decodeur** (p. 10).*

3.3.1 Description détaillée

Joueur (p. 10) cherchant la combinaison du codeur.

Auteur

Groupe A7

Version

1.0

Date

avril 2022

Cette classe hérite de la classe **Joueur** (p. 10), 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** (p. 10).

La documentation de cette classe a été générée à partir du fichier suivant :
— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Decodeur.hpp

3.4 Référence de la classe Joueur

classe abstraite regroupant tout joueur

`#include <Joueur.hpp>`

Fonctions membres publiques

— **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 Description détaillée

classe abstraite regroupant tout joueur

Auteur

Groupe A7

Version

1.0

Date

avril 2022

Il s'agit d'une classe abstraite qui définit tout joueur du **Mastermind** (p. 12). 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 Documentation des fonctions membres

3.4.2.1 getCombinaison()

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

Pour accéder à l'attribut combinaison.

Renvoie

retourne la combinaison en attribut

3.4.2.2 getnumeroTour()

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

Pour accéder à l'attribut numeroTour.

Renvoie

un entier

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

— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Joueur.hpp

3.5 Référence de la classe Mastermind

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

`#include <Mastermind.hpp>`

Fonctions membres publiques

- **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 Description détaillée

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

Auteur

Groupe A7

Version

1.0

Date

avril 2022

Cette classe permet de définir l'objet **Combinaison** (p. 7). 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.

3.5.2 Documentation des fonctions membres

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.

Renvoie

Elle retourne la combinaison du joueur décodeur.

La documentation de cette classe a été générée à partir du fichier suivant :
— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Mastermind.hpp

3.6 Référence de la classe Menu

La classe **Menu** (p. 13) permet à l'utilisateur de choisir le mode et les parametres du jeu.

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

Fonctions membres publiques

— 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.

Attributs publics

— const int **NB_ELEMENT**
— const int **ENSEMBLE_ELEMENT**
— const int **NB_CASE**
— const int **NB_TOUR**
— const int **ModeDeJeu**

3.6.1 Description détaillée

La classe **Menu** (p. 13) permet à l'utilisateur de choisir le mode et les parametres du jeu.

Auteur

Groupe A7

Version

1.0

Date

avril 2022

3.6.2 Documentation des fonctions membres

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.

Paramètres

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

La documentation de cette classe a été générée à partir du fichier suivant :
— /home/kevin/Documents/GM4/S8/Projet_Cpp_S8/Menu.hpp

Index