

Magic

Généré par Doxygen 1.9.3



<b>1 Index hiérarchique</b>	<b>1</b>
1.1 Hiérarchie des classes	1
<b>2 Index des classes</b>	<b>3</b>
2.1 Liste des classes	3
<b>3 Index des fichiers</b>	<b>5</b>
3.1 Liste des fichiers	5
<b>4 Documentation des classes</b>	<b>7</b>
4.1 Référence de la classe Card	7
4.1.1 Documentation des constructeurs et destructeur	10
4.1.1.1 Card()	10
4.1.1.2 ~Card()	11
4.1.2 Documentation des fonctions membres	11
4.1.2.1 disengage()	11
4.1.2.2 engage()	11
4.1.2.3 getColor()	12
4.1.2.4 getColoredName()	12
4.1.2.5 getIsDiscarded()	12
4.1.2.6 getIsEngaged()	13
4.1.2.7 getManaCost()	13
4.1.2.8 getName()	13
4.1.2.9 isStillOperational()	13
4.1.2.10 manaToString()	14
4.1.2.11 print() [1/2]	14
4.1.2.12 print() [2/2]	15
4.1.2.13 printLine()	15
4.1.2.14 setColor()	16
4.1.2.15 setIsDiscarded()	16
4.1.2.16 setManaCost()	16
4.1.2.17 setName()	16
4.1.3 Documentation des données membres	17
4.1.3.1 color	17
4.1.3.2 ColorCode	17
4.1.3.3 isDiscarded	17
4.1.3.4 isEngaged	17
4.1.3.5 manaCost	17
4.1.3.6 name	17
4.1.3.7 typeOfCard	17
4.2 Référence de la classe CreatureCard	18
4.2.1 Documentation des constructeurs et destructeur	20
4.2.1.1 CreatureCard()	20

4.2.2 Documentation des fonctions membres . . . . .	20
4.2.2.1 attack() . . . . .	21
4.2.2.2 getAttackPower() . . . . .	21
4.2.2.3 getHp() . . . . .	21
4.2.2.4 invoke() . . . . .	21
4.2.2.5 isFirstTurn() . . . . .	22
4.2.2.6 print() . . . . .	22
4.2.2.7 printLine() . . . . .	22
4.2.2.8 setAttackPower() . . . . .	23
4.2.2.9 setHp() . . . . .	23
4.2.3 Documentation des données membres . . . . .	23
4.2.3.1 attackPower . . . . .	23
4.2.3.2 defaultHp . . . . .	24
4.2.3.3 firstTurn . . . . .	24
4.2.3.4 hp . . . . .	24
4.3 Référence de la classe Deck . . . . .	24
4.3.1 Documentation des constructeurs et destructeur . . . . .	26
4.3.1.1 Deck() [1/2] . . . . .	26
4.3.1.2 Deck() [2/2] . . . . .	26
4.3.1.3 ~Deck() . . . . .	27
4.3.2 Documentation des fonctions membres . . . . .	27
4.3.2.1 discardCard() . . . . .	27
4.3.2.2 disengageCards() . . . . .	28
4.3.2.3 drawCard() . . . . .	28
4.3.2.4 generateRandomDeck() . . . . .	28
4.3.2.5 getAttackCards() . . . . .	29
4.3.2.6 getDefenseCards() . . . . .	29
4.3.2.7 getPlayableCards() . . . . .	30
4.3.2.8 JsonToDeck() . . . . .	30
4.3.2.9 playCard() . . . . .	31
4.3.2.10 printInPlayCards() . . . . .	32
4.3.2.11 printLibrary() . . . . .	32
4.3.3 Documentation des données membres . . . . .	32
4.3.3.1 DECK_SIZE . . . . .	32
4.3.3.2 disCards . . . . .	32
4.3.3.3 handCards . . . . .	33
4.3.3.4 inPlayCards . . . . .	33
4.3.3.5 library . . . . .	33
4.4 Référence de la classe Game . . . . .	33
4.4.1 Documentation des constructeurs et destructeur . . . . .	35
4.4.1.1 Game() . . . . .	35
4.4.2 Documentation des fonctions membres . . . . .	35

4.4.2.1 playGame()	35
4.4.3 Documentation des données membres	36
4.4.3.1 MAX_CARDS_IN_HAND	36
4.4.3.2 p1	36
4.4.3.3 p2	36
4.4.3.4 playerTurn	36
4.4.3.5 round	36
4.5 Référence de la classe GameCards	37
4.5.1 Documentation des constructeurs et destructeur	37
4.5.1.1 GameCards()	37
4.5.2 Documentation des fonctions membres	37
4.5.2.1 getCreatures()	38
4.5.2.2 getLands()	38
4.6 Référence de la classe LandCard	38
4.6.1 Documentation des constructeurs et destructeur	41
4.6.1.1 LandCard()	41
4.6.1.2 ~LandCard()	41
4.6.2 Documentation des fonctions membres	41
4.6.2.1 print()	41
4.6.2.2 printLine()	42
4.7 Référence de la classe Player	42
4.7.1 Documentation des constructeurs et destructeur	44
4.7.1.1 Player()	44
4.7.1.2 ~Player()	44
4.7.2 Documentation des fonctions membres	45
4.7.2.1 discardCard()	45
4.7.2.2 disengageCards()	45
4.7.2.3 drawCard()	46
4.7.2.4 getAttackCards()	47
4.7.2.5 getColoredName()	47
4.7.2.6 getDefenseCards()	48
4.7.2.7 getHp()	48
4.7.2.8 getName()	49
4.7.2.9 getPlayableCards()	49
4.7.2.10 playCard()	50
4.7.2.11 setHp()	50
4.7.2.12 setPrintColor()	50
4.7.3 Documentation des données membres	51
4.7.3.1 baseHp	51
4.7.3.2 cards	51
4.7.3.3 currentHp	51
4.7.3.4 isAlive	51

4.7.3.5 name	51
4.7.3.6 printColor	51
4.8 Référence de la classe StrColor	52
4.8.1 Documentation des fonctions membres	52
4.8.1.1 blue()	52
4.8.1.2 cyan()	53
4.8.1.3 green()	53
4.8.1.4 magenta()	53
4.8.1.5 print()	53
4.8.1.6 red()	54
4.8.1.7 yellow()	54
<b>5 Documentation des fichiers</b>	<b>55</b>
5.1 Référence du fichier Source/data/testJson.cpp	55
5.1.1 Documentation des définitions de type	55
5.1.1.1 json	55
5.2 Référence du fichier Source/header/Card.h	56
5.3 Card.h	56
5.4 Référence du fichier Source/header/CreatureCard.h	57
5.5 CreatureCard.h	58
5.6 Référence du fichier Source/header/Deck.h	58
5.7 Deck.h	59
5.8 Référence du fichier Source/header/Game.h	60
5.9 Game.h	61
5.10 Référence du fichier Source/header/GameCards.h	61
5.11 GameCards.h	62
5.12 Référence du fichier Source/header/LandCard.h	63
5.13 LandCard.h	64
5.14 Référence du fichier Source/header/Player.h	64
5.15 Player.h	65
5.16 Référence du fichier Source/header/StrColor.h	66
5.17 StrColor.h	66
5.18 Référence du fichier Source/header/Util.h	67
5.18.1 Documentation des fonctions	67
5.18.1.1 getIndex()	68
5.18.1.2 instanceof()	68
5.19 Util.h	68
5.20 Référence du fichier Source/src/Card.cpp	69
5.21 Référence du fichier Source/src/CreatureCard.cpp	69
5.22 Référence du fichier Source/src/Deck.cpp	70
5.22.1 Documentation des définitions de type	71
5.22.1.1 json	71

---

5.23 Référence du fichier Source/src/Game.cpp . . . . .	71
5.23.1 Documentation des fonctions . . . . .	71
5.23.1.1 centerString() . . . . .	72
5.24 Référence du fichier Source/src/GameCards.cpp . . . . .	72
5.25 Référence du fichier Source/src/LandCard.cpp . . . . .	73
5.26 Référence du fichier Source/src/Main.cpp . . . . .	73
5.26.1 Documentation des définitions de type . . . . .	74
5.26.1.1 json . . . . .	74
5.26.2 Documentation des fonctions . . . . .	74
5.26.2.1 main() . . . . .	75
5.27 Référence du fichier Source/src/Player.cpp . . . . .	75
5.28 Référence du fichier Source/src/StrColor.cpp . . . . .	76
<b>Index</b>	<b>77</b>





# Chapitre 1

## Index hiérarchique

### 1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Card . . . . .	7
CreatureCard . . . . .	18
LandCard . . . . .	38
Deck . . . . .	24
Game . . . . .	33
GameCards . . . . .	37
Player . . . . .	42
StrColor . . . . .	52



## Chapitre 2

# Index des classes

### 2.1 Liste des classes

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

<b>Card</b>	7
<b>CreatureCard</b>	18
<b>Deck</b>	24
<b>Game</b>	33
<b>GameCards</b>	37
<b>LandCard</b>	38
<b>Player</b>	42
<b>StrColor</b>	52



## Chapitre 3

# Index des fichiers

### 3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

Source/data/ <b>testJson.cpp</b> . . . . .	55
Source/header/ <b>Card.h</b> . . . . .	56
Source/header/ <b>CreatureCard.h</b> . . . . .	57
Source/header/ <b>Deck.h</b> . . . . .	58
Source/header/ <b>Game.h</b> . . . . .	60
Source/header/ <b>GameCards.h</b> . . . . .	61
Source/header/ <b>LandCard.h</b> . . . . .	63
Source/header/ <b>Player.h</b> . . . . .	64
Source/header/ <b>StrColor.h</b> . . . . .	66
Source/header/ <b>Util.h</b> . . . . .	67
Source/src/ <b>Card.cpp</b> . . . . .	69
Source/src/ <b>CreatureCard.cpp</b> . . . . .	69
Source/src/ <b>Deck.cpp</b> . . . . .	70
Source/src/ <b>Game.cpp</b> . . . . .	71
Source/src/ <b>GameCards.cpp</b> . . . . .	72
Source/src/ <b>LandCard.cpp</b> . . . . .	73
Source/src/ <b>Main.cpp</b> . . . . .	73
Source/src/ <b>Player.cpp</b> . . . . .	75
Source/src/ <b>StrColor.cpp</b> . . . . .	76



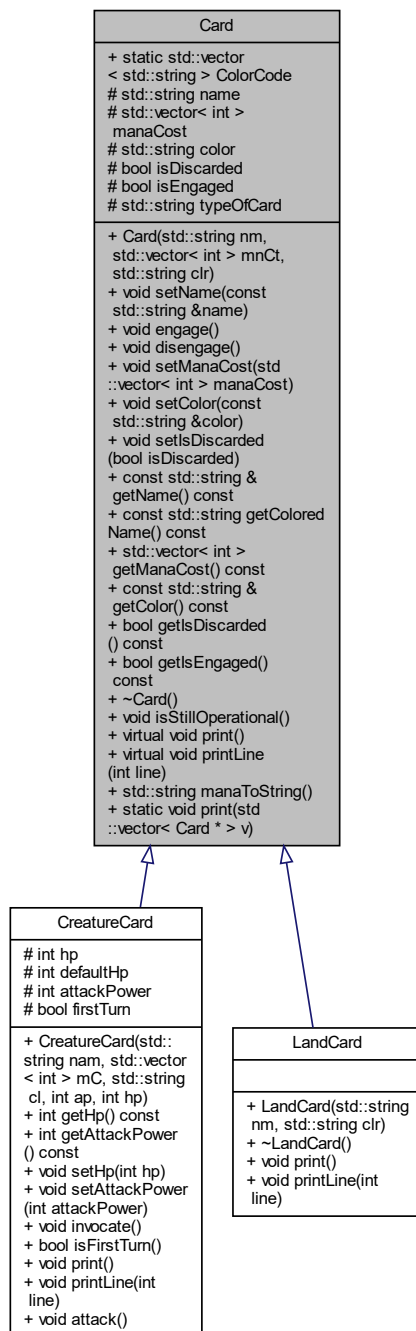
## Chapitre 4

# Documentation des classes

### 4.1 Référence de la classe Card

```
#include <Card.h>
```

Graphe d'héritage de Card:





