

Projet d'application : P4++

Généré par Doxygen 1.9.1



<b>1</b>	<b>ProjetApplicationConceptionLogicielle</b>	<b>1</b>
1.1	Fonctionnalités	1
1.1.1	Algorithmie	1
1.1.2	Interface graphique	1
1.1.3	Personnalisation	1
1.1.4	Bonus	2
<b>2</b>	<b>Liste des choses à faire</b>	<b>3</b>
<b>3</b>	<b>Liste des tests</b>	<b>5</b>
<b>4</b>	<b>Index des espaces de nommage</b>	<b>7</b>
4.1	Paquetages	7
<b>5</b>	<b>Documentation des espaces de nommage</b>	<b>9</b>
5.1	Référence de l'espace de nommage src	9
5.2	Référence de l'espace de nommage src.controller	9
5.3	Référence de l'espace de nommage src.controller.ctrl_main	9
5.3.1	Description détaillée	10
5.3.2	Documentation des fonctions	10
5.3.2.1	cm_ended_game()	10
5.3.2.2	cm_info()	10
5.3.2.3	cm_init()	10
5.3.2.4	cm_menu()	11
5.3.2.5	cm_page_accueil()	11
5.3.2.6	cm_page_bonus()	12
5.3.2.7	cm_page_parameters()	12
5.3.2.8	cm_page_play()	12
5.3.2.9	cm_quit()	13
5.3.2.10	cm_update()	13
5.3.2.11	cm_warning()	13
5.4	Référence de l'espace de nommage src.controller.ctrl_pageAccueil	14
5.4.1	Description détaillée	14
5.4.2	Documentation des fonctions	14
5.4.2.1	cpa_init()	14
5.4.2.2	cpa_play()	14
5.5	Référence de l'espace de nommage src.controller.ctrl_PageBonus	15
5.5.1	Description détaillée	15
5.5.2	Documentation des fonctions	15
5.5.2.1	cpb_get_bonuses()	15
5.5.2.2	cpb_get_chosen_bonus()	15
5.5.2.3	cpb_init()	16
5.5.2.4	cpb_show_bonus_description()	16
5.5.2.5	cpb_valider_bonus()	16

5.6 Référence de l'espace de nommage <code>src.controller.ctrl_pageJeu</code>	17
5.6.1 Description détaillée	17
5.6.2 Documentation des fonctions	17
5.6.2.1 <code>cpj_bot_play()</code>	17
5.6.2.2 <code>cpj_draw_grid()</code>	17
5.6.2.3 <code>cpj_info_turn()</code>	18
5.6.2.4 <code>cpj_init()</code>	18
5.6.2.5 <code>cpj_play()</code>	19
5.6.2.6 <code>cpj_put_coin()</code>	19
5.6.2.7 <code>cpj_quit()</code>	20
5.6.2.8 <code>cpj_redo()</code>	20
5.6.2.9 <code>cpj_undo()</code>	20
5.6.2.10 <code>cpj_update_grid()</code>	20
5.6.2.11 <code>cpj_use_bonus()</code>	21
5.7 Référence de l'espace de nommage <code>src.controller.ctrl_pageParametres</code>	21
5.7.1 Description détaillée	22
5.7.2 Documentation des fonctions	22
5.7.2.1 <code>cpp_askcolor()</code>	22
5.7.2.2 <code>cpp_custom_load()</code>	22
5.7.2.3 <code>cpp_custom_reset()</code>	22
5.7.2.4 <code>cpp_custom_save()</code>	23
5.7.2.5 <code>cpp_init()</code>	23
5.7.2.6 <code>cpp_settings_load()</code>	23
5.7.2.7 <code>cpp_settings_reset()</code>	24
5.7.2.8 <code>cpp_settings_save()</code>	24
5.8 Référence de l'espace de nommage <code>src.main</code>	24
5.8.1 Description détaillée	25
5.8.2 Documentation des fonctions	25
5.8.2.1 <code>main()</code>	25
5.9 Référence de l'espace de nommage <code>src.puissanceQuatre</code>	25
5.10 Référence de l'espace de nommage <code>src.puissanceQuatre.bonus</code>	25
5.10.1 Description détaillée	25
5.10.2 Documentation des fonctions	25
5.10.2.1 <code>p4b_flip_grid()</code>	25
5.10.2.2 <code>p4b_invert_grid()</code>	26
5.10.2.3 <code>p4b_no_bonus()</code>	26
5.10.2.4 <code>p4b_random_bonus()</code>	27
5.10.2.5 <code>p4b_random_placement()</code>	27
5.10.2.6 <code>p4b_remove_full_line()</code>	27
5.10.2.7 <code>p4b_use_min_max()</code>	28
5.11 Référence de l'espace de nommage <code>src.puissanceQuatre.gestionPartie</code>	28
5.11.1 Description détaillée	29

5.11.2 Documentation des fonctions	29
5.11.2.1 gp_choose_bonus()	29
5.11.2.2 gp_gestion_partie()	29
5.11.2.3 gp_get_player_choice()	29
5.11.2.4 gp_handle_bot_turn()	30
5.11.2.5 gp_handle_player_turn()	30
5.11.2.6 gp_handle_undo_redo()	31
5.11.2.7 gp_handle_victory()	31
5.11.2.8 gp_show_rules()	31
5.11.2.9 gp_use_bonus()	32
5.12 Référence de l'espace de nommage src.puissanceQuatre.grid	32
5.12.1 Description détaillée	32
5.12.2 Documentation des fonctions	32
5.12.2.1 pq_apply_gravity()	32
5.12.2.2 pq_init_grille()	33
5.12.2.3 pq_print_grille()	33
5.12.2.4 pq_reset_grille()	33
5.13 Référence de l'espace de nommage src.puissanceQuatre.puissanceQuatre	34
5.13.1 Description détaillée	34
5.13.2 Documentation des fonctions	34
5.13.2.1 pq_ajout_piece()	34
5.13.2.2 pq_minmax()	35
5.13.2.3 pq_partie_finie()	35
5.13.2.4 pq_redo()	36
5.13.2.5 pq_undo()	36
5.13.2.6 pq_verif_colonne()	36
5.13.2.7 pq_victoire()	37
5.13.2.8 pq_victoire_colonne()	37
5.13.2.9 pq_victoire_diago()	38
5.13.2.10 pq_victoire_ligne()	38
5.14 Référence de l'espace de nommage src.utils	39
5.15 Référence de l'espace de nommage src.utils.bonus_utils	39
5.15.1 Description détaillée	39
5.15.2 Documentation des fonctions	39
5.15.2.1 bu_format_bonus_name()	40
5.15.2.2 bu_get_bonus_description()	40
5.15.2.3 bu_get_bonus_name()	40
5.15.2.4 bu_get_bonuses()	40
5.15.2.5 bu_unformat_bonus_name()	40
5.16 Référence de l'espace de nommage src.utils.colors_utils	40
5.16.1 Description détaillée	41
5.16.2 Documentation des fonctions	41

5.16.2.1 cu_colors_too_close()	41
5.16.2.2 cu_hex_to_rgb()	41
5.16.2.3 cu_rgb_distance()	41
5.17 Référence de l'espace de nommage src.view	42
5.18 Référence de l'espace de nommage src.view.view_main	42
5.18.1 Description détaillée	42
5.18.2 Documentation des fonctions	42
5.18.2.1 vm_init()	42
5.18.2.2 vm_menu()	43
5.18.2.3 vm_message_game_ended()	43
5.18.2.4 vm_message_info()	43
5.18.2.5 vm_message_warning()	43
5.18.2.6 vm_quit()	44
5.18.2.7 vm_remove_frame()	44
5.18.2.8 vm_update()	44
5.19 Référence de l'espace de nommage src.view.view_pageAccueil	45
5.19.1 Description détaillée	45
5.19.2 Documentation des fonctions	45
5.19.2.1 vpa_destroy()	45
5.19.2.2 vpa_init()	45
5.20 Référence de l'espace de nommage src.view.view_pageBonus	45
5.20.1 Description détaillée	46
5.20.2 Documentation des fonctions	46
5.20.2.1 vpb_get_bonus()	46
5.20.2.2 vpb_get_frame()	46
5.20.2.3 vpb_init()	46
5.20.2.4 vpb_show_bonus_description()	47
5.21 Référence de l'espace de nommage src.view.view_pageJeu	47
5.21.1 Description détaillée	48
5.21.2 Documentation des fonctions	48
5.21.2.1 vpj_destroy()	48
5.21.2.2 vpj_disable_bonus()	48
5.21.2.3 vpj_draw_grid()	48
5.21.2.4 vpj_get_frame()	49
5.21.2.5 vpj_get_grid_cell()	49
5.21.2.6 vpj_init_page_jeu()	49
5.21.2.7 vpj_set_info()	50
5.21.2.8 vpj_show_coin()	50
5.22 Référence de l'espace de nommage src.view.view_pageParametres	50
5.22.1 Description détaillée	51
5.22.2 Documentation des fonctions	51
5.22.2.1 vpp_askcolor()	51

5.22.2.2 vpp_get_bot_color()	51
5.22.2.3 vpp_get_difficulty()	52
5.22.2.4 vpp_get_grid_color()	52
5.22.2.5 vpp_get_joueur_color()	52
5.22.2.6 vpp_get_nb_columns()	53
5.22.2.7 vpp_get_nb_jetons()	53
5.22.2.8 vpp_get_nb_rows()	53
5.22.2.9 vpp_init()	53
5.22.2.10 vpp_init_custom()	54
5.22.2.11 vpp_init_settings()	54
5.22.2.12 vpp_reset_customs()	55
5.22.2.13 vpp_reset_settings()	55
5.22.2.14 vpp_set_bot_color()	55
5.22.2.15 vpp_set_difficulty()	55
5.22.2.16 vpp_set_grid_color()	56
5.22.2.17 vpp_set_joueur_color()	56
5.22.2.18 vpp_set_nb_columns()	56
5.22.2.19 vpp_set_nb_jetons()	57
5.22.2.20 vpp_set_nb_rows()	57
5.23 Référence de l'espace de nommage tests	58
5.24 Référence de l'espace de nommage tests.test_grid	58
5.24.1 Description détaillée	58
5.24.2 Documentation des fonctions	58
5.24.2.1 tg_init_grille()	58
5.24.2.2 tg_test_all()	58
5.25 Référence de l'espace de nommage tests.test_puissanceQuatre	58
5.25.1 Description détaillée	59
5.25.2 Documentation des fonctions	59
5.25.2.1 tp_ajout_piece()	59
5.25.2.2 tp_test_all()	59
5.25.2.3 tp_verif_colonne()	59
5.25.2.4 tp_victoire_colonne()	60
5.25.2.5 tp_victoire_diago()	60
5.25.2.6 tp_victoire_ligne()	60





# Chapitre 1

## ProjetApplicationConceptionLogicielle

Lors de ce projet, nous devons réaliser un puissance 4. Cependant, ce puissance 4 sera amélioré, les joueurs pourront changer la taille de la grille et le nombre de jetons requis pour gagner.

De plus, le joueur aura des bonus permettant de renverser la partie ou de se donner un avantage, par exemple, il pourra avoir une bombe pour supprimer une ligne ou une colonne, ou alors retourner la grille. Nous allons aussi apporter de la personnalisation au joueur, il pourra changer la couleur des jetons et de la grille.

Ce projet se divise en trois parties : l'algorithmie, le génie logiciel ainsi que la réalisation de l'interface graphique. Ces trois parties sont essentielles à ce projet.

---

### 1.1 Fonctionnalités

#### 1.1.1 Algorithmie

- ☒ Vérification de la victoire
- ☒ Vérification de la jouabilité
- ☒ Algorithme MinMax
- ☒ Ajout de pièce
- ☒ Algorithme de gestion de partie en ligne de commande
- ☒ Implémentation du undo/redo

#### 1.1.2 Interface graphique

- ☒ Page d'accueil
- ☒ Page de paramètres
- ☒ Sauvegarde des paramètres
- ☒ Page de jeu
- ☒ Affichage de la grille
- ☒ Affichage des jetons
- ☒ Utilisation des bonus
- ☒ Affichage de la victoire/Défaite
- ☐ Affichage de l'aide
- ☒ Affichage du tour du joueur
- ☐ Interface responsive

#### 1.1.3 Personnalisation

- ☒ Personnalisation des jetons
- ☒ Personnalisation de la couleur de la grille
- ☒ Personnalisation de la taille de la grille
- ☒ Personnalisation du nombre de jetons à aligner pour gagner
- ☒ Personnalisation de la difficulté de l'IA

#### 1.1.4 Bonus

- [x] Utilisation des bonus en ligne de commande
- [x] Utilisation des bonus dans la fenêtre
- [x] Bonus d'inversion de la grille
- [x] Bonus de suppression de ligne pleine
- [x] Bonus d'aide avec MinMax
- [x] Bonus de retournement de la grille

## Chapitre 2

### Liste des choses à faire

Membre [src::controller::ctrl\\_pageJeu.cpj\\_use\\_bonus](#) (tk.Frame tkf\_page\_jeu)

Vérifier si le bonus a entraîné une victoire



## Chapitre 3

# Liste des tests

### Membre `tests::test_grid.tg_init_grille ()`

Vérifie que la grille est bien initialisée avec des 0 partout avec la bonne taille

Vérifie que toutes les combinaisons de 2 nombres de la liste *liste\_tailles* sont testées

### Membre `tests::test_puissanceQuatre.tp_ajout_piece ()`

Vérifie que la fonction renvoie les bonnes coordonnées quand le joueur joue

Vérifie que la fonction renvoie les bonnes coordonnées quand le bot joue

Vérifie que la fonction renvoie les bonnes coordonnées quand le joueur joue normalement

Vérifie que la fonction ne renvoie pas de coordonnées quand on ne peut pas ajouter de pièce

### Membre `tests::test_puissanceQuatre.tp_verif_colonne ()`

Vérifie que la fonction renvoie True si la colonne est vide

Vérifie que la fonction renvoie False si la colonne est pleine

Vérifie que la fonction renvoie True si la colonne est presque pleine

Vérifie que la fonction renvoie True si la colonne est un peu remplie

Vérifie que la fonction renvoie False si la grille est pleine

### Membre `tests::test_puissanceQuatre.tp_victoire_colonne ()`

Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne

### Membre `tests::test_puissanceQuatre.tp_victoire_diago ()`

Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne

### Membre `tests::test_puissanceQuatre.tp_victoire_ligne ()`

Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne



## Chapitre 4

# Index des espaces de nommage

### 4.1 Paquetages

Liste des paquetages avec une brève description (si disponible) :

<a href="#">src</a>	9
<a href="#">src.controller</a>	9
<a href="#">src.controller.ctrl_main</a>	
Un module qui gère la fenêtre principale du jeu	9
<a href="#">src.controller.ctrl_pageAccueil</a>	
Un module qui gère la page d'accueil du jeu	14
<a href="#">src.controller.ctrl_PageBonus</a>	
Un module qui gère la page de bonus	15
<a href="#">src.controller.ctrl_pageJeu</a>	
Un module qui gère la page de jeu	17
<a href="#">src.controller.ctrl_pageParametres</a>	
Un module qui gère la page des paramètres	21
<a href="#">src.main</a>	
Programme principal du jeu	24
<a href="#">src.puissanceQuatre</a>	25
<a href="#">src.puissanceQuatre.bonus</a>	
Ce module contient les fonctions relatives aux bonus	25
<a href="#">src.puissanceQuatre.gestionPartie</a>	
Un module qui gère la partie de puissance 4 en ligne de commande	28
<a href="#">src.puissanceQuatre.grid</a>	
Ce module contient les fonctions relatives à la grille de jeu	32
<a href="#">src.puissanceQuatre.puissanceQuatre</a>	
Ce module contient l'implémentation des règles du puissance 4	34
<a href="#">src.utils</a>	39
<a href="#">src.utils.bonus_utils</a>	
Ce module contient les fonctions utiles aux bonus	39
<a href="#">src.utils.colors_utils</a>	
Ce module contient les fonctions relatives aux couleurs	40
<a href="#">src.view</a>	42
<a href="#">src.view.view_main</a>	
Ce module contient les fonctions de base utile à la vue du jeu	42
<a href="#">src.view.view_pageAccueil</a>	
Ce module représente la vue de la page d'accueil	45
<a href="#">src.view.view_pageBonus</a>	
Ce module représente la vue de la page de choix du bonus	45
<a href="#">src.view.view_pageJeu</a>	
Ce module représente la vue de la page de jeu	47
<a href="#">src.view.view_pageParametres</a>	
Vue de la page des paramètres	50
<a href="#">tests</a>	58

<a href="#">tests.test_grid</a>	
Teste le module puissanceQuatre.grid . . . . .	58
<a href="#">tests.test_puissanceQuatre</a>	
Teste le module puissanceQuatre.puissanceQuatre . . . . .	58



## Chapitre 5

# Documentation des espaces de nommage

### 5.1 Référence de l'espace de nommage src

#### Espaces de nommage

- [controller](#)
- [main](#)  
*Programme principal du jeu.*
- [puissanceQuatre](#)
- [utils](#)
- [view](#)

### 5.2 Référence de l'espace de nommage src.controller

#### Espaces de nommage

- [ctrl\\_main](#)  
*Un module qui gère la fenêtre principale du jeu.*
- [ctrl\\_pageAccueil](#)  
*Un module qui gère la page d'accueil du jeu.*
- [ctrl\\_PageBonus](#)  
*Un module qui gère la page de bonus.*
- [ctrl\\_pageJeu](#)  
*Un module qui gère la page de jeu.*
- [ctrl\\_pageParametres](#)  
*Un module qui gère la page des paramètres.*

### 5.3 Référence de l'espace de nommage src.controller.ctrl\_main

Un module qui gère la fenêtre principale du jeu.

