Jeu de déduction

Generated by Doxygen 1.8.17

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:

Chapter 3

Class Index

3.1 Class List

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

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 chaine 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

code	Combinaison : 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

code	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

chaine	string

4.2.2.5 split()

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

permet de séparer une chaine de caractère

Parameters

chaine	string : la chaine à découper
delimiteur	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

comb	la combinaison
nb	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 parametres 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 parametres 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

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

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

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

Index