Graphe de collaboration de Card:

Card
<pre> + static std::vector &lt; std::string &gt; ColorCode # std::string name # std::vector&lt; int &gt; manaCost # std::string color # bool isDiscarded # bool isEngaged # std::string typeOfCard </pre>
<pre> + Card(std::string nm, std::vector&lt; int &gt; mnCt, std::string clr) + void setName(const std::string &amp;name) + void engage() + void disengage() + void setManaCost(std ::vector&lt; int &gt; manaCost) + void setColor(const std::string &amp;color) + void setIsDiscarded (bool isDiscarded) + const std::string &amp; getName() const + const std::string getColored Name() const + std::vector&lt; int &gt; getManaCost() const + const std::string &amp; getColor() const + bool getIsDiscarded () const + bool getIsEngaged() const + ~Card() + void isStillOperational() + virtual void print() + virtual void printLine (int line) + std::string manaToString() + static void print(std ::vector&lt; Card * &gt; v) </pre>

## Fonctions membres publiques

- **Card** (std::string nm, std::vector< int > mnCt, std::string clr)  
*Constructeur de card.*
- void **setName** (const std::string & name)  
*Fonction qui permet de saisir le nom d'une carte.*
- void **engage** ()

- *Engager une carte.*  
void **disengage** ()
- *Désengager une carte.*  
void **setManaCost** (std::vector< int > **manaCost**)
- *Saisir le cout en terrain.*  
void **setColor** (const std::string & **color**)
- *Saisir la couleur.*  
void **setIsDiscarded** (bool **isDiscarded**)
- *Mettre une carte au cimetière.*  
const std::string & **getName** () const
- *Récupérer le nom de la carte.*  
const std::string **getColoredName** () const
- *Récupérer le nom de la couleur de la carte.*  
std::vector< int > **getManaCost** () const
- *Affiche le nom de la carte coloré*  
const std::string & **getColor** () const
- *Récupérer la couleur.*  
bool **getIsDiscarded** () const
- *Etat dans le cimetière ou pas.*  
bool **getIsEngaged** () const
- *Engagé ou pas.*  
~**Card** ()
- *Destructeur.*  
void **isStillOperational** ()
- virtual void **print** ()
- *Fonction d'affichage.*  
virtual void **printLine** (int line)
- *Fonction d'affichage par ligne Pour l'affiche en ligne et non pas en colonne.*  
std::string **manaToString** ()
- *Cout en terrain en string.*

## Fonctions membres publiques statiques

- static void **print** (std::vector< **Card** \* > v)

## Attributs publics statiques

- static std::vector< std::string > **ColorCode** = {"green", "blue", "red", "yellow", "black", "white"}

## Attributs protégés

- std::string **name**
- std::vector< int > **manaCost**
- std::string **color**
- bool **isDiscarded** = false
- bool **isEngaged** = false
- std::string **typeOfCard**

### 4.1.1 Documentation des constructeurs et destructeur

#### 4.1.1.1 Card()

```
Card::Card (
    std::string nm,
    std::vector< int > mnCt,
    std::string clr )
```

Constructeur de card.

## Paramètres

<i>nm</i>	: le nom de la carte
<i>mnCt</i>	: son cout en terrain
<i>clr</i>	: sa couleur

## 4.1.1.2 ~Card()

```
Card::~~Card ( )
```

Destructeur.

## 4.1.2 Documentation des fonctions membres

## 4.1.2.1 disengage()

```
void Card::disengage ( )
```

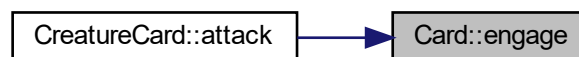
Désengager une carte.

## 4.1.2.2 engage()

```
void Card::engage ( )
```

Engager une carte.

Voici le graphe des appelants de cette fonction :



#### 4.1.2.3 getColor()

```
const std::string & Card::getColor ( ) const
```

Récupérer la couleur.

Renvoie

#### 4.1.2.4 getColoredName()

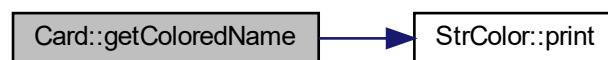
```
const std::string Card::getColoredName ( ) const
```

Récupérer le nom de la couleur de la carte.

Renvoie

couleur

Voici le graphe d'appel pour cette fonction :



#### 4.1.2.5 getIsDiscarded()

```
bool Card::getIsDiscarded ( ) const
```

Etat dans le cimetière ou pas.

Renvoie

#### 4.1.2.6 getIsEngaged()

```
bool Card::getIsEngaged ( ) const
```

Engagé ou pas.

Renvoie

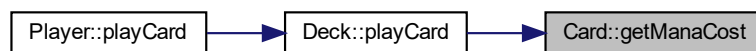
#### 4.1.2.7 getManaCost()

```
std::vector< int > Card::getManaCost ( ) const
```

Affiche le nom de la carte coloré

Renvoie

Voici le graphe des appelants de cette fonction :



#### 4.1.2.8 getName()

```
const std::string & Card::getName ( ) const
```

Récupérer le nom de la carte.

Renvoie

name

#### 4.1.2.9 isStillOperational()

```
void Card::isStillOperational ( )
```

#### 4.1.2.10 manaToString()

```
std::string Card::manaToString ( )
```

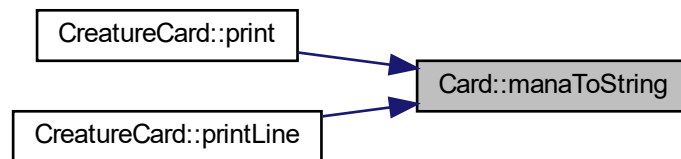
Cout en terrain en string.

Renvoie

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.1.2.11 print() [1/2]

```
void Card::print ( ) [virtual]
```

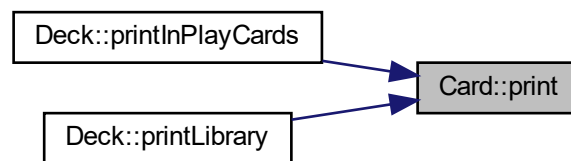
Fonction d'affichage.

Réimplémentée dans **CreatureCard** (p. 22), et **LandCard** (p. 41).

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.1.2.12 `print()` [2/2]

```
void Card::print (
    std::vector< Card * > v ) [static]
```

#### 4.1.2.13 `printLine()`

```
void Card::printLine (
    int line ) [virtual]
```

Fonction d'affichage par ligne Pour l'affiche en ligne et non pas en colonne.

##### Paramètres

<i>line</i>	
-------------	--

Réimplémentée dans **CreatureCard** (p. 22), et **LandCard** (p. 41).

Voici le graphe d'appel pour cette fonction :



#### 4.1.2.14 setColor()

```
void Card::setColor (
    const std::string & color )
```

Saisir la couleur.

##### Paramètres

<i>color</i>	
--------------	--

#### 4.1.2.15 setIsDiscarded()

```
void Card::setIsDiscarded (
    bool isDiscarded )
```

Mettre une carte au cimetière.

##### Paramètres

<i>isDiscarded</i>	
--------------------	--

#### 4.1.2.16 setManaCost()

```
void Card::setManaCost (
    std::vector< int > manaCost )
```

Saisir le cout en terrain.

##### Paramètres

<i>manaCost</i>	
-----------------	--

#### 4.1.2.17 setName()

```
void Card::setName (
    const std::string & name )
```

Fonction qui permet de saisir le nom d'une carte.

##### Paramètres

<i>name</i>	le nom de la carte
-------------	--------------------



### 4.1.3 Documentation des données membres

#### 4.1.3.1 color

```
std::string Card::color [protected]
```

#### 4.1.3.2 ColorCode

```
std::vector< std::string > Card::ColorCode = {"green","blue","red","yellow","black","white"}  
[static]
```

#### 4.1.3.3 isDiscarded

```
bool Card::isDiscarded = false [protected]
```

#### 4.1.3.4 isEngaged

```
bool Card::isEngaged = false [protected]
```

#### 4.1.3.5 manaCost

```
std::vector<int> Card::manaCost [protected]
```

#### 4.1.3.6 name

```
std::string Card::name [protected]
```

#### 4.1.3.7 typeOfCard

```
std::string Card::typeOfCard [protected]
```

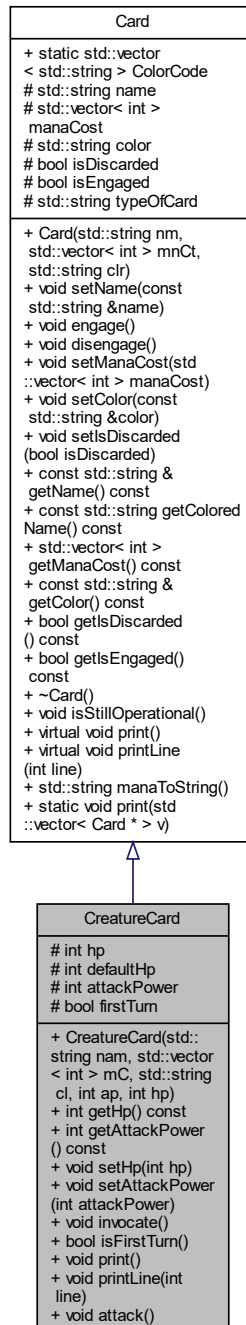
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **Card.h**
- Source/src/ **Card.cpp**

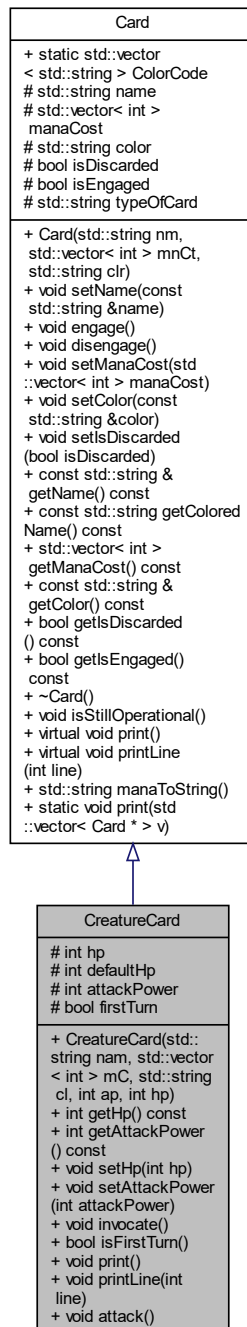
## 4.2 Référence de la classe CreatureCard

```
#include <CreatureCard.h>
```

Grappe d'héritage de CreatureCard:



Graphe de collaboration de CreatureCard:



## Fonctions membres publiques

- **CreatureCard** (std::string nam, std::vector< int > mC, std::string cl, int ap, int **hp**)  
*Constructeurs des cartes créatures.*
- int **getHp** () const  
*Récuper les points de vie.*
- int **getAttackPower** () const

*Recuper les points d'attaques.*

- void **setHp** (int **hp**)
- Set les hp.*
- void **setAttackPower** (int **attackPower**)
- Set l'attaque.*
- void **invoke** ()
- bool **isFirstTurn** ()
- boolean si premier tour*
- void **print** ()
- Affichage.*
- void **printLine** (int line)
- Affichage par ligne.*
- void **attack** ()

## Attributs protégés

- int **hp**
- int **defaultHp**
- int **attackPower**
- bool **firstTurn**

## Membres hérités additionnels

### 4.2.1 Documentation des constructeurs et destructeur

#### 4.2.1.1 CreatureCard()

```

CreatureCard::CreatureCard (
    std::string nam,
    std::vector< int > mC,
    std::string cl,
    int ap,
    int hp )

```

Constructeurs des cartes créatures.