#### Fonctions

- `def cm\_init ()`  
*Initialise la fenêtre de jeu.*
- `def cm\_quit (tk.Tk tk_root)`  
*Ferme la fenêtre de jeu.*
- `tk.Menu cm\_menu (tk.Frame tk_frame, bool b_in_game)`  
*Crée le menu de la fenêtre.*
- `def cm\_page\_play (tk.Tk tk_root, tk.Frame tkf_old_frame)`  
*Fonction permettant de passer à la fenêtre de jeu.*
- `def cm\_page\_bonus (tk.Tk tk_root, tk.Frame tkf_old_frame)`  
*Fonction permettant de passer à la fenêtre de jeu.*
- `def cm\_page\_parameters (tk.Tk tk_root, tk.Frame tkf_old_frame)`  
*Fonction permettant de passer à la fenêtre de paramètres.*
- `def cm\_page\_accueil (tk.Tk tk_root, tk.Frame tkf_old_frame)`  
*Fonction permettant de passer à la fenêtre d'accueil.*
- `def cm\_ended\_game (str str_message, tk.Frame tkf_old_frame)`

- *Fonction permettant de passer à la fenêtre de fin de partie.*  
def `cm_warning` (str str\_message)
- *Affiche un message d'avertissement.*  
def `cm_info` (str str\_message)
- *Affiche un message d'information.*  
def `cm_update` (tk.Tk tk\_root)
- *Met à jour la fenêtre.*

### 5.3.1 Description détaillée

Un module qui gère la fenêtre principale du jeu.

Ce module gère la fenêtre principale du jeu et les transitions entre les différentes fenêtres.

### 5.3.2 Documentation des fonctions

#### 5.3.2.1 `cm_ended_game()`

```
def src.controller.ctrl_main.cm_ended_game (
    str str_message,
    tk.Frame tkf_old_frame )
```

Fonction permettant de passer à la fenêtre de fin de partie.

Cette fonction appelle la fonction de suppression du cadre de la dernière fenêtre et la fonction d'initialisation de la page de fin de partie. Elle permet de passer à la fenêtre de fin de partie.

##### Précondition

`tkf_old_frame` initialisé

##### Paramètres

<code>str_message</code>	Message à afficher
<code>tkf_old_frame</code>	Le cadre de la dernière fenêtre

#### 5.3.2.2 `cm_info()`

```
def src.controller.ctrl_main.cm_info (
    str str_message )
```

Affiche un message d'information.

Cette fonction appelle la fonction d'affichage d'un message d'information.

##### Paramètres

<code>str_message</code>	Message à afficher
--------------------------	--------------------

#### 5.3.2.3 `cm_init()`

```
def src.controller.ctrl_main.cm_init ( )
```

Initialise la fenêtre de jeu.

Cette fonction initialise la fenêtre de jeu et lance la boucle principale.

##### Précondition

`tk_root` initialisé

**Postcondition**

boucle principale lancée

**Variables :**

— tk\_root : Fenêtre principale

**5.3.2.4 cm\_menu()**

```
tk.Menu src.controller.ctrl_main.cm_menu (
    tk.Frame tk_frame,
    bool b_in_game )
```

Crée le menu de la fenêtre.

Cette fonction appelle la fonction de création du menu de la fenêtre. Elle renvoie le menu créé. Elle prend en paramètre le cadre de la dernière fenêtre et un booléen indiquant si le joueur est en jeu ou non.

**Précondition**

tkf\_old\_frame initialisé

**Paramètres**

<i>tk_frame</i>	Le cadre de la dernière fenêtre
<i>b_in_game</i>	Booléen indiquant si le joueur est en jeu ou non

**Renvoie**

Menu de la fenêtre

**Postcondition**

Menu créé

**5.3.2.5 cm\_page\_accueil()**

```
def src.controller.ctrl_main.cm_page_accueil (
    tk.Tk tk_root,
    tk.Frame tkf_old_frame )
```

Fonction permettant de passer à la fenêtre d'accueil.

Cette fonction appelle la fonction de suppression du cadre de la dernière fenêtre et la fonction d'initialisation de la page d'accueil. Elle permet de passer à la fenêtre d'accueil.

**Précondition**

tk\_root initialisé

tkf\_old\_frame initialisé

**Paramètres**

<i>tk_root</i>	La fenêtre principale
<i>tkf_old_frame</i>	Le cadre de la dernière fenêtre

**Postcondition**

tkf\_old\_frame détruit  
fenêtre d'accueil initialisée

**5.3.2.6 cm\_page\_bonus()**

```
def src.controller.ctrl_main.cm_page_bonus (
    tk.Tk tk_root,
    tk.Frame tkf_old_frame )
```

Fonction permettant de passer à la fenêtre de jeu.

Cette fonction appelle la fonction de suppression du cadre de la dernière fenêtre et la fonction d'initialisation de la page de jeu. Elle permet de passer à la fenêtre de jeu.

**Précondition**

tk\_root initialisé  
tkf\_old\_frame initialisé

**Paramètres**

<i>tk_root</i>	La fenêtre principale
<i>tkf_old_frame</i>	Le cadre de la dernière fenêtre

**Postcondition**

tkf\_old\_frame détruit  
fenêtre de jeu initialisée

**5.3.2.7 cm\_page\_parameters()**

```
def src.controller.ctrl_main.cm_page_parameters (
    tk.Tk tk_root,
    tk.Frame tkf_old_frame )
```

Fonction permettant de passer à la fenêtre de paramètres.

Cette fonction appelle la fonction de suppression du cadre de la dernière fenêtre et la fonction d'initialisation de la page de paramètres. Elle permet de passer à la fenêtre de paramètres.

**Précondition**

tk\_root initialisé  
tkf\_old\_frame initialisé

**Paramètres**

<i>tk_root</i>	La fenêtre principale
<i>tkf_old_frame</i>	Le cadre de la dernière fenêtre

**Postcondition**

tkf\_old\_frame détruit  
fenêtre de paramètres initialisée

**5.3.2.8 cm\_page\_play()**

```
def src.controller.ctrl_main.cm_page_play (
```

```
tk.Tk tk_root,
tk.Frame tkf_old_frame )
```

Fonction permettant de passer à la fenêtre de jeu.

Cette fonction appelle la fonction de suppression du cadre de la dernière fenêtre et la fonction d'initialisation de la page de jeu. Elle permet de passer à la fenêtre de jeu.

#### Précondition

tk\_root initialisé  
tkf\_old\_frame initialisé

#### Paramètres

<i>tk_root</i>	La fenêtre principale
<i>tkf_old_frame</i>	Le cadre de la dernière fenêtre

#### Postcondition

tkf\_old\_frame détruit  
fenêtre de jeu initialisée

#### 5.3.2.9 `cm_quit()`

```
def src.controller.ctrl_main.cm_quit (
    tk.Tk tk_root )
```

Ferme la fenêtre de jeu.

Cette fonction appelle la fonction de fermeture de la fenêtre de jeu.

#### Précondition

tk\_root initialisé

#### Paramètres

<i>tk_root</i>	Fenêtre principale
----------------	--------------------

#### 5.3.2.10 `cm_update()`

```
def src.controller.ctrl_main.cm_update (
    tk.Tk tk_root )
```

Met à jour la fenêtre.

Cette fonction appelle la fonction de mise à jour de la fenêtre principale.

#### Précondition

tk\_root initialisé

#### Paramètres

<i>tk_root</i>	Fenêtre principale
----------------	--------------------

#### 5.3.2.11 `cm_warning()`

```
def src.controller.ctrl_main.cm_warning (
    str str_message )
```

Affiche un message d'avertissement.

Cette fonction appelle la fonction d'affichage d'un message d'avertissement.

#### Paramètres

<code>str_message</code>	Message à afficher
--------------------------	--------------------

## 5.4 Référence de l'espace de nommage `src.controller.ctrl_pageAccueil`

Un module qui gère la page d'accueil du jeu.

### Fonctions

- `def cpa_init (tk.Tk tk_root)`  
*Initialise la page d'accueil.*
- `def cpa_play (tk.Tk tk_root, tk.Frame tkf_frame)`  
*Lance une partie.*

#### 5.4.1 Description détaillée

Un module qui gère la page d'accueil du jeu.

Ce module gère la page d'accueil du jeu

#### 5.4.2 Documentation des fonctions

##### 5.4.2.1 `cpa_init()`

```
def src.controller.ctrl_pageAccueil.cpa_init (
    tk.Tk tk_root )
```

Initialise la page d'accueil.

Cette fonction initialise la page d'accueil du jeu.

#### Précondition

`tk_root` initialisé

#### Paramètres

<code>tk_root</code>	Fenêtre principale
----------------------	--------------------

##### 5.4.2.2 `cpa_play()`

```
def src.controller.ctrl_pageAccueil.cpa_play (
    tk.Tk tk_root,
    tk.Frame tkf_frame )
```

Lance une partie.

Cette fonction lance une partie de puissance 4++. Elle est appelée lorsque l'utilisateur clique sur le bouton "Jouer".

#### Précondition

`tk_root` initialisé

`tkf_frame` initialisé

## Paramètres

<code>tk_root</code>	Fenêtre principale
<code>tkf_frame</code>	Frame de la page d'accueil

## Postcondition

Fenêtre de choix du bonus ouverte

## 5.5 Référence de l'espace de nommage src.controller.ctrl\_PageBonus

Un module qui gère la page de bonus.

## Fonctions

- `def cpb_init (tk.Tk tk_win_root)`  
*Initialise la page de choix du bonus.*
- `list[str] cpb_get_bonuses ()`  
*Récupère les bonus disponibles.*
- `def cpb_valider_bonus ()`  
*Récupère les bonus sélectionnés par le joueur.*
- `def cpb_show_bonus_description (str s_bonus)`  
*Affiche la description du bonus sélectionné.*
- `str cpb_get_chosen_bonus ()`  
*Accesseur du bonus choisi par l'utilisateur.*

### 5.5.1 Description détaillée

Un module qui gère la page de bonus.

Ce module contient les fonctions permettant de gérer la page de choix du bonus.

### 5.5.2 Documentation des fonctions

#### 5.5.2.1 cpb\_get\_bonuses()

```
list[str] src.controller.ctrl_PageBonus.cpb_get_bonuses ( )
```

Récupère les bonus disponibles.

Cette fonction récupère les bonus disponibles et les formate pour l'affichage.

## Renvoie

: Liste des bonus disponibles

#### 5.5.2.2 cpb\_get\_chosen\_bonus()

```
str src.controller.ctrl_PageBonus.cpb_get_chosen_bonus ( )
```

Accesseur du bonus choisi par l'utilisateur.

Cette fonction retourne le bonus choisi par l'utilisateur.

## Précondition

S\_BONUS non nul

## Renvoie

: Bonus choisi par l'utilisateur

## Variables :

- S\_BONUS : nom du bonus sélectionné

### 5.5.2.3 `cpb_init()`

```
def src.controller.ctrl_PageBonus.cpb_init (
    tk.Tk tk_win_root )
```

Initialise la page de choix du bonus.

Cette fonction initialise la page de choix du bonus en affichant la fenêtre principale et en initialisant la page de choix du bonus.

#### Précondition

tk\_root initialisé

#### Paramètres

<code>tk_win_root</code>	Fenêtre principale
--------------------------	--------------------

#### Postcondition

page de choix du bonus initialisée

#### Variables :

- NPA\_GRID : Grille de jeu
- TK\_ROOT : Fenêtre principale

### 5.5.2.4 `cpb_show_bonus_description()`

```
def src.controller.ctrl_PageBonus.cpb_show_bonus_description (
    str s_bonus )
```

Affiche la description du bonus sélectionné.

Cette fonction affiche la description du bonus sélectionné. Pour cela, elle récupère la description du bonus et l'affiche.

#### Précondition

s\_bonus non vide

#### Paramètres

<code>s_bonus</code>	Nom du bonus
----------------------	--------------

#### Postcondition

Description du bonus affichée

#### Variables :

- s\_desc : Description du bonus

### 5.5.2.5 `cpb_valider_bonus()`

```
def src.controller.ctrl_PageBonus.cpb_valider_bonus ( )
```

Récupère les bonus sélectionnés par le joueur.

Cette fonction récupère les bonus sélectionnés par le joueur et passe à la fenêtre de jeu.

#### Précondition

fenêtre de choix du bonus affichée

#### Postcondition

Choix du bonus enregistré

#### Variables :

- S\_BONUS : Nom du bonus sélectionné
- TK\_ROOT : Fenêtre principale



## 5.6 Référence de l'espace de nommage src.controller.ctrl\_pageJeu

Un module qui gère la page de jeu.

### Fonctions

- def `cpj_init` (tk.Tk tk\_win\_root)  
*Affiche la page de jeu.*
- def `cpj_draw_grid` (int i\_nb\_rows, int i\_nb\_columns)  
*Dessine la grille de jeu.*
- def `cpj_put_coin` (int i\_row, int i\_cols, int i\_joueur)  
*Place un jeton dans la grille de jeu.*
- def `cpj_undo` ()  
*Annule le dernier coup.*
- def `cpj_redo` ()  
*Refait le dernier coup.*
- def `cpj_quit` ()  
*Quitte la partie.*
- def `cpj_play` (tk.Event event, tk.Frame tkf\_page\_jeu)  
*Joue un coup dans la grille de jeu.*
- def `cpj_use_bonus` (tk.Frame tkf\_page\_jeu)  
*Utilise un bonus puis met à jour la grille de jeu.*
- def `cpj_bot_play` (tk.Frame tkf\_page\_jeu)  
*Fait jouer le bot.*
- def `cpj_update_grid` ()  
*Réinitialise la grille de jeu.*
- def `cpj_info_turn` (bool b\_is\_player)  
*Affiche le joueur qui doit jouer.*

### 5.6.1 Description détaillée

Un module qui gère la page de jeu.

Ce module contient les fonctions permettant de gérer la page de jeu.

### 5.6.2 Documentation des fonctions

#### 5.6.2.1 `cpj_bot_play()`

```
def src.controller.ctrl_pageJeu.cpj_bot_play (
    tk.Frame tkf_page_jeu )
```

Fait jouer le bot.

Cette fonction fait jouer le bot puis met à jour la grille de jeu.

#### Précondition

tkf\_page\_jeu initialisé

#### Paramètres

<code>tkf_page_jeu</code>	: Frame de la page de jeu
---------------------------	---------------------------

#### Variables :

- `I_NB_JETONS` : Nombre de jetons à aligner pour gagner
- `I_DIFFICULTY` : Difficulté du bot
- `i_grid_x` : Colonne de la grille de jeu
- `i_grid_y` : Ligne de la grille de jeu

#### 5.6.2.2 `cpj_draw_grid()`

```
def src.controller.ctrl_pageJeu.cpj_draw_grid (
```

```
int i_nb_rows,
int i_nb_columns )
```

Dessine la grille de jeu.

Cette fonction appelle la fonction de la vue permettant de dessiner la grille de jeu.

#### Précondition

```
i_nb_rows > 0
i_nb_columns > 0
```

#### Paramètres

<i>i_nb_rows</i>	Nombre de lignes de la grille de jeu
<i>i_nb_columns</i>	Nombre de colonnes de la grille de jeu

#### Postcondition

grille de jeu dessinée

### 5.6.2.3 cpj\_info\_turn()

```
def src.controller.ctrl_pageJeu.cpj_info_turn (
    bool b_is_player )
```

Affiche le joueur qui doit jouer.

Cette fonction affiche le joueur qui doit jouer.

#### Paramètres

<i>b_is_player</i>	: Booléen indiquant si c'est au joueur de jouer
--------------------	---

#### Postcondition

Affichage du joueur qui doit jouer

#### Variables :

— *b\_is\_player* : Booléen indiquant si c'est au joueur de jouer

### 5.6.2.4 cpj\_init()

```
def src.controller.ctrl_pageJeu.cpj_init (
    tk.Tk tk_win_root )
```

Affiche la page de jeu.