#### Paramètres

<i>nam</i>	
<i>cout</i>	en terrain
<i>couelur</i>	
<i>attaque</i>	
<i>point</i>	de vie

### 4.2.2 Documentation des fonctions membres

#### 4.2.2.1 attack()

```
void CreatureCard::attack ( )
```

Voici le graphe d'appel pour cette fonction :



#### 4.2.2.2 getAttackPower()

```
int CreatureCard::getAttackPower ( ) const
```

Récupérer les points d'attaques.

**Renvoie**

attaque

#### 4.2.2.3 getHp()

```
int CreatureCard::getHp ( ) const
```

Récupérer les points de vie.

**Renvoie**

hp

#### 4.2.2.4 invoke()

```
void CreatureCard::invoke ( )
```

#### 4.2.2.5 isFirstTurn()

```
bool CreatureCard::isFirstTurn ( )
```

boolean si premier tour

Renvoie

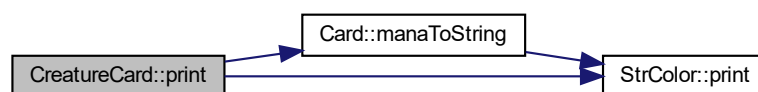
#### 4.2.2.6 print()

```
void CreatureCard::print ( ) [virtual]
```

Affichage.

Réimplémentée à partir de **Card** (p. 14).

Voici le graphe d'appel pour cette fonction :



#### 4.2.2.7 printLine()

```
void CreatureCard::printLine (
    int line ) [virtual]
```

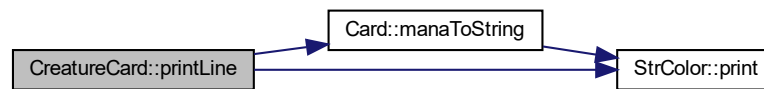
Affichage par ligne.

Paramètres

<i>line</i>	
-------------	--

Réimplémentée à partir de **Card** (p. 15).

Voici le graphe d'appel pour cette fonction :



#### 4.2.2.8 setAttackPower()

```
void CreatureCard::setAttackPower (
    int attackPower )
```

Set l'attaque.

Paramètres

<i>attackPower</i>	
--------------------	--

#### 4.2.2.9 setHp()

```
void CreatureCard::setHp (
    int hp )
```

Set les hp.

Paramètres

<i>hp</i>	
-----------	--

### 4.2.3 Documentation des données membres

#### 4.2.3.1 attackPower

```
int CreatureCard::attackPower [protected]
```

#### 4.2.3.2 defaultHp

```
int CreatureCard::defaultHp [protected]
```

#### 4.2.3.3 firstTurn

```
bool CreatureCard::firstTurn [protected]
```

#### 4.2.3.4 hp

```
int CreatureCard::hp [protected]
```

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

- Source/header/ **CreatureCard.h**
- Source/src/ **CreatureCard.cpp**

### 4.3 Référence de la classe Deck

```
#include <Deck.h>
```



Graphe de collaboration de Deck:

Deck
<pre> + static const int DECK_SIZE # std::vector&lt; Card * &gt; inPlayCards # std::vector&lt; Card * &gt; handCards # std::vector&lt; Card * &gt; library # std::vector&lt; Card * &gt; disCards </pre>
<pre> + Deck() + Deck(std::string nomDeck) + ~Deck() + void printLibrary() + void printInPlayCards() + void generateRandomDeck() + bool drawCard() + void disengageCards() + std::vector&lt; Card * &gt; getPlayableCards() + std::vector&lt; Card * &gt; getAttackCards() + std::vector&lt; Card * &gt; getDefenseCards() + void playCard(Card *c) + void discardCard(Card *c) + void JsonToDeck(std ::string nomDeck) </pre>

## Fonctions membres publiques

- **Deck** ()  
*Constructeur avec un deck par default.*
- **Deck** (std::string nomDeck)  
*Constructeur avec le choix d'un deck stocké en json.*
- **~Deck** ()
- void **printLibrary** ()  
*Affiche toute les cartes du joueurs.*
- void **printInPlayCards** ()  
*Affiche les cartes que le joueur peut jouer.*
- void **generateRandomDeck** ()  
*Mélange le deck.*
- bool **drawCard** ()  
*Piocher une carte.*
- void **disengageCards** ()  
*désengager la carte*
- std::vector< **Card** \* > **getPlayableCards** ()  
*Obtenir les cartes jouable par le joueurs.*

- `std::vector< Card * > getAttackCards ()`  
*Les cartes qui peuvent attaquer.*
- `std::vector< Card * > getDefenseCards ()`  
*Les cartes qui peuvent défendre.*
- `void playCard ( Card *c)`  
*Les cartes jouables.*
- `void discardCard ( Card *c)`  
*Les cartes dans le cimetière.*
- `void JsonToDeck (std::string nomDeck)`  
*Fonction qui convertit un fichier json de carte en deck jouable.*

### Attributs publics statiques

- `static const int DECK_SIZE = 30`

### Attributs protégés

- `std::vector< Card * > inPlayCards`
- `std::vector< Card * > handCards`
- `std::vector< Card * > library`
- `std::vector< Card * > disCards`

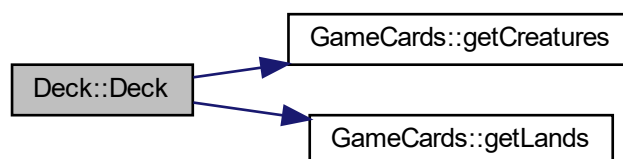
## 4.3.1 Documentation des constructeurs et destructeur

### 4.3.1.1 Deck() [1/2]

`Deck::Deck ( )`

Constructeur avec un deck par default.

Voici le graphe d'appel pour cette fonction :



### 4.3.1.2 Deck() [2/2]

`Deck::Deck (`  
    `std::string nomDeck )`

Constructeur avec le choix d'un deck stocké en json.

## Paramètres

<i>nomDeck</i>	
----------------	--

Voici le graphe d'appel pour cette fonction :



## 4.3.1.3 ~Deck()

Deck::~~Deck ( )

## 4.3.2 Documentation des fonctions membres

## 4.3.2.1 discardCard()

```
void Deck::discardCard (  
    Card * c )
```

Les cartes dans le cimetière.

## Paramètres

<i>c</i>	
----------	--

Voici le graphe des appelants de cette fonction :

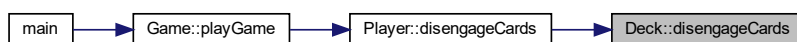


#### 4.3.2.2 disengageCards()

```
void Deck::disengageCards ( )
```

désengager la carte

Voici le graphe des appelants de cette fonction :



#### 4.3.2.3 drawCard()

```
bool Deck::drawCard ( )
```

Piocher une carte.

Renvoie

Voici le graphe des appelants de cette fonction :



#### 4.3.2.4 generateRandomDeck()

```
void Deck::generateRandomDeck ( )
```

Mélange le deck.

#### 4.3.2.5 getAttackCards()

```
std::vector< Card * > Deck::getAttackCards ( )
```

Les cartes qui peuvent attaquer.

Renvoie

Voici le graphe des appelants de cette fonction :



#### 4.3.2.6 getDefenseCards()

```
std::vector< Card * > Deck::getDefenseCards ( )
```

Les cartes qui peuvent défendre.

Renvoie

Voici le graphe des appelants de cette fonction :



#### 4.3.2.7 getPlayableCards()

```
std::vector< Card * > Deck::getPlayableCards ( )
```

Obtenir les cartes jouable par le joueurs.

Renvoie

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.3.2.8 JsonToDeck()

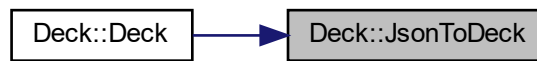
```
void Deck::JsonToDeck (
    std::string nomDeck )
```

Fonction qui convertit un fichier json de carte en deck jouable.

Paramètres

<i>nomDeck</i>	
----------------	--

Voici le graphe des appelants de cette fonction :



#### 4.3.2.9 playCard()

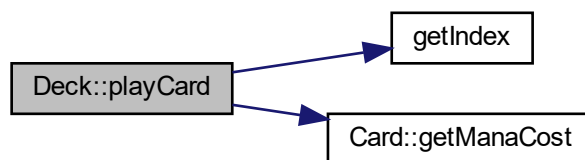
```
void Deck::playCard (  
    Card * c )
```

Les cartes jouables.

##### Paramètres

<i>c</i>	
----------	--

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.3.2.10 `printlnPlayCards()`

```
void Deck::printlnPlayCards ( )
```

Affiche les cartes que le joueur peut jouer.

Voici le graphe d'appel pour cette fonction :

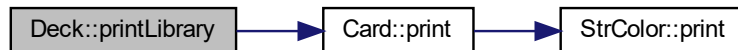


#### 4.3.2.11 `printLibrary()`

```
void Deck::printLibrary ( )
```

Affiche toute les cartes du joueurs.

Voici le graphe d'appel pour cette fonction :



### 4.3.3 Documentation des données membres

#### 4.3.3.1 `DECK_SIZE`

```
const int Deck::DECK_SIZE = 30 [static]
```

#### 4.3.3.2 `disCards`

```
std::vector< Card*> Deck::disCards [protected]
```



#### 4.3.3.3 handCards

```
std::vector< Card*> Deck::handCards [protected]
```

#### 4.3.3.4 inPlayCards

```
std::vector< Card*> Deck::inPlayCards [protected]
```

#### 4.3.3.5 library

```
std::vector< Card*> Deck::library [protected]
```

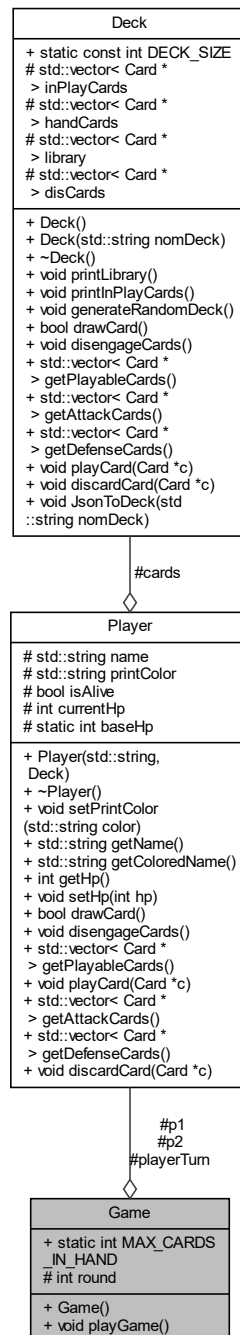
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **Deck.h**
- Source/src/ **Deck.cpp**

## 4.4 Référence de la classe Game

```
#include <Game.h>
```

Graphe de collaboration de Game:



## Fonctions membres publiques

- **Game** ()
- void **playGame** ()  
Lance la partie.

## Attributs publics statiques

- static int **MAX\_CARDS\_IN\_HAND** = 7  
*Nombre de carte maximum dans la main.*

## Attributs protégés

- Player p1
- Player p2
- Player \* playerTurn
- int round

### 4.4.1 Documentation des constructeurs et destructeur

#### 4.4.1.1 Game()

```
Game::Game ( )
```

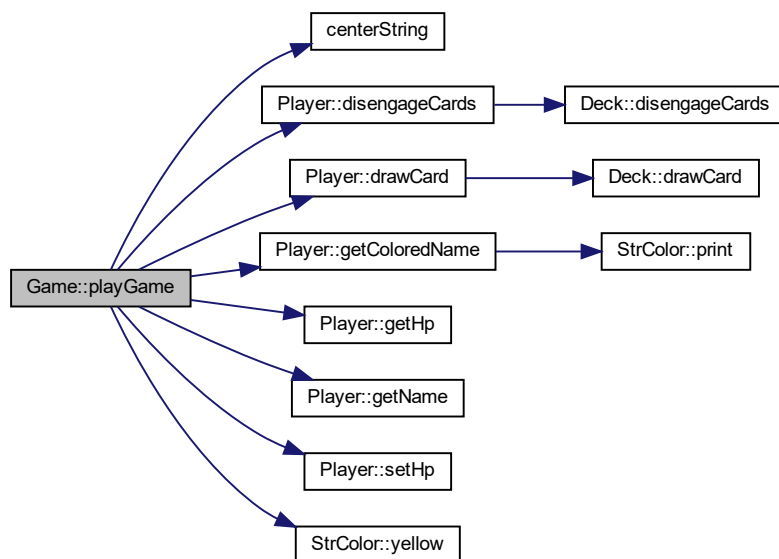
### 4.4.2 Documentation des fonctions membres

#### 4.4.2.1 playGame()

```
void Game::playGame ( )
```

Lance la partie.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 4.4.3 Documentation des données membres

#### 4.4.3.1 MAX\_CARDS\_IN\_HAND

```
int Game::MAX_CARDS_IN_HAND = 7 [static]
```

Nombre de carte maximum dans la main.

#### 4.4.3.2 p1

```
Player Game::p1 [protected]
```

#### 4.4.3.3 p2

```
Player Game::p2 [protected]
```

#### 4.4.3.4 playerTurn

```
Player* Game::playerTurn [protected]
```

#### 4.4.3.5 round

```
int Game::round [protected]
```

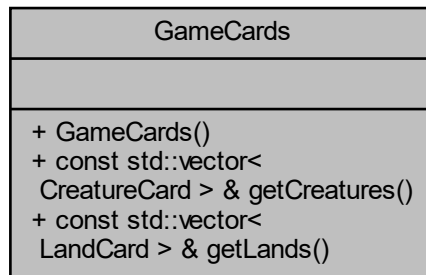
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **Game.h**
- Source/src/ **Game.cpp**

## 4.5 Référence de la classe GameCards

```
#include <GameCards.h>
```

Graphe de collaboration de GameCards:



### Fonctions membres publiques

- **GameCards ()**  
*Créer un deck par défaut.*
- const std::vector< **CreatureCard** > & **getCreatures ()**
- const std::vector< **LandCard** > & **getLands ()**

### 4.5.1 Documentation des constructeurs et destructeur

#### 4.5.1.1 GameCards()

```
GameCards::GameCards ( )
```

Créer un deck par défaut.

### 4.5.2 Documentation des fonctions membres

#### 4.5.2.1 getCreatures()

```
const std::vector< CreatureCard > & GameCards::getCreatures ( )
```

Voici le graphe des appelants de cette fonction :



#### 4.5.2.2 getLands()

```
const std::vector< LandCard > & GameCards::getLands ( )
```

Voici le graphe des appelants de cette fonction :



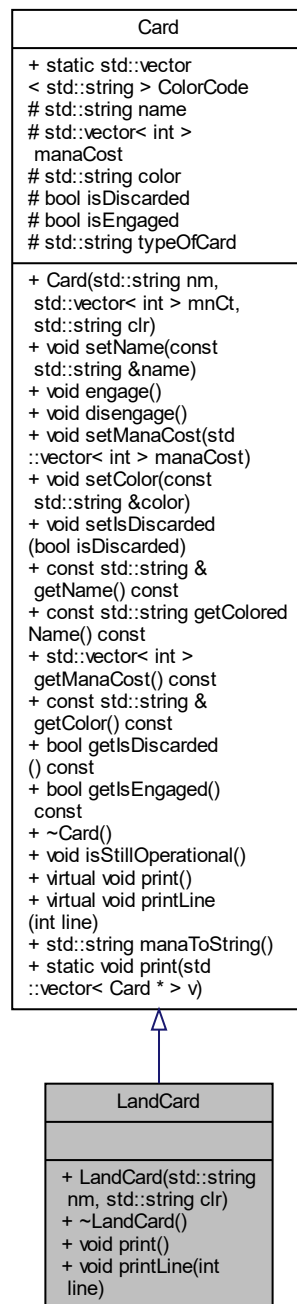
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **GameCards.h**
- Source/src/ **GameCards.cpp**

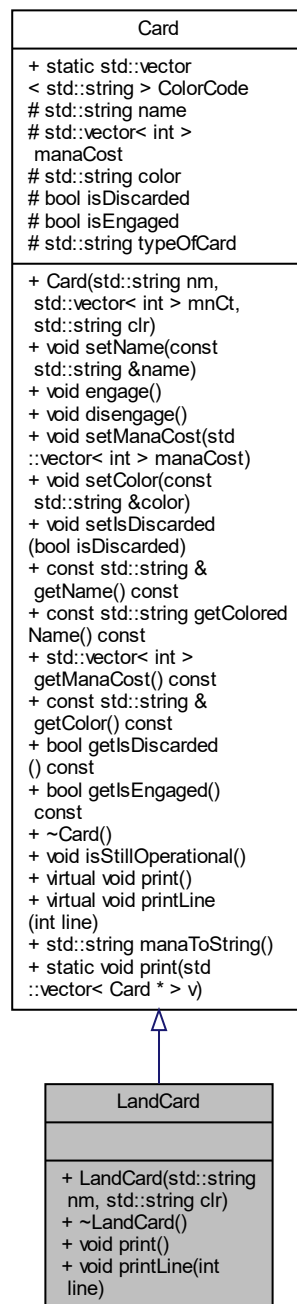
## 4.6 Référence de la classe LandCard

```
#include <LandCard.h>
```

Graphe d'héritage de LandCard:



Graphe de collaboration de LandCard:



## Fonctions membres publiques

- **LandCard** (std::string nm, std::string clr)  
*Constructeur d'une carte terrain.*
- **~LandCard** ()
- void **print** ()  
*Fonction d'affichage.*
- void **printLine** (int line)  
*Fonction d'affichage par ligne Pour l'affiche en ligne et non pas en colonne.*



## Membres hérités additionnels

### 4.6.1 Documentation des constructeurs et destructeur

#### 4.6.1.1 LandCard()

```
LandCard::LandCard (
    std::string nm,
    std::string clr )
```

Constructeur d'une carte terrain.

##### Paramètres

<i>nom</i>	
<i>couleur</i>	

#### 4.6.1.2 ~LandCard()

```
LandCard::~~LandCard ( ) [inline]
```

### 4.6.2 Documentation des fonctions membres

#### 4.6.2.1 print()

```
void LandCard::print ( ) [virtual]
```

Fonction d'affichage.

Réimplémentée à partir de **Card** (p. 14).

Voici le graphe d'appel pour cette fonction :



#### 4.6.2.2 printLine()

```
void LandCard::printLine (
    int line ) [virtual]
```

Fonction d'affichage par ligne Pour l'affiche en ligne et non pas en colonne.

##### Paramètres

<i>line</i>	
-------------	--

Réimplémentée à partir de **Card** (p. 15).

Voici le graphe d'appel pour cette fonction :



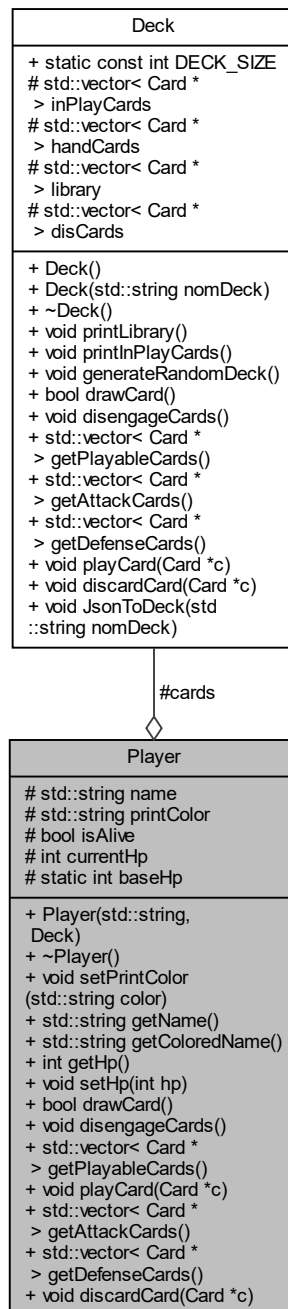
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **LandCard.h**
- Source/src/ **LandCard.cpp**

## 4.7 Référence de la classe Player

```
#include <Player.h>
```

Graphe de collaboration de Player:



## Fonctions membres publiques

- **Player** (std::string, **Deck**)  
*Constructeur d'un joueur.*
- **~Player** ()
- void **setPrintColor** (std::string color)  
*set la couleur du joueur*

- std::string **getName** ()  
*Set le nom du joueurs.*
- std::string **getColoredName** ()
- int **getHp** ()  
*récupérer les hp du joueurs*
- void **setHp** (int hp)  
*set les hp du joueurs*
- bool **drawCard** ()  
*piocher une carte*
- void **disengageCards** ()  
*désengager ses cartes*
- std::vector< **Card** \* > **getPlayableCards** ()  
*ses cartes jouables*
- void **playCard** ( **Card** \*c)
- std::vector< **Card** \* > **getAttackCards** ()  
*ses cartes qui peuvent attaquer*
- std::vector< **Card** \* > **getDefenseCards** ()  
*ses cartes qui peuvent défendre*
- void **discardCard** ( **Card** \*c)  
*envoyer une carte au cimetière*

### Attributs protégés

- std::string **name**
- std::string **printColor**
- bool **isAlive**
- int **currentHp**
- **Deck** **cards**

### Attributs protégés statiques

- static int **baseHp** = 4

## 4.7.1 Documentation des constructeurs et destructeur

### 4.7.1.1 Player()

```
Player::Player (
    std::string name,
    Deck deck )
```

Constructeur d'un joueur.

### 4.7.1.2 ~Player()

```
Player::~~Player ( ) [inline]
```

## 4.7.2 Documentation des fonctions membres

### 4.7.2.1 discardCard()

```
void Player::discardCard (
    Card * c )
```

envoyer une carte au cimetière

#### Paramètres

<i>c</i>	
----------	--

Voici le graphe d'appel pour cette fonction :



### 4.7.2.2 disengageCards()

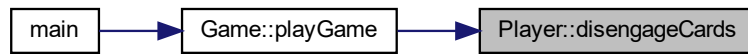
```
void Player::disengageCards ( )
```

désengager ses cartes

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.7.2.3 drawCard()

```
bool Player::drawCard ( )
```

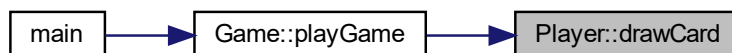
piocher une carte

Renvoie

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.7.2.4 getAttackCards()

```
std::vector< Card * > Player::getAttackCards ( )
```

ses cartes qui peuvent attaquer

Renvoie

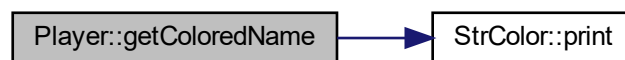
Voici le graphe d'appel pour cette fonction :



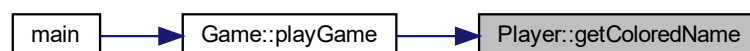
#### 4.7.2.5 getColoredName()

```
std::string Player::getColoredName ( )
```

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 4.7.2.6 getDefenseCards()

```
std::vector< Card * > Player::getDefenseCards ( )
```

ses cartes qui peuvent défendre

Renvoie

Voici le graphe d'appel pour cette fonction :



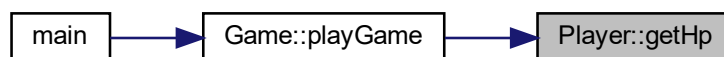
#### 4.7.2.7 getHp()

```
int Player::getHp ( )
```

récupérer les hp du joueurs

Renvoie

Voici le graphe des appelants de cette fonction :





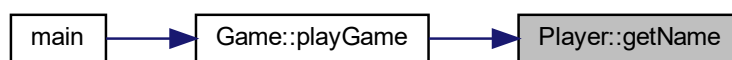
#### 4.7.2.8 getName()

```
std::string Player::getName ( )
```

Set le nom du joueurs.