Cette fonction affiche la page de jeu et initialise la grille de jeu. Elle récupère également les paramètres de la partie. Elle initialise également les variables globales.

#### Précondition

tk\_root initialisé

#### Postcondition

page de jeu affichée

#### Variables :

- TK\_ROOT : Fenêtre principale
- ST\_COLOR\_JOUEUR : Couleur des jetons du joueur
- ST\_COLOR\_BOT : Couleur des jetons du bot
- I\_NB\_ROWS : Nombre de lignes de la grille de jeu
- I\_NB\_COLS : Nombre de colonnes de la grille de jeu
- NPA\_GRID : Grille de jeu

- T\_UNDO\_REDO : Liste des coups joués
- T\_REDO : Liste des coups annulés
- I\_DIFFICULTY : Difficulté du bot
- I\_NB\_JETONS : Nombre de jetons à aligner pour gagner
- st\_color\_grid : Couleur de la grille de jeu

#### 5.6.2.5 cpj\_play()

```
def src.controller.ctrl_pageJeu.cpj_play (
    tk.Event event,
    tk.Frame tkf_page_jeu )
```

Joue un coup dans la grille de jeu.

Cette fonction joue un coup dans la grille de jeu et met à jour la grille de jeu.

##### Précondition

event est un évènement de la souris  
tkf\_page\_jeu initialisé

##### Paramètres

<i>event</i>	Évènement de la souris sur la grille de jeu
<i>tkf_page_jeu</i>	: Frame de la page de jeu

##### Variables :

- I\_NB\_JETONS : Nombre de jetons à aligner pour gagner
- i\_grid\_x : Colonne de la grille de jeu
- i\_grid\_y : Ligne de la grille de jeu
- b\_joueur\_gagne : Booléen indiquant si le joueur a gagné
- b\_joueur\_joue : Booléen indiquant si le joueur a joué

#### 5.6.2.6 cpj\_put\_coin()

```
def src.controller.ctrl_pageJeu.cpj_put_coin (
    int i_row,
    int i_cols,
    int i_joueur )
```

Place un jeton dans la grille de jeu.

Cette fonction appelle la fonction de la vue permettant de placer un jeton dans la grille de jeu.

##### Précondition

i\_row >= 0  
i\_cols >= 0  
i\_joueur >= 1 et i\_joueur <= 2

##### Paramètres

<i>i_row</i>	Ligne de la grille de jeu
<i>i_cols</i>	Colonne de la grille de jeu
<i>i_joueur</i>	Joueur qui joue

##### Postcondition

jeton placé dans la grille de jeu

##### Variables :

- ST\_COLOR\_JOUEUR : Couleur des jetons du joueur
- ST\_COLOR\_BOT : Couleur des jetons du bot

#### 5.6.2.7 cpj\_quit()

```
def src.controller.ctrl_pageJeu.cpj_quit ( )
```

Quitte la partie.

Cette fonction détruit la page de jeu et ferme la fenêtre principale.

##### Précondition

tk\_root initialisé

##### Postcondition

page de jeu détruite

#### 5.6.2.8 cpj\_redo()

```
def src.controller.ctrl_pageJeu.cpj_redo ( )
```

Refait le dernier coup.

Cette fonction refait le dernier coup annulé et met à jour la grille de jeu.

##### Précondition

tk\_root initialisé

##### Postcondition

dernier coup refait

#### 5.6.2.9 cpj\_undo()

```
def src.controller.ctrl_pageJeu.cpj_undo ( )
```

Annule le dernier coup.

Cette fonction annule le dernier coup joué et met à jour la grille de jeu.

##### Précondition

tk\_root initialisé

##### Postcondition

dernier coup annulé

#### 5.6.2.10 cpj\_update\_grid()

```
def src.controller.ctrl_pageJeu.cpj_update_grid ( )
```

Réinitialise la grille de jeu.

Cette fonction réinitialise la grille de jeu puis met à jour la grille de jeu. Elle permet de mettre à jour la grille de jeu après un undo ou un redo.

##### Précondition

NPA\_GRID initialisé

**Postcondition**

Grille de jeu mise à jour

**Variables :**

- I\_NB\_ROWS : Nombre de lignes de la grille de jeu
- I\_NB\_COLS : Nombre de colonnes de la grille de jeu
- NPA\_GRID : Grille de jeu
- i\_boucle\_row : Ligne de la grille de jeu
- i\_boucle\_col : Colonne de la grille de jeu

**5.6.2.11 cpj\_use\_bonus()**

```
def src.controller.ctrl_pageJeu.cpj_use_bonus (
    tk.Frame tkf_page_jeu )
```

Utilise un bonus puis met à jour la grille de jeu.

Cette fonction utilise un bonus puis met à jour la grille de jeu.

**Précondition**

S\_BONUS est un bonus

NPA\_GRID est une grille de jeu

**Paramètres**

<i>tkf_page_jeu</i>	: Frame de la page de jeu
---------------------	---------------------------

**Postcondition**

Bonus utilisé si B\_BONUS\_USED est faux

**Variables :**

- S\_BONUS : Bonus sélectionné
- NPA\_GRID : Grille de jeu
- B\_BONUS\_USED : Booléen indiquant si le bonus a été utilisé
- m\_module : Module du bonus
- f\_bonus : Fonction du bonus
- **A faire** Vérifier si le bonus a entraîné une victoire

## 5.7 Référence de l'espace de nommage src.controller.ctrl\_pageParametres

Un module qui gère la page des paramètres.

**Fonctions**

- def `cpp_init` (tk.Tk tk\_win\_root)  
*Initialise la page des paramètres.*
- def `cpp_settings_save` ()  
*Sauvegarde les paramètres.*
- def `cpp_settings_reset` ()  
*Réinitialise les paramètres.*
- def `cpp_custom_save` ()  
*Sauvegarde des paramètres de personnalisation.*
- (int, int, int, int) `cpp_settings_load` ()  
*Charge les paramètres.*
- (str, str, str) `cpp_custom_load` ()  
*Charge les paramètres de personnalisation.*
- def `cpp_askcolor` (str s\_element)  
*Ouvre un sélecteur de couleur.*
- def `cpp_custom_reset` ()  
*Réinitialise les paramètres de personnalisation.*

### 5.7.1 Description détaillée

Un module qui gère la page des paramètres.

Contient les fonctions permettant de gérer la page des paramètres

### 5.7.2 Documentation des fonctions

#### 5.7.2.1 `cpp_askcolor()`

```
def src.controller.ctrl_pageParametres.cpp_askcolor (
    str s_element )
```

Ouvre un sélecteur de couleur.

Cette fonction ouvre un sélecteur de couleur pour l'élément passé en paramètre. L'élément peut être le joueur, le bot ou la grille.

##### Précondition

tk\_root initialisé  
view\_pp initialisé

##### Paramètres

<code>s_element</code>	Élément à colorer (joueur, bot ou grille)
------------------------	---

#### 5.7.2.2 `cpp_custom_load()`

```
(str, str, str) src.controller.ctrl_pageParametres.cpp_custom_load ( )
```

Charge les paramètres de personnalisation.

Cette fonction charge les paramètres de personnalisation sauvegardés dans un fichier texte. Elle récupère la couleur des jetons du joueur, la couleur des jetons du bot et la couleur de la grille. Si le fichier n'existe pas, les paramètres par défaut sont retournés.

##### Précondition

res/custom.txt existant

##### Renvoie

st\_color\_joueur: Couleur des jetons du joueur  
st\_color\_bot: Couleur des jetons du bot  
st\_color\_grid: Couleur de la grille

##### Variables :

- st\_color\_joueur : Couleur des jetons du joueur
- st\_color\_bot : Couleur des jetons du bot
- st\_color\_grid : Couleur de la grille

#### 5.7.2.3 `cpp_custom_reset()`

```
def src.controller.ctrl_pageParametres.cpp_custom_reset ( )
```

Réinitialise les paramètres de personnalisation.

Cette fonction réinitialise les paramètres de personnalisation par défaut.

##### Précondition

tk\_root initialisé  
view\_pp initialisé

#### 5.7.2.4 `cpp_custom_save()`

```
def src.controller.ctrl_pageParametres.cpp_custom_save ( )
```

Sauvegarde des paramètres de personnalisation.

Cette fonction sauvegarde les paramètres de personnalisation dans un fichier texte. Si les paramètres ne sont pas valides, un message d'erreur est affiché. Les paramètres sont valides si deux couleurs ne sont pas trop proches. Deux couleurs sont trop proches si la différence entre les composantes rouges, vertes et bleues est inférieure à 50.

##### Précondition

tk\_root initialisé  
res/custom.txt existant

##### Postcondition

paramètres de personnalisation sauvegardés ou str\_message d'erreur affiché

##### Variables :

- st\_color\_joueur : Couleur des jetons du joueur
- st\_color\_bot : Couleur des jetons du bot
- st\_color\_grid : Couleur de la grille
- str\_message : Message d'information
- f\_custom : Fichier de sauvegarde des paramètres de personnalisation

#### 5.7.2.5 `cpp_init()`

```
def src.controller.ctrl_pageParametres.cpp_init (
    tk.Tk tk_win_root )
```

Initialise la page des paramètres.

Cette fonction initialise la page des paramètres en chargeant les paramètres sauvegardés et en les affichant.

##### Précondition

tk\_root initialisé

##### Paramètres

<code>tk_win_root</code>	Fenêtre principale
--------------------------	--------------------

##### Postcondition

page des paramètres initialisée

##### Variables :

- tk\_root : Fenêtre principale
- i\_rows : Nombre de lignes de la grille
- i\_columns : Nombre de colonnes de la grille
- i\_nb\_jetons : Nombre de jetons à aligner pour gagner
- i\_difficulty : Difficulté du bot
- st\_color\_joueur : Couleur des jetons du joueur
- st\_color\_bot : Couleur des jetons du bot
- st\_color\_grid : Couleur de la grille

#### 5.7.2.6 `cpp_settings_load()`

```
(int, int, int, int) src.controller.ctrl_pageParametres.cpp_settings_load ( )
```

Charge les paramètres.

Cette fonction charge les paramètres sauvegardés dans un fichier texte. Elle récupère le nombre de lignes, le nombre de colonnes, le nombre de jetons à aligner et la difficulté du bot. Si le fichier n'existe pas, les paramètres par défaut sont retournés.

**Précondition**

res/settings.txt existant

**Renvoie**

i\_rows: Nombre de lignes de la grille  
 i\_columns: Nombre de colonnes de la grille  
 i\_nb\_jetons: Nombre de jetons à aligner pour gagner  
 i\_difficulty: Difficulté du bot

**Variables :**

- i\_rows : Nombre de lignes de la grille
- i\_columns : Nombre de colonnes de la grille
- i\_nb\_jetons : Nombre de jetons à aligner pour gagner
- i\_difficulty : Difficulté du bot

**5.7.2.7 cpp\_settings\_reset()**

```
def src.controller.ctrl_pageParametres.cpp_settings_reset ( )
```

Réinitialise les paramètres.

Cette fonction réinitialise les paramètres par défaut.

**Précondition**

tk\_root initialisé  
 view\_pp initialisé

**5.7.2.8 cpp\_settings\_save()**

```
def src.controller.ctrl_pageParametres.cpp_settings_save ( )
```

Sauvegarde les paramètres.

Cette fonction sauvegarde les paramètres dans un fichier texte. Si les paramètres ne sont pas valides, un message d'erreur est affiché. Les paramètres sont valides si le nombre de jetons à aligner est inférieur ou égal au nombre de lignes ou au nombre de colonnes.

**Précondition**

tk\_root initialisé  
 res/settings.txt existant

**Postcondition**

paramètres sauvegardés ou str\_message d'erreur affiché

**Variables :**

- i\_rows : Nombre de lignes de la grille
- i\_columns : Nombre de colonnes de la grille
- i\_nb\_jetons : Nombre de jetons à aligner pour gagner
- i\_difficulty : Difficulté du bot
- str\_message : Message d'information
- f\_settings : Fichier de sauvegarde des paramètres

**5.8 Référence de l'espace de nommage src.main**

Programme principal du jeu.

**Fonctions**

- def `main` ()  
*Fonction principale du jeu.*



### 5.8.1 Description détaillée

Programme principal du jeu.  
Ce module permet de lancer le jeu.

### 5.8.2 Documentation des fonctions

#### 5.8.2.1 main()

```
def src.main.main ( )
```

Fonction principale du jeu.

## 5.9 Référence de l'espace de nommage src.puissanceQuatre

### Espaces de nommage

- [bonus](#)  
*Ce module contient les fonctions relatives aux bonus.*
- [gestionPartie](#)  
*Un module qui gère la partie de puissance 4 en ligne de commande.*
- [grid](#)  
*Ce module contient les fonctions relatives à la grille de jeu.*
- [puissanceQuatre](#)  
*Ce module contient l'implémentation des règles du puissance 4.*

## 5.10 Référence de l'espace de nommage src.puissanceQuatre.bonus

Ce module contient les fonctions relatives aux bonus.

### Fonctions

- np.array [p4b\\_no\\_bonus](#) (np.array npa\_grid)  
*Bonus permettant de ne pas jouer de bonus.*
- np.array [p4b\\_invert\\_grid](#) (np.array npa\_grid)  
*Echange les pions des joueurs.*
- np.array [p4b\\_remove\\_full\\_line](#) (np.array npa\_grid)  
*Supprime une ligne pleine.*
- np.array [p4b\\_use\\_min\\_max](#) (np.array npa\_grid)  
*Bonus permettant au joueur d'utiliser l'algorithme min max pour son prochain coup.*
- np.array [p4b\\_flip\\_grid](#) (np.array npa\_grid)  
*Bonus permettant de retourner la grille.*
- np.array [p4b\\_random\\_placement](#) (np.array npa\_grid)  
*Bonus permettant de placer un jeton aléatoirement.*
- np.array [p4b\\_random\\_bonus](#) (np.array npa\_grid)  
*Bonus permettant de choisir un bonus aléatoirement.*

#### 5.10.1 Description détaillée

Ce module contient les fonctions relatives aux bonus.  
Ce module contient les fonctions de bonus.

#### 5.10.2 Documentation des fonctions

##### 5.10.2.1 p4b\_flip\_grid()

```
np.array src.puissanceQuatre.bonus.p4b_flip_grid (
    np.array npa_grid )
```

Bonus permettant de retourner la grille.

#### Précondition

`npa_grid` initialisé

#### Paramètres

<code>npa_grid</code>	Grille de jeu
-----------------------	---------------

#### Renvoie

`npa_grid` : Grille retournée

### 5.10.2.2 `p4b_invert_grid()`

```
np.array src.puissanceQuatre.bonus.p4b_invert_grid (
    np.array npa_grid )
```

Echange les pions des joueurs.

#### Précondition

`npa_grid` initialisé

#### Paramètres

<code>npa_grid</code>	Grille
-----------------------	--------

#### Renvoie

`npa_grid`: Grille inversée

#### Variables :

- `i_nb_rows` : Nombre de lignes de la grille
- `i_nb_cols` : Nombre de colonnes de la grille
- `i_row` : Indice de ligne
- `i_col` : Indice de colonne

### 5.10.2.3 `p4b_no_bonus()`

```
np.array src.puissanceQuatre.bonus.p4b_no_bonus (
    np.array npa_grid )
```

Bonus permettant de ne pas jouer de bonus.

#### Précondition

`npa_grid` initialisé

#### Paramètres

<code>npa_grid</code>	Grille de jeu
-----------------------	---------------

**Renvoie**

`npa_grid` : Grille retournée

**5.10.2.4 p4b\_random\_bonus()**

```
np.array src.puissanceQuatre.bonus.p4b_random_bonus (
    np.array npa_grid )
```

Bonus permettant de choisir un bonus aléatoirement.

**Précondition**

`npa_grid` initialisé

**Paramètres**

<code>npa_grid</code>	Grille de jeu
-----------------------	---------------

**Renvoie**

`npa_grid` : Grille retournée

**5.10.2.5 p4b\_random\_placement()**

```
np.array src.puissanceQuatre.bonus.p4b_random_placement (
    np.array npa_grid )
```

Bonus permettant de placer un jeton aléatoirement.

**Précondition**

`npa_grid` initialisé

**Paramètres**

<code>npa_grid</code>	Grille de jeu
-----------------------	---------------

**Renvoie**

`npa_grid` : Grille retournée

**5.10.2.6 p4b\_remove\_full\_line()**

```
np.array src.puissanceQuatre.bonus.p4b_remove_full_line (
    np.array npa_grid )
```

Supprime une ligne pleine.