Renvoie

Voici le graphe des appelants de cette fonction :



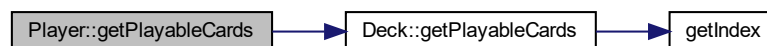
#### 4.7.2.9 getPlayableCards()

```
std::vector< Card * > Player::getPlayableCards ( )
```

ses cartes jouables

Renvoie

Voici le graphe d'appel pour cette fonction :



#### 4.7.2.10 playCard()

```
void Player::playCard (
    Card * c )
```

Voici le graphe d'appel pour cette fonction :



#### 4.7.2.11 setHp()

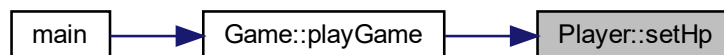
```
void Player::setHp (
    int hp )
```

set les hp du joueurs

Paramètres

<i>hp</i>	
-----------	--

Voici le graphe des appelants de cette fonction :



#### 4.7.2.12 setPrintColor()

```
void Player::setPrintColor (
    std::string color )
```

set la couleur du joueur

## Paramètres

<i>color</i>	
--------------	--

### 4.7.3 Documentation des données membres

#### 4.7.3.1 baseHp

```
int Player::baseHp = 4 [static], [protected]
```

#### 4.7.3.2 cards

```
Deck Player::cards [protected]
```

#### 4.7.3.3 currentHp

```
int Player::currentHp [protected]
```

#### 4.7.3.4 isAlive

```
bool Player::isAlive [protected]
```

#### 4.7.3.5 name

```
std::string Player::name [protected]
```

#### 4.7.3.6 printColor

```
std::string Player::printColor [protected]
```

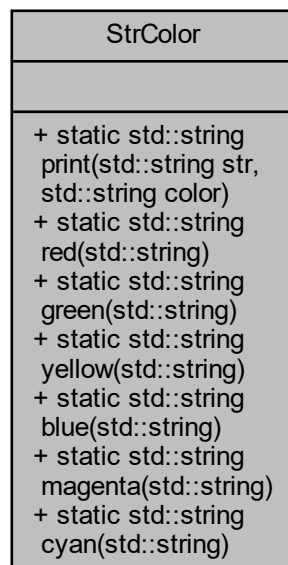
La documentation de cette classe a été générée à partir du fichier suivant :

- Source/header/ **Player.h**
- Source/src/ **Player.cpp**

## 4.8 Référence de la classe StrColor

```
#include <StrColor.h>
```

Graphe de collaboration de StrColor:



### Fonctions membres publiques statiques

- static std::string **print** (std::string str, std::string color)
- static std::string **red** (std::string)
- static std::string **green** (std::string)
- static std::string **yellow** (std::string)
- static std::string **blue** (std::string)
- static std::string **magenta** (std::string)
- static std::string **cyan** (std::string)

### 4.8.1 Documentation des fonctions membres

#### 4.8.1.1 blue()

```
std::string StrColor::blue (  
    std::string str ) [static]
```

## 4.8.1.2 cyan()

```
std::string StrColor::cyan (
    std::string str ) [static]
```

## 4.8.1.3 green()

```
std::string StrColor::green (
    std::string str ) [static]
```

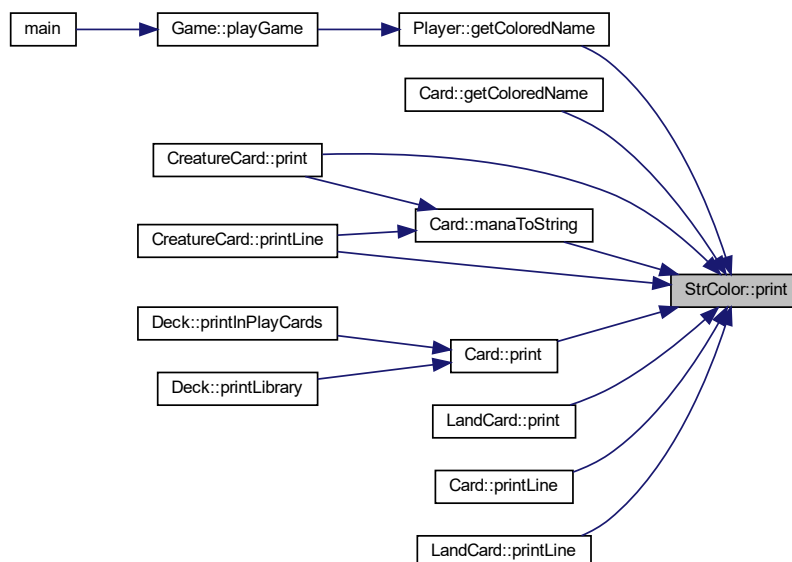
## 4.8.1.4 magenta()

```
std::string StrColor::magenta (
    std::string str ) [static]
```

## 4.8.1.5 print()

```
std::string StrColor::print (
    std::string str,
    std::string color ) [static]
```

Voici le graphe des appelants de cette fonction :



#### 4.8.1.6 red()

```
std::string StrColor::red (
    std::string str ) [static]
```

#### 4.8.1.7 yellow()

```
std::string StrColor::yellow (
    std::string str ) [static]
```

Voici le graphe des appelants de cette fonction :



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

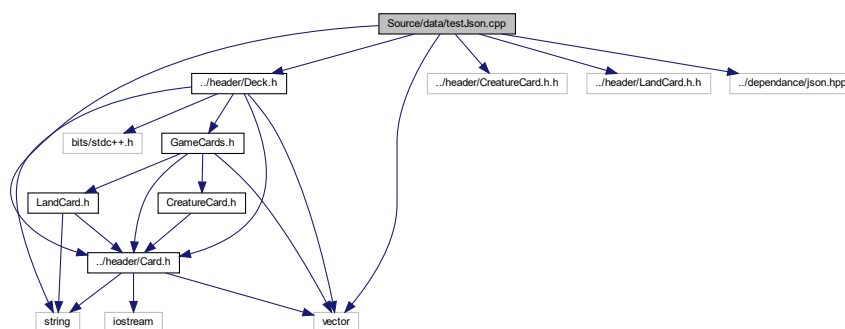
- Source/header/ **StrColor.h**
- Source/src/ **StrColor.cpp**

## Chapitre 5

# Documentation des fichiers

### 5.1 Référence du fichier Source/data/testJson.cpp

```
#include "../header/Card.h"
#include "../header/CreatureCard.h.h"
#include "../header/LandCard.h.h"
#include "../header/Deck.h"
#include <vector>
#include "../dependance/json.hpp"
Graphe des dépendances par inclusion de testJson.cpp:
```



### Définitions de type

— using **json** = nlohmann::json

#### 5.1.1 Documentation des définitions de type

##### 5.1.1.1 json

```
using json = nlohmann::json
```





```

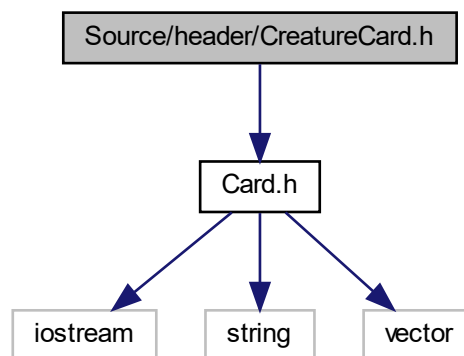
12 class Card {
13 protected:
14     std::string name;
15     std::vector<int> manaCost; // { Forest Green , Island Blue , Mountain Red, Plain Yellow, Swamp Black,
    Other }
16     std::string color;
17     bool isDiscarded = false;
18     bool isEngaged = false;
19     std::string typeOfCard;
20
21 public:
22     Card(std::string nm, std::vector<int> mnCt, std::string clr);
23     static std::vector<std::string> ColorCode;
24     void setName(const std::string &name);
25     void engage();
26
27     void disengage();
28     void setManaCost(std::vector<int> manaCost);
29     void setColor(const std::string &color);
30     void setIsDiscarded(bool isDiscarded);
31     const std::string &getName() const;
32     const std::string &getColoredName() const;
33     std::vector<int> &getManaCost() const;
34     const std::string &getColor() const;
35     bool &getIsDiscarded() const;
36     bool &getIsEngaged() const;
37     ~Card();
38     void isStillOperational();
39
40     virtual void print();
41     virtual void printLine(int line);
42     std::string manaToString();
43     static void print(std::vector<Card*> v);
44 };
45
46 #endif //MAGIC_CARD_H

```

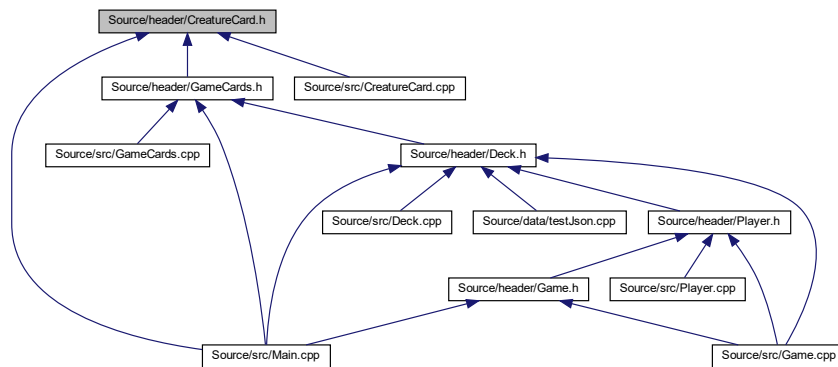
## 5.4 Référence du fichier Source/header/CreatureCard.h

#include "Card.h"

Graphe des dépendances par inclusion de CreatureCard.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **CreatureCard**

## 5.5 CreatureCard.h

Aller à la documentation de ce fichier.

```

1 //
2 // Created by bilai on 04/01/2022.
3 //
4
5 #ifndef MAGIC_CREATURECARD_H
6 #define MAGIC_CREATURECARD_H
7
8 #include "Card.h"
9
10 class CreatureCard : public Card {
11 public:
12     CreatureCard(std::string nam, std::vector<int> mC, std::string cl, int ap, int hp);
13     int getHp() const;
14     int getAttackPower() const;
15     void setHp(int hp);
16     void setAttackPower(int attackPower);
17     void invoke();
18     bool isFirstTurn();
19     void print();
20     void printLine(int line);
21     void attack();
22
23 protected:
24     int hp;
25     int defaultHp;
26     int attackPower;
27     bool firstTurn;
28 };
29
30 #endif //MAGIC_CREATURECARD_H

```

## 5.6 Référence du fichier Source/header/Deck.h

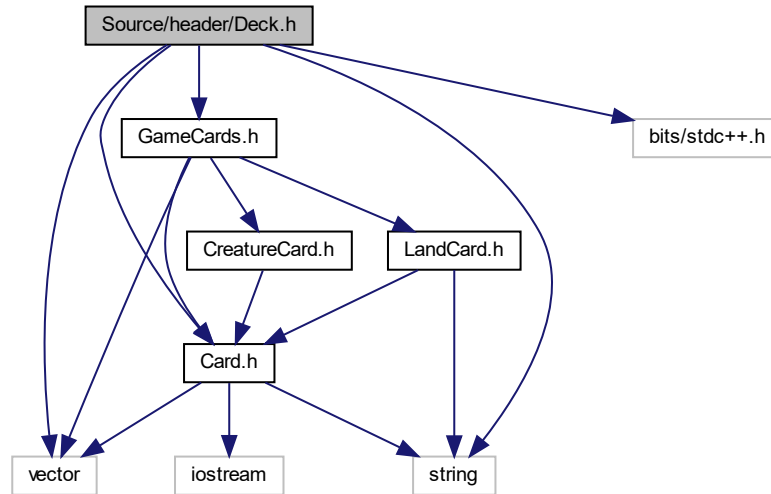
```

#include <vector>
#include <bits/stdc++.h>
#include "Card.h"

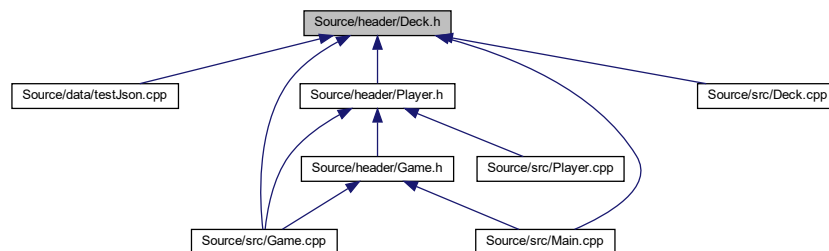
```

```
#include "GameCards.h"
#include <string>
```