**Précondition**

`npa_grid` initialisé

**Paramètres**

<code>npa_grid</code>	Grille
-----------------------	--------

**Renvoie**

`npa_grid`: Grille avec une ligne pleine en moins

**Variables :**

- `i_nb_rows` : Nombre de lignes de la grille
- `i_nb_cols` : Nombre de colonnes de la grille
- `i_row` : Indice de ligne
- `i_col` : Indice de colonne
- `b_full` : Booléen indiquant si la ligne est pleine
- `i_row2` : Indice de ligne

**5.10.2.7 p4b\_use\_min\_max()**

```
np.array src.puissanceQuatre.bonus.p4b_use_min_max (
    np.array npa_grid )
```

Bonus permettant au joueur d'utiliser l'algorithme min max pour son prochain coup.

**Précondition**

`npa_grid` initialisé

**Paramètres**

<code>npa_grid</code>	Grille de jeu
-----------------------	---------------

**Renvoie**

`npa_grid` : Grille avec un coup de plus de joué

**Variables :**

- `i_col` : La colonne qui va être jouée avec l'algorithme min max

## 5.11 Référence de l'espace de nommage

### `src.puissanceQuatre.gestionPartie`

Un module qui gère la partie de puissance 4 en ligne de commande.

**Fonctions**

- `def gp_choose_bonus ()`  
*Récupère le bonus choisi par le joueur.*
- `def gp_show_rules ()`  
*Affiche les règles du jeu.*
- `str gp_get_player_choice (int i_nb_colonnes, np.array npa_grille)`  
*Récupère le choix du joueur lors de son tour.*
- `np.array gp_handle_undo_redo (bool b_undo, np.array npa_grille)`  
*Méthode permettant au joueur d'annuler ou de refaire son dernier coup.*
- `def gp_use_bonus (str s_bonus, np.array npa_grille)`  
*Méthode permettant au joueur d'utiliser son bonus.*
- `def gp_handle_player_turn (np.array npa_grille, str s_bonus)`  
*Méthode permettant de gérer le tour du joueur.*
- `def gp_handle_bot_turn (np.array npa_grille, str s_bonus, int i_nb_jeton_victoire)`  
*Méthode permettant de gérer le tour du bot.*
- `def gp_handle_victory (np.array npa_grille, int i_ligne_joueur, int i_joueur, int i_colonne_joueur, int i_nb_jeton_victoire)`  
*Méthode permettant de gérer la victoire d'un joueur.*
- `def gp_gestion_partie (int i_nb_lignes=6, int i_nb_colonnes=7, int i_nb_jeton_victoire=4)`  
*Gère le déroulement d'une partie de puissance 4.*

### 5.11.1 Description détaillée

Un module qui gère la partie de puissance 4 en ligne de commande.

Ce module contient les fonctions permettant de gérer la partie en ligne de commande.

### 5.11.2 Documentation des fonctions

#### 5.11.2.1 `gp_choose_bonus()`

```
def src.puissanceQuatre.gestionPartie.gp_choose_bonus ( )
```

Récupère le bonus choisi par le joueur.

**Renvoie**

Le bonus choisi par le joueur

**Variables :**

- `ts_bonus` : Liste, contient les fonctions des bonus
- `i_bonus` : Entier, le bonus choisi par le joueur

#### 5.11.2.2 `gp_gestion_partie()`

```
def src.puissanceQuatre.gestionPartie.gp_gestion_partie (
    int i_nb_lignes = 6,
    int i_nb_colonnes = 7,
    int i_nb_jeton_victoire = 4 )
```

Gère le déroulement d'une partie de puissance 4.

Méthode gérant le déroulement d'une partie de puissance 4 en ligne de commande.

**Variables :**

- `b_victoire` : Booléen, True si un joueur a gagné, False sinon
- `b_bonus_utilise` : Booléen, True si le joueur a utilisé son bonus, False sinon
- `t_undo_redo` : Liste, contient les grilles pour l'undo et le redo
- `npa_grille` : `np.array`, la grille de jeu
- `i_colonne_joueur` : Entier, la colonne où le joueur veut jouer
- `i_ligne_joueur` : Entier, la ligne où le joueur veut jouer

**Paramètres**

<code>i_nb_lignes</code>	Taille de la grille en lignes
<code>i_nb_colonnes</code>	Taille de la grille en colonnes
<code>i_nb_jeton_victoire</code>	Nombre de jetons à aligner pour gagner

#### 5.11.2.3 `gp_get_player_choice()`

```
str src.puissanceQuatre.gestionPartie.gp_get_player_choice (
    int i_nb_colonnes,
    np.array npa_grille )
```

Récupère le choix du joueur lors de son tour.

**Paramètres**

<code>i_nb_colonnes</code>	Nombre de colonnes de la grille
<code>npa_grille</code>	Grille de jeu

**Précondition**

`i_nb_colonnes > 0`  
`npa_grille` initialisé

**Renvoie**

Le choix du joueur

**Postcondition**

Le choix du joueur récupéré

**Variables :**

- `i_colonne_joueur` : Entier, la colonne où le joueur veut jouer
- `s_colonne_joueur` : Chaîne de caractères, le choix du joueur

**5.11.2.4 gp\_handle\_bot\_turn()**

```
def src.puissanceQuatre.gestionPartie.gp_handle_bot_turn (
    np.array npa_grille,
    str s_bonus,
    int i_nb_jeton_victoire )
```

Méthode permettant de gérer le tour du bot.

**Paramètres**

<i>npa_grille</i>	np.array, la grille de jeu
<i>s_bonus</i>	str, le nom du bonus
<i>i_nb_jeton_victoire</i>	int, le nombre de jetons à aligner pour gagner

**Variables :**

- `npa_grille` : np.array, la grille de jeu
- `s_bonus` : str, le nom du bonus choisi par le joueur
- `i_colonne_joueur` : Entier, la colonne où le joueur veut jouer
- `i_ligne_joueur` : Entier, la ligne où le joueur veut jouer

**5.11.2.5 gp\_handle\_player\_turn()**

```
def src.puissanceQuatre.gestionPartie.gp_handle_player_turn (
    np.array npa_grille,
    str s_bonus )
```

Méthode permettant de gérer le tour du joueur.

**Paramètres**

<i>npa_grille</i>	np.array, la grille de jeu
<i>s_bonus</i>	str, le nom du bonus

**Renvoie**

: `np.array`, la grille de jeu modifiée

**Variables :**

- `npa_grille` : `np.array`, la grille de jeu
- `s_bonus` : `str`, le nom du bonus choisi par le joueur

**5.11.2.6 `gp_handle_undo_redo()`**

```
np.array src.puissanceQuatre.gestionPartie.gp_handle_undo_redo (
    bool b_undo,
    np.array npa_grille )
```

Méthode permettant au joueur d'annuler ou de refaire son dernier coup.

**Paramètres**

<code>b_undo</code>	booléen indiquant si c'est un undo ou un redo (True pour undo, False pour redo)
<code>npa_grille</code>	<code>np.array</code> , la grille de jeu

**Renvoie**

: `np.array`, la grille de jeu modifiée

**5.11.2.7 `gp_handle_victory()`**

```
def src.puissanceQuatre.gestionPartie.gp_handle_victory (
    np.array npa_grille,
    int i_ligne_joueur,
    int i_joueur,
    int i_colonne_joueur,
    int i_nb_jeton_victoire )
```

Méthode permettant de gérer la victoire d'un joueur.

**Paramètres**

<code>npa_grille</code>	<code>np.array</code> , la grille de jeu
<code>i_ligne_joueur</code>	Entier, la ligne où le joueur veut jouer
<code>i_joueur</code>	Entier, le numéro du joueur
<code>i_colonne_joueur</code>	Entier, la colonne où le joueur veut jouer
<code>i_nb_jeton_victoire</code>	Entier, le nombre de jetons à aligner pour gagner

**Renvoie**

: Booléen, True si un joueur a gagné, False sinon

**Variables :**

- `b_victoire` : Booléen, True si un joueur a gagné, False sinon

**5.11.2.8 `gp_show_rules()`**

```
def src.puissanceQuatre.gestionPartie.gp_show_rules ( )
```

Affiche les règles du jeu.

### 5.11.2.9 gp\_use\_bonus()

```
def src.puissanceQuatre.gestionPartie.gp_use_bonus (
    str s_bonus,
    np.array npa_grille )
```

Méthode permettant au joueur d'utiliser son bonus.

#### Paramètres

<i>s_bonus</i>	str, le nom du bonus
<i>npa_grille</i>	np.array, la grille de jeu

#### Renvoie

: np.array, la grille de jeu modifiée

## 5.12 Référence de l'espace de nommage src.puissanceQuatre.grid

Ce module contient les fonctions relatives à la grille de jeu.

### Fonctions

- np.array [pq\\_init\\_grille](#) (int i\_max\_ligne, int i\_max\_colonne)  
*L'initiateur de la grille.*
- np.array [pq\\_reset\\_grille](#) (np.array npa\_grille)  
*Réinitialise la grille.*
- def [pq\\_print\\_grille](#) (np.array npa\_grille)  
*Affiche la grille.*
- np.array [pq\\_apply\\_gravity](#) (np.array npa\_grille)  
*Applique la gravité sur la grille, cela permet de faire tomber les jetons dans la grille.*

### 5.12.1 Description détaillée

Ce module contient les fonctions relatives à la grille de jeu.

Ce module contient les fonctions de gestion de la grille de jeu. Notamment l'initialisation de la grille, l'affichage de la grille et la réinitialisation de la grille.

### 5.12.2 Documentation des fonctions

#### 5.12.2.1 pq\_apply\_gravity()

```
np.array src.puissanceQuatre.grid.pq_apply_gravity (
    np.array npa_grille )
```

Applique la gravité sur la grille, cela permet de faire tomber les jetons dans la grille.

#### Précondition

npa\_grille initialisé

#### Paramètres

<i>npa_grille</i>	La grille à modifier
-------------------	----------------------

#### Postcondition

npa\_grille contient des 0 et des 1 ou 2

il n'y a pas de 0 sous un 1 ou un 2

#### Variables :



- `i_nb_ligne` : Nombre de lignes de la grille
- `i_nb_colonne` : Nombre de colonnes de la grille
- `i_boucle_colonne` : Compteur de boucle pour les colonnes de la grille
- `i_boucle_ligne` : Compteur de boucle pour les lignes de la grille
- `i_compt` : Compteur de sécurité

#### 5.12.2.2 `pq_init_grille()`

```
np.array src.puissanceQuatre.grid.pq_init_grille (
    int i_max_ligne,
    int i_max_colonne )
```

L'initiateur de la grille.

##### Précondition

`i_max_ligne > 1` et `i_max_colonne > 1`

##### Paramètres

<code>i_max_ligne</code>	Le nombre de lignes de la grille
<code>i_max_colonne</code>	Le nombre de colonnes de la grille

##### Postcondition

`npa_grille` initialisé

##### Renvoie

La grille créée

##### Variables :

- `npa_grille` : `np.array`

#### 5.12.2.3 `pq_print_grille()`

```
def src.puissanceQuatre.grid.pq_print_grille (
    np.array npa_grille )
```

Affiche la grille.

##### Variables :

- `char_joueur` : Le caractère du jeton du joueur
- `char_bot` : Le caractère du jeton du bot
- `char_vide` : Le caractère représentant une case vide
- `i_max_ligne` : Le nombre de lignes de la grille
- `i_max_colonne` : Le nombre de colonnes de la grille
- `i_boucle_colonne` : Le compteur de boucle pour les colonnes de la grille
- `i_boucle_ligne` : Le compteur de boucle pour les lignes de la grille

##### Paramètres

<code>npa_grille</code>	La grille à afficher
-------------------------	----------------------

#### 5.12.2.4 `pq_reset_grille()`

```
np.array src.puissanceQuatre.grid.pq_reset_grille (
    np.array npa_grille )
```

Réinitialise la grille.

**Variables :**

- i\_boucle : Entier
- i\_max\_ligne : Entier
- i\_max\_colonne : Entier

**Précondition**

npa\_grille initialisé

**Paramètres**

<i>npa_grille</i>	La grille à réinitialiser
-------------------	---------------------------

**Postcondition**

npa\_grille contient des 0

**Renvoie**

La grille réinitialisée

## 5.13 Référence de l'espace de nommage src.puissanceQuatre.puissanceQuatre

Ce module contient l'implémentation des règles du puissance 4.

**Fonctions**

- bool [pq\\_verif\\_colonne](#) (int i\_colonne, np.array npa\_grille)  
*Vérifie si on peut poser un jeton dans cette colonne.*
- (int, int) [pq\\_ajout\\_piece](#) (np.array npa\_grille, int i\_colonne, int i\_joueur)  
*La méthode qui gère le placement de jetons.*
- (int, float) [pq\\_minmax](#) (iJoueur, npaGrilleCopy, i\_nb\_victoire, s\_bonus, b\_bonus\_used, iColonne=0, is←  
First=False, tour=0, isthebonus=False)  
*Méthode implémentant l'algorithme minmax.*
- bool [pq\\_victoire](#) (np.array npa\_grille, int i\_ligne, int i\_colonne, int i\_joueur, int i\_nb\_victoire)
- bool [pq\\_victoire\\_ligne](#) (np.array npa\_grille, int i\_ligne, int i\_colonne, int i\_joueur, int i\_nb\_victoire)  
*Vérification de la victoire sur la ligne.*
- bool [pq\\_victoire\\_colonne](#) (np.array npa\_grille, int i\_ligne, int i\_colonne, int i\_joueur, int i\_nb\_victoire)  
*Vérification de la victoire sur une colonne.*
- bool [pq\\_victoire\\_diago](#) (np.array npa\_grille, int i\_ligne, int i\_colonne, int i\_joueur, int i\_nb\_victoire)  
*Vérification de la victoire sur les diagonales.*
- np.array [pq\\_undo](#) (np.array npa\_grille, list t\_undo\_redo)  
*Méthode permettant de revenir en arrière dans le jeu.*
- np.array [pq\\_redo](#) (np.array npa\_grille, list t\_redo)  
*Méthode permettant de revenir en avant dans le jeu.*
- bool [pq\\_partie\\_finie](#) (np.array npa\_grille, bool b\_bonus\_utilise)  
*Vérification de si la partie est finie ou non.*

**5.13.1 Description détaillée**

Ce module contient l'implémentation des règles du puissance 4.

Ce module contient la gestion de la structure du puissance 4 et la gestion du jeu.

**5.13.2 Documentation des fonctions****5.13.2.1 pq\_ajout\_piece()**

```
(int, int) src.puissanceQuatre.puissanceQuatre.pq_ajout_piece (
    np.array npa_grille,
```

```
int i_colonne,
int i_joueur )
```

La méthode qui gère le placement de jetons.

Cette méthode permet de placer un jeton dans une colonne donnée. Elle vérifie si la colonne est valide et si elle n'est pas pleine. Elle renvoie les coordonnées du nouveau jeton.

#### Précondition

$0 < i\_colonne \leq npa\_grille.shape[0]$   
 npa\_grille initialisé

#### Paramètres

<i>i_colonne</i>	La colonne où le joueur pose le jeton
<i>i_joueur</i>	Le joueur qui joue (1 pour le joueur, 2 pour le bot)
<i>npa_grille</i>	La grille du puissance 4

#### Postcondition

npa\_grille contient un nouvel entier

#### Renvoie

Les coordonnées du nouveau jeton

#### Variables :

- *i\_boucle* **Entier** : Compteur de boucle
- *i\_max\_ligne* **Entier** : Nombre de lignes dans la grille
- *ti\_coords* **Tuple d'entiers** : Coordonnées du nouveau jeton

#### 5.13.2.2 pq\_minmax()

```
(int, float) src.puissanceQuatre.puissanceQuatre.pq_minmax (
    iJoueur,
    npaGrilleCopy,
    i_nb_victoire,
    s_bonus,
    b_bonus_used,
    iColonne = 0,
    isFirst = False,
    tour = 0,
    isthebonus = False )
```

Méthode implémentant l'algorithme minmax.

#### 5.13.2.3 pq\_partie\_finie()

```
bool src.puissanceQuatre.puissanceQuatre.pq_partie_finie (
    np.array npa_grille,
    bool b_bonus_utilise )
```

Vérification de si la partie est finie ou non.

La vérification se fait avec deux critères : Si la grille est pleine ou non, ainsi que si le joueur peut encore utiliser son bonus.