Graphe des dépendances par inclusion de Deck.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **Deck**

## 5.7 Deck.h

**Aller à la documentation de ce fichier.**

```
1 //
2 // Created by bilai on 05/01/2022.
3 //
4
5 #ifndef MAGIC_DECK_H
6 #define MAGIC_DECK_H
7
8 #include <vector>
```

```

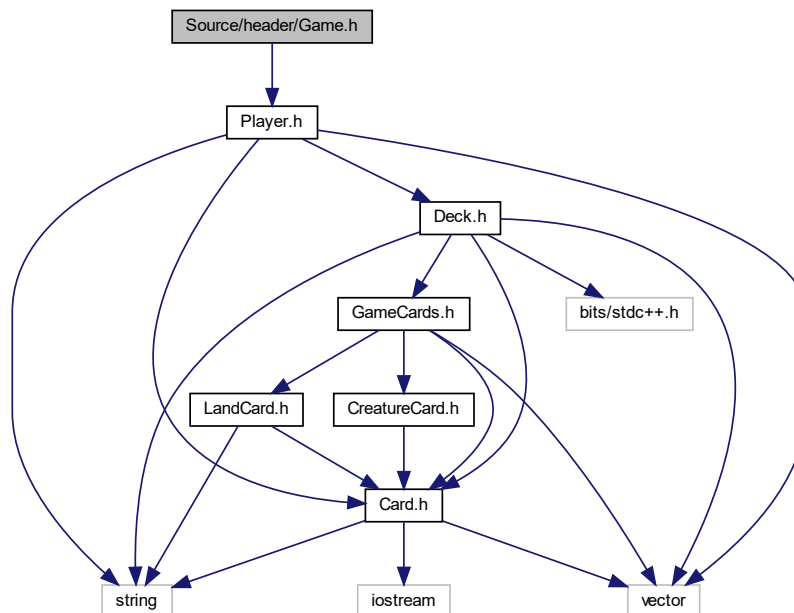
9 #include <bits/stdc++.h>
10 #include "Card.h"
11 #include "GameCards.h"
12 #include <string>
13
14 class Deck {
15 protected:
16     std::vector<Card*> inPlayCards;
17     std::vector<Card*> handCards;
18     std::vector<Card*> library;
19     std::vector<Card*> disCards;
20 public:
21     static const int DECK_SIZE;
22     Deck();
23     Deck(std::string nomDeck);
24     ~Deck();
25     void printLibrary();
26     void printInPlayCards();
27     void generateRandomDeck();
28     bool drawCard();
29     void disengageCards();
30     std::vector<Card*> getPlayableCards();
31     std::vector<Card*> getAttackCards();
32     std::vector<Card*> getDefenseCards();
33     void playCard(Card* c);
34     void discardCard(Card* c);
35     void JsonToDeck(std::string nomDeck);
36 };
37
38 #endif //MAGIC_DECK_H

```

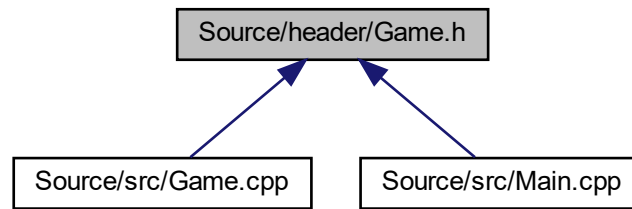
## 5.8 Référence du fichier Source/header/Game.h

#include "Player.h"

Graphe des dépendances par inclusion de Game.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **Game**

## 5.9 Game.h

**Aller à la documentation de ce fichier.**

```

1 //
2 // Created by bilai on 05/01/2022.
3 //
4
5 #ifndef MAGIC_GAME_H
6 #define MAGIC_GAME_H
7
8 #include "Player.h"
9
10
11 class Game {
12 private:
13     void showBanner();*
14     void initGame();
15     void randomDraw();
16     void mainPhase();
17     void fightPhase();
18     Player* getOpponent();
19
20 protected:
21     Player p1;
22     Player p2;
23     Player* playerTurn;
24     int round;
25
26 public :
27     Game();
28     //Game(Game const &); //reprendre une game
29     void playGame();
30     static int MAX_CARDS_IN_HAND;
31 };
32
33 #endif //MAGIC_GAME_H
  
```

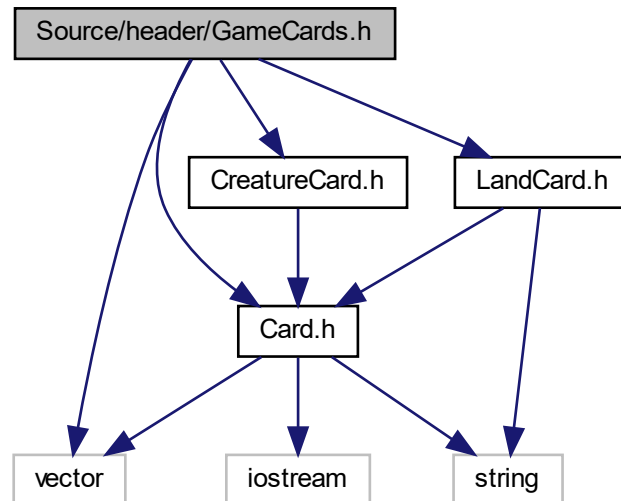
## 5.10 Référence du fichier Source/header/GameCards.h

```

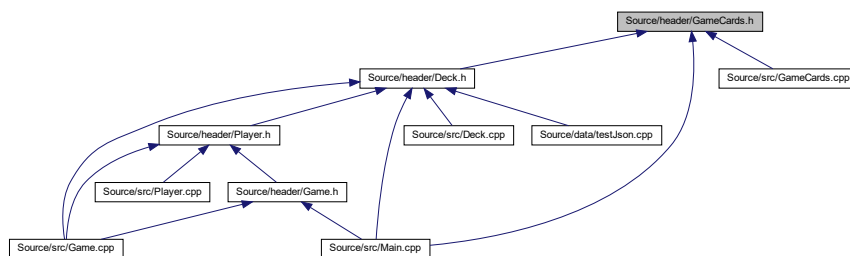
#include <vector>
#include "Card.h"
#include "CreatureCard.h"
  
```

```
#include "LandCard.h"
```

Graphe des dépendances par inclusion de GameCards.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **GameCards**

## 5.11 GameCards.h

**Aller à la documentation de ce fichier.**

```

1 //
2 // Created by dope on 08/01/2022.
3 //
4
5 #ifndef MAGIC_GAMECARDS_H
6 #define MAGIC_GAMECARDS_H
7

```

```

8 #include <vector>
9 #include "Card.h"
10 #include "CreatureCard.h"
11 #include "LandCard.h"
12
13 class GameCards {
14 private:
15     std::vector<CreatureCard> creatures;
16     std::vector<LandCard> lands;
17 public:
21     GameCards();
22     const std::vector<CreatureCard>& getCreatures();
23     const std::vector<LandCard>& getLands();
24 };
25
26
27 #endif //MAGIC_GAMECARDS_H

```

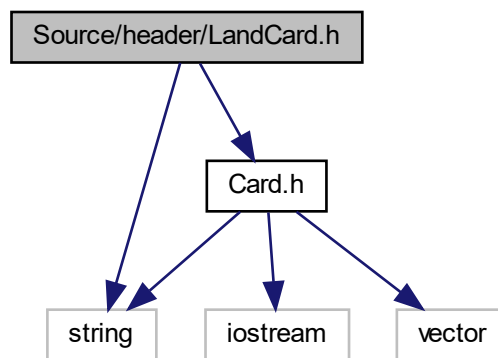
## 5.12 Référence du fichier Source/header/LandCard.h

```

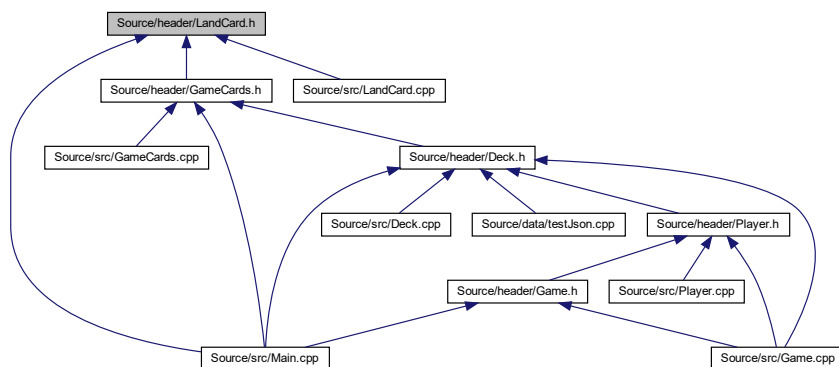
#include <string>
#include "Card.h"

```

Graphe des dépendances par inclusion de LandCard.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **LandCard**

### 5.13 LandCard.h

Aller à la documentation de ce fichier.

```

1 //
2 // Created by bilai on 05/01/2022.
3 //
4
5 #ifndef MAGIC_LANDCARD_H
6 #define MAGIC_LANDCARD_H
7
8 #include <string>
9
10 #include "Card.h"
11
12 class LandCard : public Card {
13 public :
14     LandCard(std::string nm, std::string clr);
15     ~LandCard(){};
16     void print();
17     void printLine(int line);
18 };
19
20 #endif //MAGIC_LANDCARD_H

```

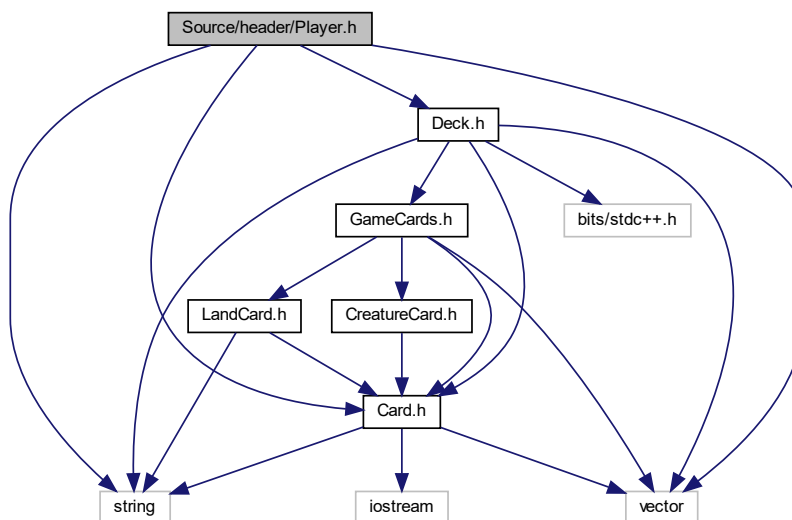
### 5.14 Référence du fichier Source/header/Player.h

```

#include <string>
#include <vector>
#include "Card.h"
#include "Deck.h"

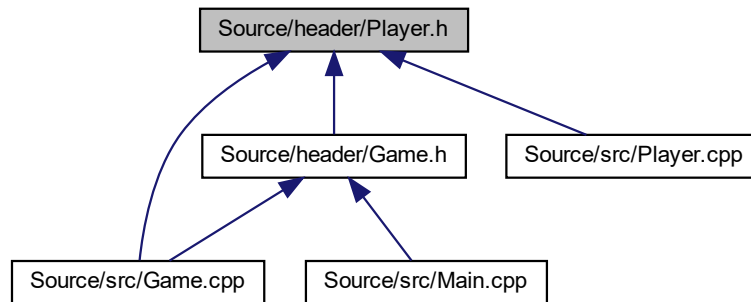
```

Graphe des dépendances par inclusion de Player.h:





Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **Player**

## 5.15 Player.h

**Aller à la documentation de ce fichier.**

```

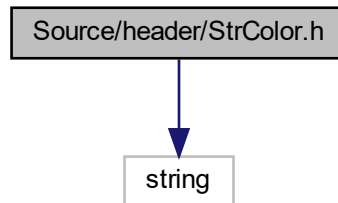
1 //
2 // Created by bilai on 05/01/2022.
3 //
4
5 #ifndef MAGIC_PLAYER_H
6 #define MAGIC_PLAYER_H
7
8 #include <string>
9 #include <vector>
10 #include "Card.h"
11 #include "Deck.h"
12
13
14 class Player {
15 protected :
16     std::string name;
17     std::string printColor;
18     bool isAlive;
19     int currentHp;
20     static int baseHp;
21     Deck cards;
22 public :
23     Player(std::string, Deck);
24     ~Player(){};
25
26     void setPrintColor(std::string color);
27     std::string getName();
28     std::string getColoredName();
29     int getHp();
30     void setHp(int hp);
31     bool drawCard();
32     void disengageCards();
33     std::vector<Card*> getPlayableCards();
34     void playCard(Card* c);
35     std::vector<Card*> getAttackCards();
36     std::vector<Card*> getDefenseCards();
37     void discardCard(Card* c);
38 };
39
40 #endif //MAGIC_PLAYER_H

```

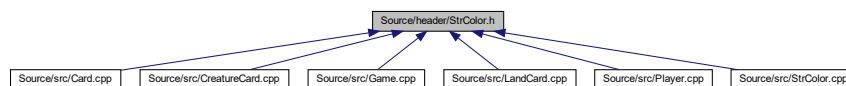
## 5.16 Référence du fichier Source/header/StrColor.h

```
#include <string>
```

Graphe des dépendances par inclusion de StrColor.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **StrColor**

## 5.17 StrColor.h

Aller à la documentation de ce fichier.

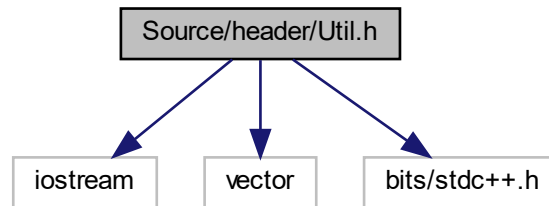
```

1 //
2 // Created by Rayane on 05/01/2022.
3 //
4
5 #ifndef MAGIC_STRCOLOR_H
6 #define MAGIC_STRCOLOR_H
7
8 #include <string>
9
10 class StrColor {
11 public:
12     static std::string print(std::string str, std::string color);
13     static std::string red(std::string);
14     static std::string green(std::string);
15     static std::string yellow(std::string);
16     static std::string blue(std::string);
17     static std::string magenta(std::string);
18     static std::string cyan(std::string);
19 };
20
21 #endif //MAGIC_STRCOLOR_H
  
```

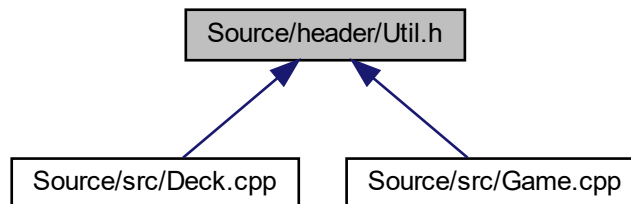
## 5.18 Référence du fichier Source/header/Util.h

```
#include <iostream>
#include <vector>
#include <bits/stdc++.h>
```

Graphe des dépendances par inclusion de Util.h:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Fonctions

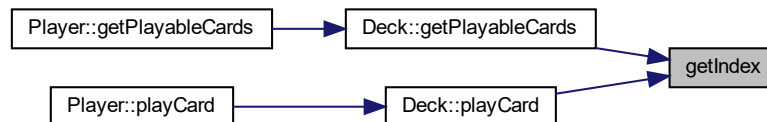
```
— template<typename Base , typename T >
  bool instanceof (const T *)
— template<typename T >
  int getIndex (std::vector< T > v, T K)
```

#### 5.18.1 Documentation des fonctions

### 5.18.1.1 getIndex()

```
template<typename T >
int getIndex (
    std::vector< T > v,
    T K )
```

Voici le graphe des appelants de cette fonction :



### 5.18.1.2 instanceof()

```
template<typename Base , typename T >
bool instanceof (
    const T * ) [inline]
```

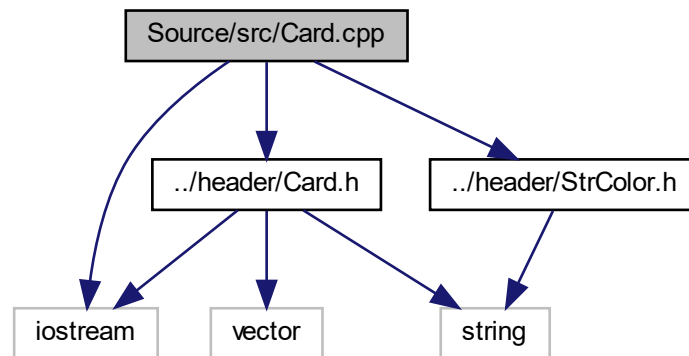
## 5.19 Util.h

**Aller à la documentation de ce fichier.**

```
1 //
2 // Created by dope on 12/01/2022.
3 //
4
5 #ifndef MAGIC_UTIL_H
6 #define MAGIC_UTIL_H
7
8 #include <iostream>
9 #include <vector>
10 #include <bits/stdc++.h>
11
12 template<typename Base, typename T>
13 inline bool instanceof(const T*) {
14     return std::is_base_of<Base, T>::value;
15 }
16
17 template<typename T>
18 int getIndex(std::vector<T> v, T K) {
19     auto it = find(v.begin(), v.end(), K);
20     if (it != v.end())
21         return it - v.begin();
22     else
23         return -1;
24 }
25
26 #endif //MAGIC_UTIL_H
```

## 5.20 Référence du fichier Source/src/Card.cpp

```
#include <iostream>
#include "../header/StrColor.h"
#include "../header/Card.h"
Graphe des dépendances par inclusion de Card.cpp:
```



## 5.21 Référence du fichier Source/src/CreatureCard.cpp

```
#include <iostream>
#include <string>
#include "../header/StrColor.h"
#include "../header/CreatureCard.h"
```



## 5.22.1 Documentation des définitions de type

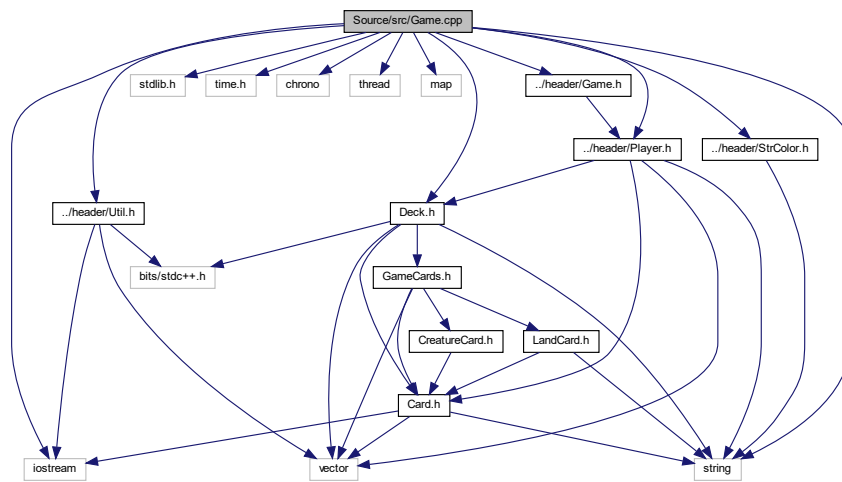
### 5.22.1.1 json

```
using json = nlohmann::json
```

## 5.23 Référence du fichier Source/src/Game.cpp

```
#include <string>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <chrono>
#include <thread>
#include <map>
#include "../header/Util.h"
#include "../header/Player.h"
#include "../header/Deck.h"
#include "../header/Game.h"
#include "../header/StrColor.h"
```

Graphe des dépendances par inclusion de Game.cpp:



## Fonctions

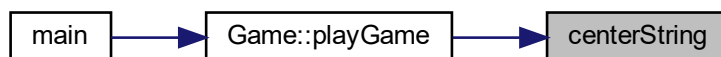
— `std::string centerString (std::string s, int size)`

### 5.23.1 Documentation des fonctions

### 5.23.1.1 centerString()

```
std::string centerString (  
    std::string s,  
    int size )
```

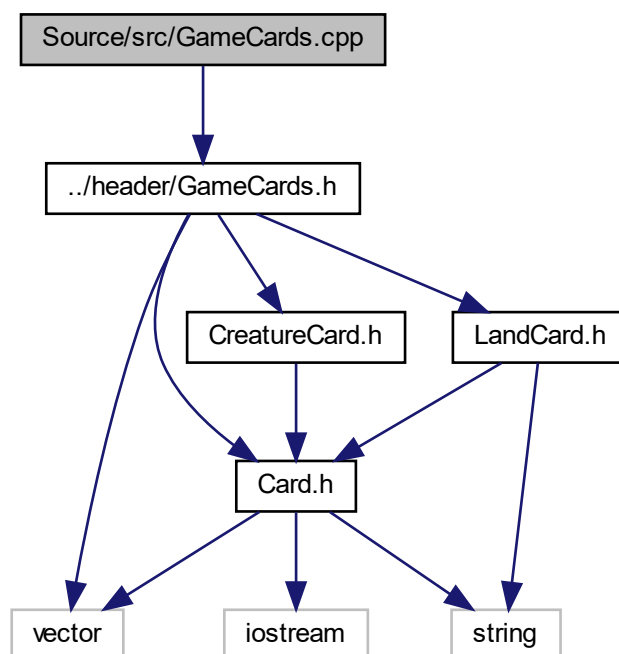
Voici le graphe des appelants de cette fonction :



## 5.24 Référence du fichier Source/src/GameCards.cpp

```
#include "../header/GameCards.h"
```

Graphe des dépendances par inclusion de GameCards.cpp:

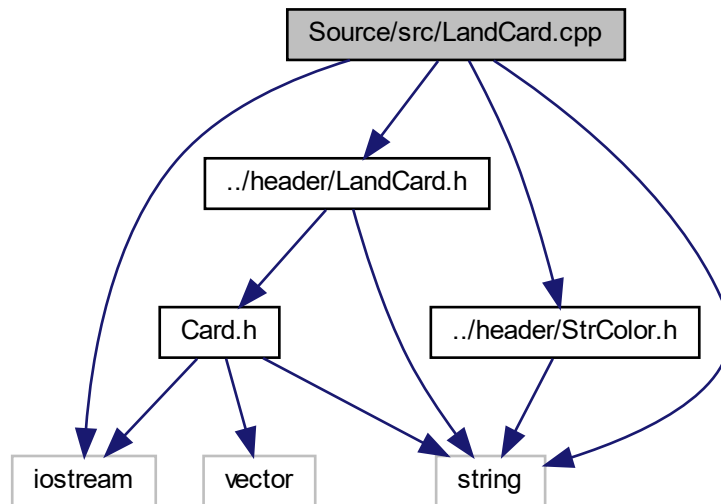




## 5.25 Référence du fichier Source/src/LandCard.cpp

```
#include <iostream>
#include <string>
#include "../header/StrColor.h"
#include "../header/LandCard.h"
```