#### Variables :

- *i\_nb\_lignes* : Le nombre de lignes de la grille
- *i\_nb\_colonnes* : Le nombre de colonnes de la grille
- *b\_tableau\_plein* : Booléen, True si la grille est pleine, False sinon
- *i\_boucle\_ligne* : Entier, Compteur de boucle pour les lignes
- *i\_boucle\_colonne* : Entier, Compteur de boucle pour les colonnes

**Préconditions :**

- `npa_grille` initialisé
- $2 \leq i\_nb\_lignes$
- $2 \leq i\_nb\_colonnes$

**Paramètres**

<i>npa_grille</i>	La grille du puissance 4
<i>b_bonus_utilise</i>	Un booléen permettant de savoir si le joueur a utilisé son bonus ou non.

**Renvoie**

True si la partie est finie, False sinon

**5.13.2.4 `pq_redo()`**

```
np.array src.puissanceQuatre.puissanceQuatre.pq_redo (
    np.array npa_grille,
    list t_redo )
```

Méthode permettant de revenir en avant dans le jeu.

**Paramètres**

<i>npa_grille</i>	La grille du puissance 4
<i>t_redo</i>	La liste contenant les grilles pour l'undo et le redo

**Renvoie**

La grille du puissance 4 après le redo

**Variables :**

- `npa` : np.array, la grille du puissance 4 au coup annulé

**5.13.2.5 `pq_undo()`**

```
np.array src.puissanceQuatre.puissanceQuatre.pq_undo (
    np.array npa_grille,
    list t_undo_redo )
```

Méthode permettant de revenir en arrière dans le jeu.

**Paramètres**

<i>npa_grille</i>	La grille du puissance 4
<i>t_undo_redo</i>	La liste contenant les grilles pour l'undo et le redo

**Renvoie**

La grille du puissance 4 après l'undo

**Variables :**

- `npa` : np.array, la grille du puissance 4 au coup précédent

**5.13.2.6 `pq_verif_colonne()`**

```
bool src.puissanceQuatre.puissanceQuatre.pq_verif_colonne (
    int i_colonne,
    np.array npa_grille )
```

Vérifie si on peut poser un jeton dans cette colonne.

Cette fonction permet de vérifier si on peut poser un jeton dans une colonne donnée. Elle vérifie si la colonne est valide et si elle n'est pas pleine. Elle renvoie un booléen indiquant si on peut poser un jeton dans cette colonne ou non.

#### Précondition

$0 < i\_colonne \leq npa\_grille.shape[0]$   
 npa\_grille initialisé

#### Paramètres

<i>i_colonne</i>	La colonne où on souhaite poser un jeton
<i>npa_grille</i>	La grille de jeu

#### Postcondition

npa\_grille[i\_colonne] contient au moins un 0

#### Renvoie

True si on peut poser le jeton, False sinon

#### Variable :

- b\_resultat : Booléen
- i\_boucle : Entier

#### 5.13.2.7 pq\_victoire()

```
bool src.puissanceQuatre.puissanceQuatre.pq_victoire (
    np.array npa_grille,
    int i_ligne,
    int i_colonne,
    int i_joueur,
    int i_nb_victoire )

: Méthode appelant les trois vérifications de victoire.

@param npa_grille: La grille du puissance 4
@param i_ligne: La ligne où le jeton a été posé
@param i_colonne: La colonne où le jeton a été posé
@param i_joueur: Le joueur qui a joué (1 pour le joueur humain, 2 pour le
    bot)
@param i_nb_victoire: Nombre de jetons à combiner pour gagner
@return True si le joueur i_joueur a gagné, False sinon
```

#### 5.13.2.8 pq\_victoire\_colonne()

```
bool src.puissanceQuatre.puissanceQuatre.pq_victoire_colonne (
    np.array npa_grille,
    int i_ligne,
    int i_colonne,
    int i_joueur,
    int i_nb_victoire )
```

Vérification de la victoire sur une colonne.

#### Variables :

- i\_compteur : Entier, Le nombre de jetons du joueur dans la ligne
- i\_nb\_lignes : Entier, Nombre de lignes dans la grille
- b\_victoire : Booléen, Indique si le joueur a gagné ou non

#### Préconditions :

- `npa_grille` initialisé
- `npa_grille` contient un jeton en `i_ligne`, `i_colonne`
- $1 \leq i_{\text{joueur}} \leq 2$

**Paramètres**

<i>npa_grille</i>	La grille du puissance 4
<i>i_ligne</i>	La ligne où le jeton a été posé
<i>i_colonne</i>	La colonne où le jeton a été posé
<i>i_joueur</i>	Le joueur qui a joué
<i>i_nb_victoire</i>	Le nombre de jetons nécessaire pour la victoire

**Renvoie**

True si le joueur `i_joueur` a gagné, False sinon

**5.13.2.9 pq\_victoire\_diago()**

```
bool src.puissanceQuatre.puissanceQuatre.pq_victoire_diago (
    np.array npa_grille,
    int i_ligne,
    int i_colonne,
    int i_joueur,
    int i_nb_victoire )
```

Vérification de la victoire sur les diagonales.

**Variables :**

- `i_compteur` : Entier, Le nombre de jetons du joueur dans la ligne
- `i_nb_lignes` : Entier, Nombre de lignes dans la grille
- `i_nb_colonnes` : Entier, Nombre de colonnes dans la grille
- `tti_directions` : Tableau de tuples d'entiers, Les directions à vérifier
- `i_dx` : Entier, Composante x de la direction
- `i_dy` : Entier, Composante y de la direction

**Préconditions :**

- `npa_grille` initialisé
- `npa_grille` contient un jeton en `i_ligne`, `i_colonne`
- $1 \leq i_{\text{joueur}} \leq 2$

**Paramètres**

<i>npa_grille</i>	La grille du puissance 4
<i>i_ligne</i>	La ligne où le jeton a été posé
<i>i_colonne</i>	La colonne où le jeton a été posé
<i>i_joueur</i>	Le joueur qui a joué
<i>i_nb_victoire</i>	Le nombre de jetons nécessaire pour la victoire

**Renvoie**

True si le joueur `i_joueur` a gagné, False sinon

**5.13.2.10 pq\_victoire\_ligne()**

```
bool src.puissanceQuatre.puissanceQuatre.pq_victoire_ligne (
    np.array npa_grille,
    int i_ligne,
```

```
int i_colonne,
int i_joueur,
int i_nb_victoire )
```

Vérification de la victoire sur la ligne.

#### Variables :

- `i_compteur` : Entier, Le nombre de jetons du joueur dans la ligne
- `b_vu` : Booléen, ajouter explication
- `b_suite` : Booléen, ajouter explication
- `i_nb_colonnes` : Entier, Nombre de colonnes dans la grille
- `i_boucle` : Entier, Compteur de boucle
- `i_debut` : Entier, premier emplacement possible pour la combinaison de victoire dans la ligne
- `i_fin` : Entier, dernier emplacement possible pour la combinaison de victoire dans la ligne

#### Préconditions :

- `npa_grille` initialisé
- `npa_grille` contient un jeton en `i_ligne`, `i_colonne`
- `1 <= i_joueur <= 2`

```
@param npa_grille: La grille du jeu
@param i_ligne: La ligne où le jeton a été posé
@param i_colonne: La colonne où le jeton a été posé
@param i_joueur: Le joueur qui a joué
@param i_nb_victoire: Le nombre de jetons nécessaire pour la victoire

@return True si le joueur a gagné, False sinon
```

## 5.14 Référence de l'espace de nommage src.utils

### Espaces de nommage

- [bonus\\_utils](#)  
*Ce module contient les fonctions utiles aux bonus.*
- [colors\\_utils](#)  
*Ce module contient les fonctions relatives aux couleurs.*

## 5.15 Référence de l'espace de nommage src.utils.bonus\_utils

Ce module contient les fonctions utiles aux bonus.

### Fonctions

- `list[tuple[str,...]]` [bu\\_get\\_bonuses](#) ()  
*Retourne la liste des noms des fonctions bonus.*
- `str` [bu\\_get\\_bonus\\_name](#) (tuple `t_function`)  
*Retourne le nom d'une fonction bonus.*
- `str` [bu\\_format\\_bonus\\_name](#) (str `s_bonus_name`)  
*Formate le nom d'un bonus.*
- `str` [bu\\_unformat\\_bonus\\_name](#) (str `s_bonus_name`)  
*Déformate le nom d'un bonus.*
- `str` [bu\\_get\\_bonus\\_description](#) (str `s_bonus_name`)  
*Retourne la description d'un bonus.*

#### 5.15.1 Description détaillée

Ce module contient les fonctions utiles aux bonus.

Ce module contient les fonctions utilitaires relatives aux bonus.

#### 5.15.2 Documentation des fonctions

### 5.15.2.1 bu\_format\_bonus\_name()

```
str src.utils.bonus_utils.bu_format_bonus_name (
    str s_bonus_name )
```

Formate le nom d'un bonus.

Cette fonction formate le nom d'un bonus pour l'afficher dans le menu. Elle enlève le préfixe "p4b\_" et remplace les "\_" par des espaces.

### 5.15.2.2 bu\_get\_bonus\_description()

```
str src.utils.bonus_utils.bu_get_bonus_description (
    str s_bonus_name )
```

Retourne la description d'un bonus.

Cette fonction retourne la description d'un bonus. Elle utilise pour cela la documentation de la fonction bonus.

### 5.15.2.3 bu\_get\_bonus\_name()

```
str src.utils.bonus_utils.bu_get_bonus_name (
    tuple t_function )
```

Retourne le nom d'une fonction bonus.

Cette fonction retourne le nom d'une fonction bonus.

### 5.15.2.4 bu\_get\_bonuses()

```
list[tuple[str, ...]] src.utils.bonus_utils.bu_get_bonuses ( )
```

Retourne la liste des noms des fonctions bonus.

Cette fonction retourne la liste des noms des fonctions bonus. Elle utilise le module bonus pour récupérer les fonctions bonus. Elle utilise aussi l'inspecteur pour récupérer le nom des fonctions bonus.

#### Postcondition

La liste des noms des fonctions bonus est retournée

#### Variables :

— ts\_functions : Liste des fonctions du module bonus

### 5.15.2.5 bu\_unformat\_bonus\_name()

```
str src.utils.bonus_utils.bu_unformat_bonus_name (
    str s_bonus_name )
```

Déformate le nom d'un bonus.

Cette fonction déformate le nom d'un bonus pour l'utiliser dans le code. Elle ajoute le préfixe "p4b\_" et remplace les espaces par des "\_" afin de retrouver le nom de la fonction.

## 5.16 Référence de l'espace de nommage src.utils.colors\_utils

Ce module contient les fonctions relatives aux couleurs.

### Fonctions

- (int, int, int) [cu\\_hex\\_to\\_rgb](#) (str s\_color)  
*Convertit une couleur hexadécimale en RGB.*
- int [cu\\_rgb\\_distance](#) ((int, int, int) rgb1, (int, int, int) rgb2)  
*Calcule la distance entre deux couleurs RGB.*
- bool [cu\\_colors\\_too\\_close](#) (str color1, str color2)  
*Vérifie si deux couleurs sont trop proches.*



### 5.16.1 Description détaillée

Ce module contient les fonctions relatives aux couleurs.

Ce module contient les fonctions de gestion des couleurs. Notamment la conversion d'une couleur hexadécimale en RGB et la vérification de la distance entre deux couleurs.

### 5.16.2 Documentation des fonctions

#### 5.16.2.1 `cu_colors_too_close()`

```
bool src.utils.colors_utils.cu_colors_too_close (
    str color1,
    str color2 )
```

Vérifie si deux couleurs sont trop proches.

Cette fonction vérifie si deux couleurs sont trop proches. Elle utilise la fonction `cu_rgb_distance` pour calculer la distance entre les deux couleurs. Si la distance est inférieure à 50, les couleurs sont trop proches.

##### Paramètres

<i>color1</i>	Couleur 1 au format hexadécimal
<i>color2</i>	Couleur 2 au format hexadécimal

##### Renvoie

True si les couleurs sont trop proches, False sinon

#### 5.16.2.2 `cu_hex_to_rgb()`

```
(int, int, int) src.utils.colors_utils.cu_hex_to_rgb (
    str s_color )
```

Convertit une couleur hexadécimale en RGB.

Cette fonction convertit une couleur hexadécimale en RGB.

##### Paramètres

<i>s_color</i>	Couleur hexadécimale
----------------	----------------------

##### Renvoie

Couleur RGB

#### 5.16.2.3 `cu_rgb_distance()`

```
int src.utils.colors_utils.cu_rgb_distance (
    (int, int, int) rgb1,
    (int, int, int) rgb2 )
```

Calcule la distance entre deux couleurs RGB.

Cette fonction calcule la distance entre deux couleurs RGB. Elle utilise la formule de la distance euclidienne.

##### Paramètres

<i>rgb1</i>	Couleur 1
<i>rgb2</i>	Couleur 2

## Renvoi

Distance entre les deux couleurs

## 5.17 Référence de l'espace de nommage src.view

### Espaces de nommage

- [view\\_main](#)  
*Ce module contient les fonctions de base utile à la vue du jeu.*
- [view\\_pageAccueil](#)  
*Ce module représente la vue de la page d'accueil.*
- [view\\_pageBonus](#)  
*Ce module représente la vue de la page de choix du bonus.*
- [view\\_pageJeu](#)  
*Ce module représente la vue de la page de jeu.*
- [view\\_pageParametres](#)  
*Vue de la page des paramètres.*

## 5.18 Référence de l'espace de nommage src.view.view\_main

Ce module contient les fonctions de base utile à la vue du jeu.

### Fonctions

- `tk.Tk vm_init ()`  
*Initialise la fenêtre de jeu.*
- `def vm_quit (tk.Tk tk_win_root)`  
*Ferme la fenêtre de jeu.*
- `tk.Menu vm_menu (tk.Frame tk_old_frame, bool b_in_game)`  
*Initialise le menu de la fenêtre de jeu.*
- `def vm_message_game_ended (str s_message, tk.Frame tkf_page_jeu)`  
*Affiche un message de fin de partie.*
- `def vm_message_warning (str str_message)`  
*Affiche un message d'avertissement.*
- `def vm_message_info (str str_message)`  
*Affiche un message d'information.*
- `def vm_remove_frame (tk.Frame frame)`  
*Supprime un cadre.*
- `def vm_update (tk.Tk tk_root)`  
*Met à jour la fenêtre principale.*

### 5.18.1 Description détaillée

Ce module contient les fonctions de base utile à la vue du jeu.

Ce module permet de facilement avoir le menu sur toutes les fenêtres. De quitter le jeu, de recommencer une partie, etc. Il permet aussi d'afficher des messages d'information ou d'avertissement.

### 5.18.2 Documentation des fonctions

#### 5.18.2.1 vm\_init()

```
tk.Tk src.view.view_main.vm_init ( )
```

Initialise la fenêtre de jeu.

Cette fonction initialise la fenêtre principale du jeu. Elle crée la fenêtre, lui donne un titre, un logo, la rend non redimensionnable et renvoie la fenêtre créée.

#### Variables :

- `tk_root` : Fenêtre principale

**Renvoie**

Fenêtre principale

**5.18.2.2 `vm_menu()`**

```
tk.Menu src.view.view_main.vm_menu (
    tk.Frame tk_old_frame,
    bool b_in_game )
```

Initialise le menu de la fenêtre de jeu.

Cette fonction initialise le menu de la fenêtre principale du jeu. Elle crée le menu, les sous-menus, les commandes et renvoie le menu créé. Ce menu est affiché dans toutes les fenêtres.

**Variables :**

- `tkm_menu_bar` : Menu de la fenêtre de jeu
- `tkm_menu_partie` : sous-menu permettant de gérer la partie
- `tkm_menu_param` : sous-menu permettant d'accéder aux paramètres
- `tkm_menu_a_propos` : sous-menu permettant d'accéder à la page "À propos"

**Renvoie**

Menu de la fenêtre de jeu

**5.18.2.3 `vm_message_game_ended()`**

```
def src.view.view_main.vm_message_game_ended (
    str s_message,
    tk.Frame tkf_page_jeu )
```

Affiche un message de fin de partie.

Cette fonction affiche un message de fin de partie. Elle demande à l'utilisateur s'il veut rejouer ou non. Si oui, elle relance une partie sinon elle revient à la page d'accueil. Elle est appelée lorsque la partie est terminée.

**Préconditions :**

- `tk_root` initialisé

**Paramètres**

<code>s_message</code>	Message à afficher
<code>tkf_page_jeu</code>	La page de jeu

**5.18.2.4 `vm_message_info()`**

```
def src.view.view_main.vm_message_info (
    str str_message )
```

Affiche un message d'information.

Cette fonction affiche un message d'information. Elle est appelée lorsque l'utilisateur fait une action qui est autorisée. Par exemple, lorsque l'utilisateur change les paramètres du jeu.

**Paramètres**

<code>str_message</code>	Message à afficher
--------------------------	--------------------

**5.18.2.5 `vm_message_warning()`**

```
def src.view.view_main.vm_message_warning (
```

```
str str_message )
```

Affiche un message d'avertissement.

Cette fonction affiche un message d'avertissement. Elle est appelée lorsque l'utilisateur fait une action qui n'est pas autorisée. Par exemple, si l'utilisateur choisit des paramètres qui ne sont pas compatibles avec le jeu.

#### Paramètres

<i>str_message</i>	Message à afficher
--------------------	--------------------

### 5.18.2.6 vm\_quit()

```
def src.view.view_main.vm_quit (
    tk.Tk tk_win_root )
```

Ferme la fenêtre de jeu.

Cette fonction ferme la fenêtre principale du jeu.

```
**Préconditions :**
* tk_root initialisé
```

```
@param tk_win_root: Fenêtre principale
```

### 5.18.2.7 vm\_remove\_frame()

```
def src.view.view_main.vm_remove_frame (
    tk.Frame frame )
```

Supprime un cadre.

Cette fonction supprime un cadre. Elle est appelée lorsque l'on veut changer de page.

#### Précondition

frame existe

#### Paramètres

<i>frame</i>	Le cadre à supprimer
--------------	----------------------

#### Postcondition

frame n'existe plus

### 5.18.2.8 vm\_update()

```
def src.view.view_main.vm_update (
    tk.Tk tk_root )
```

Met à jour la fenêtre principale.

Cette fonction met à jour la fenêtre principale. Elle est appelée lorsque l'on veut mettre à jour la fenêtre principale.

#### Paramètres

<i>tk_root</i>	Fenêtre principale
----------------	--------------------

## 5.19 Référence de l'espace de nommage src.view.view\_pageAccueil

Ce module représente la vue de la page d'accueil.

### Fonctions

- def `vpa_init` (tk.Tk tk\_root)  
*Initialise la page d'accueil.*
- def `vpa_destroy` ()  
*Détruit la page d'accueil.*

#### 5.19.1 Description détaillée

Ce module représente la vue de la page d'accueil.

Ce module contient les fonctions d'initialisation et de destruction de la page d'accueil.

#### 5.19.2 Documentation des fonctions

##### 5.19.2.1 `vpa_destroy()`

```
def src.view.view_pageAccueil.vpa_destroy ( )
```

Détruit la page d'accueil.

Cette fonction détruit la page d'accueil. Elle efface le cadre et le supprime.

##### Variables :

- `tkf_page_accueil` : Frame de la page d'accueil

##### 5.19.2.2 `vpa_init()`

```
def src.view.view_pageAccueil.vpa_init (
    tk.Tk tk_root )
```

Initialise la page d'accueil.

Cette fonction initialise la page d'accueil. Elle crée un cadre, un label contenant le titre et un bouton pour lancer une partie.

##### Variables :

- `tkf_page_accueil` : Frame de la page d'accueil
- `tkl_title` : Label contenant le titre
- `tkB_play` : Bouton pour lancer une partie

##### Préconditions :

- `tk_root` initialisé

##### Paramètres

<code>tk_root</code>	Fenêtre principale
----------------------	--------------------

##### Voir également

`src/controller/ctrl_pageAccueil.py`

## 5.20 Référence de l'espace de nommage src.view.view\_pageBonus

Ce module représente la vue de la page de choix du bonus.

### Fonctions

- None `vpb_init` (tk.Tk tk\_root)

- *Initialisation de la fenêtre de choix du bonus.*  
`tuple[str,...] vpb_get_bonus ()`
- *Récupère le nom du bonus sélectionné par le joueur.*  
`def vpb_show_bonus_description (str s_description)`
- *Affiche la description d'un bonus.*  
`tk.Frame vpb_get_frame ()`
- *Accesseur du cadre de la fenêtre de choix du bonus.*

## 5.20.1 Description détaillée

Ce module représente la vue de la page de choix du bonus.

Ce module contient les fonctions permettant de gérer la vue de la page de choix du bonus.

## 5.20.2 Documentation des fonctions

### 5.20.2.1 vpb\_get\_bonus()

```
tuple[str, ...] src.view.view_pageBonus.vpb_get_bonus ( )
```

Récupère le nom du bonus sélectionné par le joueur.

Cette fonction récupère le nom du bonus sélectionné par le joueur. Elle renvoie le nom du bonus sélectionné.

**Précondition**

TKS\_BONUS initialisé

**Renvoie**

: Le nom du bonus sélectionné par le joueur

**Variables :**

- TKS\_BONUS : Variable de choix du bonus.

### 5.20.2.2 vpb\_get\_frame()

```
tk.Frame src.view.view_pageBonus.vpb_get_frame ( )
```

Accesseur du cadre de la fenêtre de choix du bonus.

Cette fonction renvoie le cadre de la fenêtre de choix du bonus. Elle est utilisée par le contrôleur principal pour afficher la fenêtre de choix du bonus.

**Précondition**

Cadre initialisé

**Renvoie**

: Cadre de la fenêtre de choix du bonus.

**Variables :**

- TKF\_PAGE\_CHOIX : Cadre de la fenêtre de choix du bonus.

### 5.20.2.3 vpb\_init()

```
None src.view.view_pageBonus.vpb_init (
    tk.Tk tk_root )
```

Initialisation de la fenêtre de choix du bonus.

Cette fonction initialise la fenêtre de choix du bonus. Elle crée un cadre, un titre, un menu déroulant pour le choix du bonus, un label pour la description du bonus et un bouton pour valider le bonus. Elle affiche aussi la description du premier bonus. Elle affiche aussi le menu sur la fenêtre.

**Précondition**

tk\_root initialisé

## Paramètres

<code>tk_root</code>	la fenêtre de base
----------------------	--------------------

## Postcondition

Fenêtre de choix du bonus initialisée

## Variables :

- TKS\_BONUS : Variable de choix du bonus.
- TKL\_DESCRIPTION\_BONUS : Label de la description du bonus.
- TKF\_PAGE\_CHOIX : Cadre de la fenêtre de choix du bonus.
- tkL\_titre : Label du titre de la fenêtre de choix du bonus.
- tkC\_bonus : Menu déroulant pour le choix du bonus.
- tkL\_description : Label de la description du bonus.
- tkB\_valider : Bouton de validation du bonus.

## 5.20.2.4 vpb\_show\_bonus\_description()

```
def src.view.view_pageBonus.vpb_show_bonus_description (
    str s_description )
```

Affiche la description d'un bonus.

Cette fonction affiche la description d'un bonus. Elle prend en paramètre la description du bonus à afficher. Elle affiche la description du bonus dans le label de la description du bonus.

## Précondition

TKL\_DESCRIPTION\_BONUS initialisé

## Paramètres

<code>s_description</code>	Description du bonus
----------------------------	----------------------

## Variables :

- TKL\_DESCRIPTION\_BONUS : Label de la description du bonus.

## 5.21 Référence de l'espace de nommage src.view.view\_pageJeu

Ce module représente la vue de la page de jeu.

## Fonctions

- def `vpj_init_page_jeu` (tk.Tk tk\_root, str st\_color\_grid)  
*Initialise la page de jeu.*
- def `vpj_destroy` ()  
*Détruit la page de jeu.*
- def `vpj_draw_grid` (int rows, int columns)  
*Dessine la grille de jeu.*
- def `vpj_show_coin` (int row, int column, str color)  
*Dessine un jeton dans une cellule.*
- (int, int) `vpj_get_grid_cell` (int i\_x, int i\_y)  
*Récupère les coordonnées dans la grille de la cellule cliquée.*
- def `vpj_disable_bonus` ()  
*Désactive le bouton bonus.*
- tk.Frame `vpj_get_frame` ()  
*Accesseur de la frame de la page de jeu.*
- def `vpj_set_info` (str st\_info)  
*Modifie le texte du label d'information.*

### 5.21.1 Description détaillée

Ce module représente la vue de la page de jeu.

Ce module contient les fonctions permettant de gérer la vue de la page de jeu.

### 5.21.2 Documentation des fonctions

#### 5.21.2.1 `vpj_destroy()`

```
def src.view.view_pageJeu.vpj_destroy ( )
```

Détruit la page de jeu.

Cette fonction détruit la page de jeu. Elle efface le cadre et le supprime.

**Variables :**

— `tkf_page_jeu` : Frame de la page de jeu

**Précondition**

`tkf_page_jeu` initialisé

#### 5.21.2.2 `vpj_disable_bonus()`

```
def src.view.view_pageJeu.vpj_disable_bonus ( )
```

Désactive le bouton bonus.

Cette fonction désactive le bouton bonus et change son relief. Elle est utilisée par le contrôleur de la page de jeu pour désactiver le bouton.

**Précondition**

`TKS_BONUS` initialisé

#### 5.21.2.3 `vpj_draw_grid()`

```
def src.view.view_pageJeu.vpj_draw_grid (
    int rows,
    int columns )
```

Dessine la grille de jeu.

Cette fonction dessine la grille de jeu. Elle prend en paramètre le nombre de lignes et de colonnes de la grille. Elle dessine la grille dans le canvas.

**Variables :**

- `tkc_grid` : Canvas de la page de jeu
- `i_canvas_width` : Largeur du canvas
- `i_canvas_height` : Hauteur du canvas
- `cell_width` : Largeur d'une cellule
- `cell_height` : Hauteur d'une cellule
- `ti_upper_left` : Coordonnées du coin supérieur gauche d'une cellule
- `ti_lower_right` : Coordonnées du coin inférieur droit d'une cellule

**Paramètres**

<i>rows</i>	Nombre de lignes de la grille
<i>columns</i>	Nombre de colonnes de la grille



#### 5.21.2.4 `vpj_get_frame()`

```
tk.Frame src.view.view_pageJeu.vpj_get_frame ( )
```

Accesseur de la frame de la page de jeu.

Cette fonction renvoie la frame de la page de jeu. Elle est utilisée par le contrôleur principal pour afficher la page de jeu.

##### Précondition

TKF\_PAGE\_JEU initialisé

##### Renvoie

: Frame de la page de jeu

##### Variables :

— TKF\_PAGE\_JEU : Frame de la page de jeu

#### 5.21.2.5 `vpj_get_grid_cell()`

```
(int, int) src.view.view_pageJeu.vpj_get_grid_cell (
    int i_x,
    int i_y )
```

Récupère les coordonnées dans la grille de la cellule cliquée.

Récupère les coordonnées dans la grille de la cellule cliquée en fonction des coordonnées du clic dans le canvas.

##### Variables :

— `i_canvas_width` : Largeur du canvas  
— `i_canvas_height` : Hauteur du canvas

##### Paramètres

<code>i_x</code> ↔ _↔ <code>x</code>	Coordonnée x du clic dans le canvas
<code>i_y</code> ↔ _↔ <code>y</code>	Coordonnée y du clic dans le canvas

##### Renvoie

: Coordonnées de la cellule cliquée dans la grille

#### 5.21.2.6 `vpj_init_page_jeu()`

```
def src.view.view_pageJeu.vpj_init_page_jeu (
    tk.Tk tk_root,
    str st_color_grid )
```

Initialise la page de jeu.

Cette fonction initialise la page de jeu. Elle crée un cadre, un canvas pour afficher la grille, un bouton pour annuler le dernier coup, un bouton pour refaire le dernier coup, un bouton pour utiliser un bonus et un bouton pour quitter la partie. Elle affiche aussi le menu sur la fenêtre.

##### Variables :

— `tkf_page_jeu` : Frame de la page de jeu  
— `tkc_grid` : Canvas de la page de jeu  
— `i_canvas_width` : Largeur du canvas  
— `i_canvas_height` : Hauteur du canvas  
— `tkB_undo` : Bouton pour annuler le dernier coup  
— `tkB_redo` : Bouton pour refaire le dernier coup  
— `tkB_bonus` : Bouton pour utiliser un bonus

- tkB\_quit : Bouton pour quitter la partie

#### Préconditions :

- tk\_root initialisé

#### Paramètres

<i>tk_root</i>	Fenêtre principale
<i>st_color_grid</i>	Couleur de la grille au format hexadécimal

#### 5.21.2.7 vpj\_set\_info()

```
def src.view.view_pageJeu.vpj_set_info (
    str st_info )
```

Modifie le texte du label d'information.

Cette fonction modifie le texte du label d'information. Elle prend en paramètre le texte à afficher dans le label d'information. Elle affiche le texte dans le label d'information.

#### Paramètres

<i>st_info</i>	Texte à afficher dans le label d'information
----------------	--

#### 5.21.2.8 vpj\_show\_coin()

```
def src.view.view_pageJeu.vpj_show_coin (
    int row,
    int column,
    str color )
```

Dessine un jeton dans une cellule.

Cette fonction dessine un jeton dans une cellule. Elle prend en paramètre la ligne et la colonne de la cellule où l'on va dessiner le jeton et la couleur du jeton. Elle dessine le jeton dans la cellule.

#### Paramètres

<i>row</i>	Ligne de la cellule où l'on va dessiner le jeton
<i>column</i>	Colonne de la cellule où l'on va dessiner le jeton
<i>color</i>	Couleur du jeton

## 5.22 Référence de l'espace de nommage src.view.view\_pageParametres

Vue de la page des paramètres.

### Fonctions

- def [vpp\\_init](#) (tk.Tk tk\_root)  
*Fonction initialisant la page des paramètres.*
- def [vpp\\_init\\_custom](#) ()  
*Fonction initialisant la partie personnalisation des paramètres.*
- def [vpp\\_init\\_settings](#) ()  
*Fonction initialisant la partie paramètres du jeu.*
- def [vpp\\_get\\_nb\\_rows](#) ()  
*Accesseur retournant le nombre de lignes sélectionné*
- def [vpp\\_get\\_nb\\_columns](#) ()  
*Accesseur retournant le nombre de colonnes sélectionné*

- `def vpp_get_nb_jetons ()`  
*Accesseur retournant le nombre de jetons requis sélectionné*
- `def vpp_get_difficulty ()`  
*Accesseur retournant la difficulté sélectionnée.*
- `def vpp_set_nb_rows (int i_rows)`  
*Mutateur pour le nombre de lignes.*
- `def vpp_set_nb_columns (int i_columns)`  
*Mutateur pour le nombre de colonnes.*
- `def vpp_set_nb_jetons (int i_nb_jetons)`  
*Mutateur pour le nombre de jetons requis.*
- `def vpp_set_difficulty (i_difficulty)`  
*Mutateur pour la difficulté*
- `def vpp_reset_settings ()`  
*Réinitialise les paramètres du jeu.*
- `def vpp_get_joueur_color ()`  
*Accesseur retournant la couleur des jetons du joueur.*
- `def vpp_get_bot_color ()`  
*Accesseur retournant la couleur des jetons du bot.*
- `def vpp_get_grid_color ()`  
*Accesseur retournant la couleur de la grille.*
- `def vpp_set_joueur_color (str s_color)`  
*Mutateur pour la couleur des jetons du joueur.*
- `def vpp_set_bot_color (str s_color)`  
*Mutateur pour la couleur des jetons du bot.*
- `def vpp_set_grid_color (str s_color)`  
*Mutateur pour la couleur de la grille.*
- `def vpp_reset_customs ()`  
*Réinitialise les paramètres de personnalisation.*
- `def vpp_askcolor (str s_element)`  
*Ouvre un sélecteur de couleur.*

### 5.22.1 Description détaillée

Vue de la page des paramètres.

La vue de la page des paramètres permet de gérer l'affichage de la page des paramètres et traiter les paramètres.

### 5.22.2 Documentation des fonctions

#### 5.22.2.1 `vpp_askcolor()`

```
def src.view.view_pageParametres.vpp_askcolor (
    str s_element )
```

Ouvre un sélecteur de couleur.

Cette fonction ouvre un sélecteur de couleur. Elle est appelée lorsque l'utilisateur clique sur le bouton pour choisir la couleur.

**Précondition**

`s_element` est soit "joueur", "bot" ou "grille"

**Paramètres**

<code>s_element</code>	L'élément dont on veut changer la couleur
------------------------	---

**Postcondition**

La couleur de l'élément est modifiée

#### 5.22.2.2 `vpp_get_bot_color()`

```
def src.view.view_pageParametres.vpp_get_bot_color ( )
```

Accesseur retournant la couleur des jetons du bot.

Cette fonction retourne la couleur des jetons du bot. Elle est appelée lorsque l'utilisateur clique sur le bouton "↔ Enregistrer".

**Précondition**

TIS\_CUSTOM\_COLOR\_BOT initialisé

**Renvoie**

La couleur des jetons du bot

**Variables :**

- TIS\_CUSTOM\_COLOR\_BOT : Tableau d'entier pour la couleur des jetons du bot

### 5.22.2.3 vpp\_get\_difficulty()

```
def src.view.view_pageParametres.vpp_get_difficulty ( )
```

Accesseur retournant la difficulté sélectionnée.

Cette fonction retourne la difficulté sélectionnée par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer".