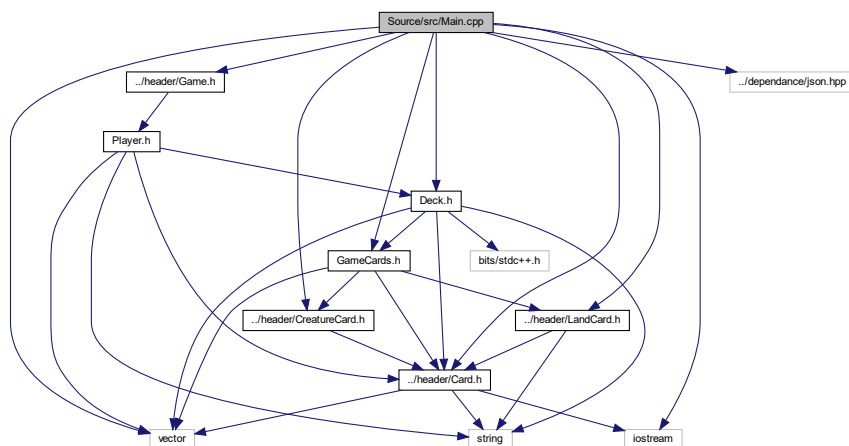
Graphe des dépendances par inclusion de LandCard.cpp:



## 5.26 Référence du fichier Source/src/Main.cpp

```
#include <vector>
#include <iostream>
#include "../header/Card.h"
#include "../header/CreatureCard.h"
#include "../header/LandCard.h"
#include "../header/Game.h"
#include "../header/Deck.h"
#include "../header/GameCards.h"
#include "../dependance/json.hpp"
```

Graphe des dépendances par inclusion de Main.cpp:



## Définitions de type

— using **json** = nlohmann::json

## Fonctions

— int **main** ()

### 5.26.1 Documentation des définitions de type

#### 5.26.1.1 json

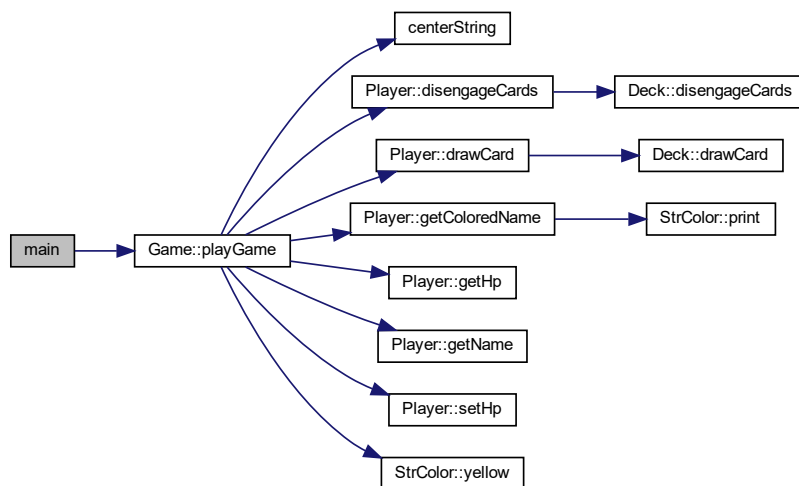
```
using json = nlohmann::json
```

#### 5.26.2 Documentation des fonctions

## 5.26.2.1 main()

```
int main ( )
```

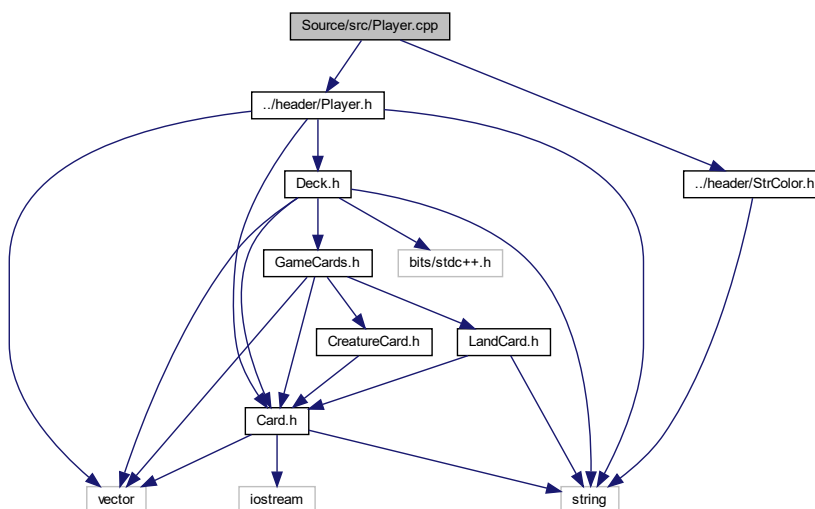
Voici le graphe d'appel pour cette fonction :



## 5.27 Référence du fichier Source/src/Player.cpp

```
#include "../header/Player.h"
#include "../header/StrColor.h"
```

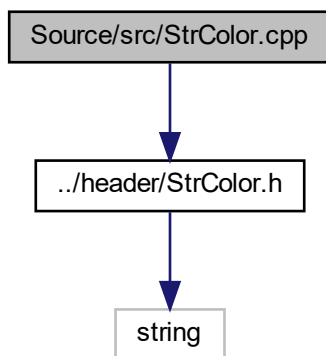
Graphe des dépendances par inclusion de Player.cpp:



## 5.28 Référence du fichier Source/src/StrColor.cpp

```
#include "../header/StrColor.h"
```

Graphe des dépendances par inclusion de StrColor.cpp:



# Index

- ~Card
  - Card, 11
- ~Deck
  - Deck, 27
- ~LandCard
  - LandCard, 41
- ~Player
  - Player, 44
- attack
  - CreatureCard, 20
- attackPower
  - CreatureCard, 23
- baseHp
  - Player, 51
- blue
  - StrColor, 52
- Card, 7
  - ~Card, 11
  - Card, 10
  - color, 17
  - ColorCode, 17
  - disengage, 11
  - engage, 11
  - getColor, 11
  - getColorName, 12
  - getIsDiscarded, 12
  - getIsEngaged, 12
  - getManaCost, 13
  - getName, 13
  - isDiscarded, 17
  - isEngaged, 17
  - isStillOperational, 13
  - manaCost, 17
  - manaToString, 13
  - name, 17
  - print, 14, 15
  - printLine, 15
  - setColor, 15
  - setIsDiscarded, 16
  - setManaCost, 16
  - setName, 16
  - typeOfCard, 17
- cards
  - Player, 51
- centerString
  - Game.cpp, 71
- color
  - Card, 17
- ColorCode
  - Card, 17
- CreatureCard, 18
  - attack, 20
  - attackPower, 23
  - CreatureCard, 20
  - defaultHp, 23
  - firstTurn, 24
  - getAttackPower, 21
  - getHp, 21
  - hp, 24
  - invoke, 21
  - isFirstTurn, 21
  - print, 22
  - printLine, 22
  - setAttackPower, 23
  - setHp, 23
- currentHp
  - Player, 51
- cyan
  - StrColor, 52
- Deck, 24
  - ~Deck, 27
  - Deck, 26
  - DECK\_SIZE, 32
  - discardCard, 27
  - disCards, 32
  - disengageCards, 28
  - drawCard, 28
  - generateRandomDeck, 28
  - getAttackCards, 28
  - getDefenseCards, 29
  - getPlayableCards, 29
  - handCards, 32
  - inPlayCards, 33
  - JsonToDeck, 30
  - library, 33
  - playCard, 31
  - printInPlayCards, 31
  - printLibrary, 32
- Deck.cpp
  - json, 71
- DECK\_SIZE
  - Deck, 32
- defaultHp
  - CreatureCard, 23
- discardCard
  - Deck, 27

- Player, 45
- disCards
  - Deck, 32
- disengage
  - Card, 11
- disengageCards
  - Deck, 28
  - Player, 45
- drawCard
  - Deck, 28
  - Player, 46
- engage
  - Card, 11
- firstTurn
  - CreatureCard, 24
- Game, 33
  - Game, 35
  - MAX\_CARDS\_IN\_HAND, 36
  - p1, 36
  - p2, 36
  - playerTurn, 36
  - playGame, 35
  - round, 36
- Game.cpp
  - centerString, 71
- GameCards, 37
  - GameCards, 37
  - getCreatures, 37
  - getLands, 38
- generateRandomDeck
  - Deck, 28
- getAttackCards
  - Deck, 28
  - Player, 46
- getAttackPower
  - CreatureCard, 21
- getColor
  - Card, 11
- getColoredName
  - Card, 12
  - Player, 47
- getCreatures
  - GameCards, 37
- getDefenseCards
  - Deck, 29
  - Player, 47
- getHp
  - CreatureCard, 21
  - Player, 48
- getIndex
  - Util.h, 67
- getIsDiscarded
  - Card, 12
- getIsEngaged
  - Card, 12
- getLands
  - GameCards, 38
- getManaCost
  - Card, 13
- getName
  - Card, 13
  - Player, 48
- getPlayableCards
  - Deck, 29
  - Player, 49
- green
  - StrColor, 53
- handCards
  - Deck, 32
- hp
  - CreatureCard, 24
- inPlayCards
  - Deck, 33
- instanceof
  - Util.h, 68
- invoke
  - CreatureCard, 21
- isAlive
  - Player, 51
- isDiscarded
  - Card, 17
- isEngaged
  - Card, 17
- isFirstTurn
  - CreatureCard, 21
- isStillOperational
  - Card, 13
- json
  - Deck.cpp, 71
  - Main.cpp, 74
  - testJson.cpp, 55
- JsonToDeck
  - Deck, 30
- LandCard, 38
  - ~LandCard, 41
  - LandCard, 41
  - print, 41
  - printLine, 41
- library
  - Deck, 33
- magenta
  - StrColor, 53
- main
  - Main.cpp, 74
- Main.cpp
  - json, 74
  - main, 74
- manaCost
  - Card, 17
- manaToString

- Card, 13
- MAX\_CARDS\_IN\_HAND
  - Game, 36
- name
  - Card, 17
  - Player, 51
- p1
  - Game, 36
- p2
  - Game, 36
- playCard
  - Deck, 31
  - Player, 49
- Player, 42
  - ~Player, 44
  - baseHp, 51
  - cards, 51
  - currentHp, 51
  - discardCard, 45
  - disengageCards, 45
  - drawCard, 46
  - getAttackCards, 46
  - getColoredName, 47
  - getDefenseCards, 47
  - getHp, 48
  - getName, 48
  - getPlayableCards, 49
  - isAlive, 51
  - name, 51
  - playCard, 49
  - Player, 44
  - printColor, 51
  - setHp, 50
  - setPrintColor, 50
- playerTurn
  - Game, 36
- playGame
  - Game, 35
- print
  - Card, 14, 15
  - CreatureCard, 22
  - LandCard, 41
  - StrColor, 53
- printColor
  - Player, 51
- printlnPlayCards
  - Deck, 31
- printLibrary
  - Deck, 32
- printLine
  - Card, 15
  - CreatureCard, 22
  - LandCard, 41
- red
  - StrColor, 53
- round
  - Game, 36
- setAttackPower
  - CreatureCard, 23
- setColor
  - Card, 15
- setHp
  - CreatureCard, 23
  - Player, 50
- setIsDiscarded
  - Card, 16
- setManaCost
  - Card, 16
- setName
  - Card, 16
- setPrintColor
  - Player, 50
- Source/data/testJson.cpp, 55
- Source/header/Card.h, 56
- Source/header/CreatureCard.h, 57, 58
- Source/header/Deck.h, 58, 59
- Source/header/Game.h, 60, 61
- Source/header/GameCards.h, 61, 62
- Source/header/LandCard.h, 63, 64
- Source/header/Player.h, 64, 65
- Source/header/StrColor.h, 66
- Source/header/Util.h, 67, 68
- Source/src/Card.cpp, 69
- Source/src/CreatureCard.cpp, 69
- Source/src/Deck.cpp, 70
- Source/src/Game.cpp, 71
- Source/src/GameCards.cpp, 72
- Source/src/LandCard.cpp, 73
- Source/src/Main.cpp, 73
- Source/src/Player.cpp, 75
- Source/src/StrColor.cpp, 76
- StrColor, 52
  - blue, 52
  - cyan, 52
  - green, 53
  - magenta, 53
  - print, 53
  - red, 53
  - yellow, 54
- testJson.cpp
  - json, 55
- typeofCard
  - Card, 17
- Util.h
  - getIndex, 67
  - instanceof, 68
- yellow
  - StrColor, 54