**Précondition**

TKS\_SCALE initialisé

**Renvoie**

La difficulté sélectionnée

**Variables :**

- TKS\_SCALE : Slider pour la difficulté

### 5.22.2.4 vpp\_get\_grid\_color()

```
def src.view.view_pageParametres.vpp_get_grid_color ( )
```

Accesseur retournant la couleur de la grille.

Cette fonction retourne la couleur de la grille. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer".

**Précondition**

TIS\_CUSTOM\_COLOR\_GRID initialisé

**Renvoie**

La couleur de la grille

**Variables :**

- TIS\_CUSTOM\_COLOR\_GRID : Tableau d'entier pour la couleur de la grille

### 5.22.2.5 vpp\_get\_joueur\_color()

```
def src.view.view_pageParametres.vpp_get_joueur_color ( )
```

Accesseur retournant la couleur des jetons du joueur.

Cette fonction retourne la couleur des jetons du joueur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer".

**Précondition**

TIS\_CUSTOM\_COLOR\_JOUEUR initialisé

**Renvoie**

La couleur des jetons du joueur

**Variables :**

- TIS\_CUSTOM\_COLOR\_JOUEUR : Tableau d'entier pour la couleur des jetons du joueur

#### 5.22.2.6 `vpp_get_nb_columns()`

```
def src.view.view_pageParametres.vpp_get_nb_columns ( )
```

Accesseur retournant le nombre de colonnes sélectionné

Cette fonction retourne le nombre de colonnes sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer"

**Précondition**

STV\_COLUMNS initialisé

**Renvoie**

Le nombre de colonnes sélectionné

**Variables :**

— STV\_COLUMNS : StringVar pour récupérer la valeur du nombre de colonnes

#### 5.22.2.7 `vpp_get_nb_jetons()`

```
def src.view.view_pageParametres.vpp_get_nb_jetons ( )
```

Accesseur retournant le nombre de jetons requis sélectionné

Cette fonction retourne le nombre de jetons requis sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer".

**Précondition**

STV\_NB\_JETONS initialisé

**Renvoie**

Le nombre de jetons requis sélectionné

**Variables :**

— STV\_NB\_JETONS : StringVar pour récupérer la valeur du nombre de jetons requis

#### 5.22.2.8 `vpp_get_nb_rows()`

```
def src.view.view_pageParametres.vpp_get_nb_rows ( )
```

Accesseur retournant le nombre de lignes sélectionné

Cette fonction retourne le nombre de lignes sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Enregistrer"

**Précondition**

STV\_ROWS initialisé

**Renvoie**

Le nombre de lignes sélectionné

**Variables :**

— STV\_ROWS : StringVar pour récupérer la valeur du nombre de lignes

#### 5.22.2.9 `vpp_init()`

```
def src.view.view_pageParametres.vpp_init (
    tk.Tk tk_root )
```

Fonction initialisant la page des paramètres.

Cette fonction initialise la page des paramètres. Elle crée un cadre et affiche le menu sur la fenêtre. Elle initialise aussi les paramètres de jeu et de personnalisation.

**Variables :**

— tkf\_page\_parameter : Frame de la page des paramètres

**Préconditions :**

— tk\_root initialisé

## Paramètres

<code>tk_root</code>	La fenêtre principale
----------------------	-----------------------

5.22.2.10 `vpp_init_custom()`

```
def src.view.view_pageParametres.vpp_init_custom ( )
```

Fonction initialisant la partie personnalisation des paramètres.

Cette fonction initialise la partie personnalisation des paramètres. Elle crée des labels pour indiquer les choix de couleurs et des boutons pour ouvrir un sélecteur de couleur. Elle initialise aussi les couleurs par défaut.

## Précondition

TKF\_PAGE\_PARAMETER initialisé

## Postcondition

La partie personnalisation des paramètres est initialisée

TKB\_PICKER\_JOUEUR, TKB\_PICKER\_BOT et TKB\_PICKER\_GRID sont initialisés

## Variables :

- `tkl_perso` : Label pour indiquer la seconde partie des paramètres
- `tkl_color_joueur` : Label pour indiquer le choix de la couleur des jetons du joueur
- `TKB_PICKER_JOUEUR` : Bouton pour ouvrir un sélecteur de couleur pour les jetons du joueur
- `tkl_color_bot` : Label pour indiquer le choix de la couleur des jetons du bot
- `TKB_PICKER_BOT` : Bouton pour ouvrir un sélecteur de couleur pour les jetons du bot
- `tkl_color_grid` : Label pour indiquer le choix de la couleur de la grille
- `TKB_PICKER_GRID` : Bouton pour ouvrir un sélecteur de couleur pour la grille
- `tkb_save` : Bouton pour enregistrer les paramètres
- `tkb_reset` : Bouton pour réinitialiser les paramètres

5.22.2.11 `vpp_init_settings()`

```
def src.view.view_pageParametres.vpp_init_settings ( )
```

Fonction initialisant la partie paramètres du jeu.

Cette fonction initialise la partie paramètres du jeu. Elle crée des labels pour indiquer les choix de paramètres et des spinbox pour choisir les paramètres. Elle initialise aussi les paramètres par défaut.

## Précondition

TKF\_PAGE\_PARAMETER initialisé

## Postcondition

La partie paramètres du jeu est initialisée

## Variables :

- `tkl_param` : Label pour indiquer la première partie des paramètres
- `tkl_size` : Label pour indiquer le choix de la taille de la grille
- `STV_ROWS` : StringVar pour récupérer la valeur du nombre de lignes
- `tkb_nb_rows` : Spinbox pour le nombre de lignes
- `tkl_lines` : Label pour indiquer le nombre de lignes
- `STV_COLUMNS` : StringVar pour récupérer la valeur du nombre de colonnes
- `tkb_nb_columns` : Spinbox pour le nombre de colonnes
- `tkl_colonnes` : Label pour indiquer le nombre de colonnes
- `tkl_nb_jetons` : Label pour indiquer le choix du nombre de jetons requis
- `STV_NB_JETONS` : StringVar pour récupérer la valeur du nombre de jetons requis
- `tkb_nb_jetons` : Spinbox pour le nombre de jetons requis
- `tkl_difficulty` : Label pour indiquer le choix de la difficulté
- `TKS_SCALE` : Slider pour la difficulté
- `tkb_save` : Bouton pour enregistrer les paramètres
- `tkb_reset` : Bouton pour réinitialiser les paramètres

**5.22.2.12 `vpp_reset_customs()`**

```
def src.view.view_pageParametres.vpp_reset_customs ( )
```

Réinitialise les paramètres de personnalisation.

Cette fonction réinitialise les paramètres de personnalisation. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

**5.22.2.13 `vpp_reset_settings()`**

```
def src.view.view_pageParametres.vpp_reset_settings ( )
```

Réinitialise les paramètres du jeu.

Cette fonction réinitialise les paramètres du jeu. Elle est appelée lorsque l'utilisateur clique sur le bouton "↔ Réinitialiser".

**5.22.2.14 `vpp_set_bot_color()`**

```
def src.view.view_pageParametres.vpp_set_bot_color (
    str s_color )
```

Mutateur pour la couleur des jetons du bot.

Cette fonction modifie la couleur des jetons du bot. Elle est appelée lorsque l'utilisateur clique sur le bouton "↔ Réinitialiser".

**Précondition**

TIS\_CUSTOM\_COLOR\_BOT initialisé

**Paramètres**

<code>s_color</code>	La nouvelle couleur des jetons du bot
----------------------	---------------------------------------

**Postcondition**

TIS\_CUSTOM\_COLOR\_BOT est modifié

**Variables :**

- TIS\_CUSTOM\_COLOR\_BOT : Tableau d'entier pour la couleur des jetons du bot
- s\_color : La couleur des jetons du bot à modifier
- TKB\_PICKER\_BOT : Bouton pour ouvrir le sélecteur de couleur

**5.22.2.15 `vpp_set_difficulty()`**

```
def src.view.view_pageParametres.vpp_set_difficulty (
    i_difficulty )
```

Mutateur pour la difficulté

Cette fonction modifie la difficulté sélectionnée par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

**Précondition**

TKS\_SCALE initialisé

**Paramètres**

<code>i_difficulty</code>	La nouvelle difficulté
---------------------------	------------------------

**Postcondition**

TKS\_SCALE est modifié

**Variables :**

- TKS\_SCALE : Slider pour la difficulté
- i\_difficulty : La difficulté à modifier

### 5.22.2.16 vpp\_set\_grid\_color()

```
def src.view.view_pageParametres.vpp_set_grid_color (
    str s_color )
```

Mutateur pour la couleur de la grille.

Cette fonction modifie la couleur de la grille. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

#### Précondition

TIS\_CUSTOM\_COLOR\_GRID initialisé

#### Paramètres

<code>s_color</code>	La nouvelle couleur de la grille
----------------------	----------------------------------

#### Postcondition

TIS\_CUSTOM\_COLOR\_GRID est modifié

#### Variables :

- TIS\_CUSTOM\_COLOR\_GRID : Tableau d'entier pour la couleur de la grille
- s\_color : La couleur de la grille à modifier
- TKB\_PICKER\_GRID : Bouton pour ouvrir le sélecteur de couleur

### 5.22.2.17 vpp\_set\_joueur\_color()

```
def src.view.view_pageParametres.vpp_set_joueur_color (
    str s_color )
```

Mutateur pour la couleur des jetons du joueur.

Cette fonction modifie la couleur des jetons du joueur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

#### Précondition

TIS\_CUSTOM\_COLOR\_JOUEUR initialisé

#### Paramètres

<code>s_color</code>	La nouvelle couleur des jetons du joueur
----------------------	--

#### Postcondition

TIS\_CUSTOM\_COLOR\_JOUEUR est modifié

#### Variables :

- TIS\_CUSTOM\_COLOR\_JOUEUR : Tableau d'entier pour la couleur des jetons du joueur
- s\_color : La couleur des jetons du joueur à modifier
- TKB\_PICKER\_JOUEUR : Bouton pour ouvrir le sélecteur de couleur

### 5.22.2.18 vpp\_set\_nb\_columns()

```
def src.view.view_pageParametres.vpp_set_nb_columns (
    int i_columns )
```

Mutateur pour le nombre de colonnes.

Cette fonction modifie le nombre de colonnes sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".



**Précondition**

STV\_COLUMNS initialisé\*

**Paramètres**

<code>i_columns</code>	Le nouveau nombre de colonnes
------------------------	-------------------------------

**Postcondition**

STV\_COLUMNS est modifié

**Variables :**

- STV\_COLUMNS : StringVar pour récupérer la valeur du nombre de colonnes
- `i_columns` : Le nombre de colonnes à modifier

**5.22.2.19 vpp\_set\_nb\_jetons()**

```
def src.view.view_pageParametres.vpp_set_nb_jetons (
    int i_nb_jetons )
```

Mutateur pour le nombre de jetons requis.

Cette fonction modifie le nombre de jetons requis sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

**Précondition**

STV\_NB\_JETONS initialisé

**Paramètres**

<code>i_nb_jetons</code>	Le nouveau nombre de jetons requis
--------------------------	------------------------------------

**Postcondition**

STV\_NB\_JETONS est modifié

**Variables :**

- STV\_NB\_JETONS : StringVar pour récupérer la valeur du nombre de jetons requis
- `i_nb_jetons` : Le nombre de jetons requis à modifier

**5.22.2.20 vpp\_set\_nb\_rows()**

```
def src.view.view_pageParametres.vpp_set_nb_rows (
    int i_rows )
```

Mutateur pour le nombre de lignes.

Cette fonction modifie le nombre de lignes sélectionné par l'utilisateur. Elle est appelée lorsque l'utilisateur clique sur le bouton "Réinitialiser".

**Précondition**

STV\_ROWS initialisé

**Paramètres**

<code>i_rows</code>	Le nouveau nombre de lignes
---------------------	-----------------------------

**Postcondition**

STV\_ROWS est modifié

**Variables :**

- STV\_ROWS : StringVar pour récupérer la valeur du nombre de lignes
- i\_rows : Le nombre de lignes à modifier

## 5.23 Référence de l'espace de nommage tests

### Espaces de nommage

- `test_grid`  
*Teste le module puissanceQuatre.grid.*
- `test_puissanceQuatre`  
*Teste le module puissanceQuatre.puissanceQuatre.*

## 5.24 Référence de l'espace de nommage tests.test\_grid

Teste le module puissanceQuatre.grid.

### Fonctions

- `def tg_init_grille ()`  
*Teste la fonction pq\_init\_grille.*
- `def tg_test_all ()`  
*Lance tous les tests.*

### 5.24.1 Description détaillée

Teste le module puissanceQuatre.grid.

Ce module teste le module puissanceQuatre.grid.

### 5.24.2 Documentation des fonctions

#### 5.24.2.1 tg\_init\_grille()

```
def tests.test_grid.tg_init_grille ( )
```

Teste la fonction pq\_init\_grille.

#### Variables :

- *liste\_tailles* : liste des tailles de grille à tester
- *i\_boucle* : variable de boucle
- *i\_boucle\_2* : variable de boucle
- *grille* : grille de jeu

**Test** Vérifie que la grille est bien initialisée avec des 0 partout avec la bonne taille

Vérifie que toutes les combinaisons de 2 nombres de la liste *liste\_tailles* sont testées

#### 5.24.2.2 tg\_test\_all()

```
def tests.test_grid.tg_test_all ( )
```

Lance tous les tests.

## 5.25 Référence de l'espace de nommage tests.test\_puissanceQuatre

Teste le module puissanceQuatre.puissanceQuatre.

## Fonctions

- def `tp_verif_colonne` ()  
*Teste la fonction pq\_verif\_colonne.*
- def `tp_ajout_piece` ()  
*Teste la fonction pq\_ajout\_piece.*
- def `tp_victoire_ligne` ()  
*Test de la fonction pq\_victoire\_ligne.*
- def `tp_victoire_colonne` ()  
*Test de la fonction pq\_victoire\_colonne.*
- def `tp_victoire_diago` ()  
*Test de la fonction pq\_victoire\_diagonale.*
- def `tp_test_all` ()  
*Fonction qui lance tous les tests unitaires.*

### 5.25.1 Description détaillée

Teste le module puissanceQuatre.puissanceQuatre.

Ce module teste le module puissanceQuatre.puissanceQuatre.

### 5.25.2 Documentation des fonctions

#### 5.25.2.1 tp\_ajout\_piece()

```
def tests.test_puissanceQuatre.tp_ajout_piece ( )
```

Teste la fonction pq\_ajout\_piece.

##### Variables :

- *grille* : grille de jeu
- *ti\_coords* : tuple de coordonnées
- *i\_boucle* : variable de boucle

**Test** Vérifie que la fonction renvoie les bonnes coordonnées quand le joueur joue  
 Vérifie que la fonction renvoie les bonnes coordonnées quand le bot joue  
 Vérifie que la fonction renvoie les bonnes coordonnées quand le joueur joue normalement  
 Vérifie que la fonction ne renvoie pas de coordonnées quand on ne peut pas ajouter de pièce

#### 5.25.2.2 tp\_test\_all()

```
def tests.test_puissanceQuatre.tp_test_all ( )
```

Fonction qui lance tous les tests unitaires.

#### 5.25.2.3 tp\_verif\_colonne()

```
def tests.test_puissanceQuatre.tp_verif_colonne ( )
```

Teste la fonction pq\_verif\_colonne.

##### Variables :

- *grille* : grille de jeu
- *i\_boucle* : variable de boucle
- *i\_boucle\_2* : variable de boucle

**Test** Vérifie que la fonction renvoie True si la colonne est vide  
 Vérifie que la fonction renvoie False si la colonne est pleine  
 Vérifie que la fonction renvoie True si la colonne est presque pleine  
 Vérifie que la fonction renvoie True si la colonne est un peu remplie  
 Vérifie que la fonction renvoie False si la grille est pleine

#### 5.25.2.4 tp\_victoire\_colonne()

```
def tests.test_puissanceQuatre.tp_victoire_colonne ( )
```

Test de la fonction pq\_victoire\_colonne.

**Variables :**

- *grille* : Grille du jeu
- *i\_boucle* : Variable de boucle
- *ti\_coords* : Tuple de coordonnées

**Test** Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne

#### 5.25.2.5 tp\_victoire\_diago()

```
def tests.test_puissanceQuatre.tp_victoire_diago ( )
```

Test de la fonction pq\_victoire\_diagonale.

**Variables :**

- *grille* : Grille du jeu
- *i\_boucle* : Variable de boucle
- *i\_boucle\_2* : Variable de boucle
- *ti\_coords* : Tuple de coordonnées

**Test** Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne

#### 5.25.2.6 tp\_victoire\_ligne()

```
def tests.test_puissanceQuatre.tp_victoire_ligne ( )
```

Test de la fonction pq\_victoire\_ligne.

**Variables :**

- *grille* : Grille du jeu
- *i\_boucle* : Variable de boucle
- *ti\_coords* : Tuple de coordonnées

**Test** Vérifie que la fonction renvoie True si le joueur gagne

Vérifie que la fonction renvoie True si le bot gagne

Vérifie que la fonction renvoie False si personne ne gagne

# Index

- bu\_format\_bonus\_name
  - src.utils.bonus\_utils, [39](#)
- bu\_get\_bonus\_description
  - src.utils.bonus\_utils, [40](#)
- bu\_get\_bonus\_name
  - src.utils.bonus\_utils, [40](#)
- bu\_get\_bonuses
  - src.utils.bonus\_utils, [40](#)
- bu\_unformat\_bonus\_name
  - src.utils.bonus\_utils, [40](#)
- cm\_ended\_game
  - src.controller.ctrl\_main, [10](#)
- cm\_info
  - src.controller.ctrl\_main, [10](#)
- cm\_init
  - src.controller.ctrl\_main, [10](#)
- cm\_menu
  - src.controller.ctrl\_main, [11](#)
- cm\_page\_accueil
  - src.controller.ctrl\_main, [11](#)
- cm\_page\_bonus
  - src.controller.ctrl\_main, [12](#)
- cm\_page\_parameters
  - src.controller.ctrl\_main, [12](#)
- cm\_page\_play
  - src.controller.ctrl\_main, [12](#)
- cm\_quit
  - src.controller.ctrl\_main, [13](#)
- cm\_update
  - src.controller.ctrl\_main, [13](#)
- cm\_warning
  - src.controller.ctrl\_main, [13](#)
- cpa\_init
  - src.controller.ctrl\_pageAccueil, [14](#)
- cpa\_play
  - src.controller.ctrl\_pageAccueil, [14](#)
- cpb\_get\_bonuses
  - src.controller.ctrl\_PageBonus, [15](#)
- cpb\_get\_chosen\_bonus
  - src.controller.ctrl\_PageBonus, [15](#)
- cpb\_init
  - src.controller.ctrl\_PageBonus, [15](#)
- cpb\_show\_bonus\_description
  - src.controller.ctrl\_PageBonus, [16](#)
- cpb\_valider\_bonus
  - src.controller.ctrl\_PageBonus, [16](#)
- cpj\_bot\_play
  - src.controller.ctrl\_pageJeu, [17](#)
- cpj\_draw\_grid
  - src.controller.ctrl\_pageJeu, [17](#)
- cpj\_info\_turn
  - src.controller.ctrl\_pageJeu, [18](#)
- cpj\_init
  - src.controller.ctrl\_pageJeu, [18](#)
- cpj\_play
  - src.controller.ctrl\_pageJeu, [19](#)
- cpj\_put\_coin
  - src.controller.ctrl\_pageJeu, [19](#)
- cpj\_quit
  - src.controller.ctrl\_pageJeu, [20](#)
- cpj\_redo
  - src.controller.ctrl\_pageJeu, [20](#)
- cpj\_undo
  - src.controller.ctrl\_pageJeu, [20](#)
- cpj\_update\_grid
  - src.controller.ctrl\_pageJeu, [20](#)
- cpj\_use\_bonus
  - src.controller.ctrl\_pageJeu, [21](#)
- cpp\_askcolor
  - src.controller.ctrl\_pageParametres, [22](#)
- cpp\_custom\_load
  - src.controller.ctrl\_pageParametres, [22](#)
- cpp\_custom\_reset
  - src.controller.ctrl\_pageParametres, [22](#)
- cpp\_custom\_save
  - src.controller.ctrl\_pageParametres, [22](#)
- cpp\_init
  - src.controller.ctrl\_pageParametres, [23](#)
- cpp\_settings\_load
  - src.controller.ctrl\_pageParametres, [23](#)
- cpp\_settings\_reset
  - src.controller.ctrl\_pageParametres, [24](#)
- cpp\_settings\_save
  - src.controller.ctrl\_pageParametres, [24](#)
- cu\_colors\_too\_close
  - src.utils.colors\_utils, [41](#)
- cu\_hex\_to\_rgb
  - src.utils.colors\_utils, [41](#)
- cu\_rgb\_distance
  - src.utils.colors\_utils, [41](#)
- gp\_choose\_bonus
  - src.puissanceQuatre.gestionPartie, [29](#)
- gp\_gestion\_partie
  - src.puissanceQuatre.gestionPartie, [29](#)
- gp\_get\_player\_choice
  - src.puissanceQuatre.gestionPartie, [29](#)
- gp\_handle\_bot\_turn
  - src.puissanceQuatre.gestionPartie, [30](#)

- gp\_handle\_player\_turn
  - src.puissanceQuatre.gestionPartie, 30
- gp\_handle\_undo\_redo
  - src.puissanceQuatre.gestionPartie, 31
- gp\_handle\_victory
  - src.puissanceQuatre.gestionPartie, 31
- gp\_show\_rules
  - src.puissanceQuatre.gestionPartie, 31
- gp\_use\_bonus
  - src.puissanceQuatre.gestionPartie, 31
- main
  - src.main, 25
- p4b\_flip\_grid
  - src.puissanceQuatre.bonus, 25
- p4b\_invert\_grid
  - src.puissanceQuatre.bonus, 26
- p4b\_no\_bonus
  - src.puissanceQuatre.bonus, 26
- p4b\_random\_bonus
  - src.puissanceQuatre.bonus, 27
- p4b\_random\_placement
  - src.puissanceQuatre.bonus, 27
- p4b\_remove\_full\_line
  - src.puissanceQuatre.bonus, 27
- p4b\_use\_min\_max
  - src.puissanceQuatre.bonus, 28
- pq\_ajout\_piece
  - src.puissanceQuatre.puissanceQuatre, 34
- pq\_apply\_gravity
  - src.puissanceQuatre.grid, 32
- pq\_init\_grille
  - src.puissanceQuatre.grid, 33
- pq\_minmax
  - src.puissanceQuatre.puissanceQuatre, 35
- pq\_partie\_finie
  - src.puissanceQuatre.puissanceQuatre, 35
- pq\_print\_grille
  - src.puissanceQuatre.grid, 33
- pq\_redo
  - src.puissanceQuatre.puissanceQuatre, 36
- pq\_reset\_grille
  - src.puissanceQuatre.grid, 33
- pq\_undo
  - src.puissanceQuatre.puissanceQuatre, 36
- pq\_verif\_colonne
  - src.puissanceQuatre.puissanceQuatre, 36
- pq\_victoire
  - src.puissanceQuatre.puissanceQuatre, 37
- pq\_victoire\_colonne
  - src.puissanceQuatre.puissanceQuatre, 37
- pq\_victoire\_diago
  - src.puissanceQuatre.puissanceQuatre, 38
- pq\_victoire\_ligne
  - src.puissanceQuatre.puissanceQuatre, 38
- src, 9
- src.controller, 9
  - src.controller.ctrl\_main, 9
    - cm\_ended\_game, 10
    - cm\_info, 10
    - cm\_init, 10
    - cm\_menu, 11
    - cm\_page\_accueil, 11
    - cm\_page\_bonus, 12
    - cm\_page\_parameters, 12
    - cm\_page\_play, 12
    - cm\_quit, 13
    - cm\_update, 13
    - cm\_warning, 13
  - src.controller.ctrl\_pageAccueil, 14
    - cpa\_init, 14
    - cpa\_play, 14
  - src.controller.ctrl\_PageBonus, 15
    - cpb\_get\_bonuses, 15
    - cpb\_get\_chosen\_bonus, 15
    - cpb\_init, 15
    - cpb\_show\_bonus\_description, 16
    - cpb\_valider\_bonus, 16
  - src.controller.ctrl\_pageJeu, 17
    - cpj\_bot\_play, 17
    - cpj\_draw\_grid, 17
    - cpj\_info\_turn, 18
    - cpj\_init, 18
    - cpj\_play, 19
    - cpj\_put\_coin, 19
    - cpj\_quit, 20
    - cpj\_redo, 20
    - cpj\_undo, 20
    - cpj\_update\_grid, 20
    - cpj\_use\_bonus, 21
  - src.controller.ctrl\_pageParametres, 21
    - cpp\_askcolor, 22
    - cpp\_custom\_load, 22
    - cpp\_custom\_reset, 22
    - cpp\_custom\_save, 22
    - cpp\_init, 23
    - cpp\_settings\_load, 23
    - cpp\_settings\_reset, 24
    - cpp\_settings\_save, 24
- src.main, 24
  - main, 25
- src.puissanceQuatre, 25
- src.puissanceQuatre.bonus, 25
  - p4b\_flip\_grid, 25
  - p4b\_invert\_grid, 26
  - p4b\_no\_bonus, 26
  - p4b\_random\_bonus, 27
  - p4b\_random\_placement, 27
  - p4b\_remove\_full\_line, 27
  - p4b\_use\_min\_max, 28
- src.puissanceQuatre.gestionPartie, 28
  - gp\_choose\_bonus, 29
  - gp\_gestion\_partie, 29
  - gp\_get\_player\_choice, 29
  - gp\_handle\_bot\_turn, 30

- gp\_handle\_player\_turn, 30
- gp\_handle\_undo\_redo, 31
- gp\_handle\_victory, 31
- gp\_show\_rules, 31
- gp\_use\_bonus, 31
- src.puissanceQuatre.grid, 32
  - pq\_apply\_gravity, 32
  - pq\_init\_grille, 33
  - pq\_print\_grille, 33
  - pq\_reset\_grille, 33
- src.puissanceQuatre.puissanceQuatre, 34
  - pq\_ajout\_piece, 34
  - pq\_minmax, 35
  - pq\_partie\_finie, 35
  - pq\_redo, 36
  - pq\_undo, 36
  - pq\_verif\_colonne, 36
  - pq\_victoire, 37
  - pq\_victoire\_colonne, 37
  - pq\_victoire\_diago, 38
  - pq\_victoire\_ligne, 38
- src.utils, 39
- src.utils.bonus\_utils, 39
  - bu\_format\_bonus\_name, 39
  - bu\_get\_bonus\_description, 40
  - bu\_get\_bonus\_name, 40
  - bu\_get\_bonuses, 40
  - bu\_unformat\_bonus\_name, 40
- src.utils.colors\_utils, 40
  - cu\_colors\_too\_close, 41
  - cu\_hex\_to\_rgb, 41
  - cu\_rgb\_distance, 41
- src.view, 42
- src.view.view\_main, 42
  - vm\_init, 42
  - vm\_menu, 43
  - vm\_message\_game\_ended, 43
  - vm\_message\_info, 43
  - vm\_message\_warning, 43
  - vm\_quit, 44
  - vm\_remove\_frame, 44
  - vm\_update, 44
- src.view.view\_pageAccueil, 45
  - vpa\_destroy, 45
  - vpa\_init, 45
- src.view.view\_pageBonus, 45
  - vpb\_get\_bonus, 46
  - vpb\_get\_frame, 46
  - vpb\_init, 46
  - vpb\_show\_bonus\_description, 47
- src.view.view\_pageJeu, 47
  - vpj\_destroy, 48
  - vpj\_disable\_bonus, 48
  - vpj\_draw\_grid, 48
  - vpj\_get\_frame, 48
  - vpj\_get\_grid\_cell, 49
  - vpj\_init\_page\_jeu, 49
  - vpj\_set\_info, 50
  - vpj\_show\_coin, 50
- src.view.view\_pageParametres, 50
  - vpp\_askcolor, 51
  - vpp\_get\_bot\_color, 51
  - vpp\_get\_difficulty, 52
  - vpp\_get\_grid\_color, 52
  - vpp\_get\_joueur\_color, 52
  - vpp\_get\_nb\_columns, 52
  - vpp\_get\_nb\_jetons, 53
  - vpp\_get\_nb\_rows, 53
  - vpp\_init, 53
  - vpp\_init\_custom, 54
  - vpp\_init\_settings, 54
  - vpp\_reset\_customs, 54
  - vpp\_reset\_settings, 55
  - vpp\_set\_bot\_color, 55
  - vpp\_set\_difficulty, 55
  - vpp\_set\_grid\_color, 56
  - vpp\_set\_joueur\_color, 56
  - vpp\_set\_nb\_columns, 56
  - vpp\_set\_nb\_jetons, 57
  - vpp\_set\_nb\_rows, 57
- tests, 58
- tests.test\_grid, 58
  - tg\_init\_grille, 58
  - tg\_test\_all, 58
- tests.test\_puissanceQuatre, 58
  - tp\_ajout\_piece, 59
  - tp\_test\_all, 59
  - tp\_verif\_colonne, 59
  - tp\_victoire\_colonne, 59
  - tp\_victoire\_diago, 60
  - tp\_victoire\_ligne, 60
- tg\_init\_grille
  - tests.test\_grid, 58
- tg\_test\_all
  - tests.test\_grid, 58
- tp\_ajout\_piece
  - tests.test\_puissanceQuatre, 59
- tp\_test\_all
  - tests.test\_puissanceQuatre, 59
- tp\_verif\_colonne
  - tests.test\_puissanceQuatre, 59
- tp\_victoire\_colonne
  - tests.test\_puissanceQuatre, 59
- tp\_victoire\_diago
  - tests.test\_puissanceQuatre, 60
- tp\_victoire\_ligne
  - tests.test\_puissanceQuatre, 60
- vm\_init
  - src.view.view\_main, 42
- vm\_menu
  - src.view.view\_main, 43
- vm\_message\_game\_ended
  - src.view.view\_main, 43
- vm\_message\_info
  - src.view.view\_main, 43

- vm\_message\_warning
  - src.view.view\_main, [43](#)
- vm\_quit
  - src.view.view\_main, [44](#)
- vm\_remove\_frame
  - src.view.view\_main, [44](#)
- vm\_update
  - src.view.view\_main, [44](#)
- vpa\_destroy
  - src.view.view\_pageAccueil, [45](#)
- vpa\_init
  - src.view.view\_pageAccueil, [45](#)
- vpb\_get\_bonus
  - src.view.view\_pageBonus, [46](#)
- vpb\_get\_frame
  - src.view.view\_pageBonus, [46](#)
- vpb\_init
  - src.view.view\_pageBonus, [46](#)
- vpb\_show\_bonus\_description
  - src.view.view\_pageBonus, [47](#)
- vpj\_destroy
  - src.view.view\_pageJeu, [48](#)
- vpj\_disable\_bonus
  - src.view.view\_pageJeu, [48](#)
- vpj\_draw\_grid
  - src.view.view\_pageJeu, [48](#)
- vpj\_get\_frame
  - src.view.view\_pageJeu, [48](#)
- vpj\_get\_grid\_cell
  - src.view.view\_pageJeu, [49](#)
- vpj\_init\_page\_jeu
  - src.view.view\_pageJeu, [49](#)
- vpj\_set\_info
  - src.view.view\_pageJeu, [50](#)
- vpj\_show\_coin
  - src.view.view\_pageJeu, [50](#)
- vpp\_askcolor
  - src.view.view\_pageParametres, [51](#)
- vpp\_get\_bot\_color
  - src.view.view\_pageParametres, [51](#)
- vpp\_get\_difficulty
  - src.view.view\_pageParametres, [52](#)
- vpp\_get\_grid\_color
  - src.view.view\_pageParametres, [52](#)
- vpp\_get\_joueur\_color
  - src.view.view\_pageParametres, [52](#)
- vpp\_get\_nb\_columns
  - src.view.view\_pageParametres, [52](#)
- vpp\_get\_nb\_jetons
  - src.view.view\_pageParametres, [53](#)
- vpp\_get\_nb\_rows
  - src.view.view\_pageParametres, [53](#)
- vpp\_init
  - src.view.view\_pageParametres, [53](#)
- vpp\_init\_custom
  - src.view.view\_pageParametres, [54](#)
- vpp\_init\_settings
  - src.view.view\_pageParametres, [54](#)
- vpp\_reset\_customs
  - src.view.view\_pageParametres, [54](#)
- vpp\_reset\_settings
  - src.view.view\_pageParametres, [55](#)
- vpp\_set\_bot\_color
  - src.view.view\_pageParametres, [55](#)
- vpp\_set\_difficulty
  - src.view.view\_pageParametres, [55](#)
- vpp\_set\_grid\_color
  - src.view.view\_pageParametres, [56](#)
- vpp\_set\_joueur\_color
  - src.view.view\_pageParametres, [56](#)
- vpp\_set\_nb\_columns
  - src.view.view\_pageParametres, [56](#)
- vpp\_set\_nb\_jetons
  - src.view.view\_pageParametres, [57](#)
- vpp\_set\_nb\_rows
  - src.view.view\_pageParametres, [57](#)