

# **Puissance4**

BENOIT Guilleme DORCZYNSKI Lucas

Version

29/01/2020



# Table of Contents

Table of contents



## Usage :

Compilez les sources : `make ./svc ./clt {ip address (ex:127.0.0.1)} {port} ./clt {ip address (ex:127.0.0.1)} {port}`

Le port est modifiable via `PORT_SVC` dans **shared.h**, fixé à 7000 par défaut Le port `PORT_SVC+1` sera utilisé par l'host L'adresse est modifiable via `INADDR_SVC` dans **shared.h**, fixé à 127.0.0.1 par défaut

**Ce projet utilise Doxygen <http://www.doxygen.nl/> Utilisez doxygen configdox pour générer la doc. <<<<<<< HEAD:README.md**

Si Doxygen n'est pas installé, utilisez `sudo apt-get install doxygen`

master:README.txt

Pour ce projet nous utiliserons le langage C

Le plateau est constitué de 7 colonnes et de 5 lignes chaque joueur, à tour de rôle choisi une colonne dans laquelle mettre un jeton qui se place à l'endroit le plus bas disponible. Un joueur gagne la partie s'il possède 4 jetons alignés (diagonales incluses).

Le serveur doit d'abord attendre la connection de deux joueurs pour les mettre en communication, il désigne un 'host' et un 'client' à qui il transmet l'ip de l'host ensuite le serveur peut recréer une communication en attendant d'autre joueur.

Un client (le premier à s'être connecté) est désigné comme 'Host' il lancera le mode serveur. le deuxième client lancera le mode client. L'host crée un canal d'écoute en attendant la connection du client, le client se connecte avec les infos transmises par le serveur(adresse IP). La partie peut commencer: les deux joueurs affichent le plateau, le joueur désigné comme 'Host' joue en premier. Pour placer un jeton le joueur doit entrer le numéro de la colonne voulue. Une fois le coup joué 'l'Host' vérifie les conditions de victoire. Si elles ne sont pas remplies la partie continue jusqu'à ce qu'elles le soient. Lorsque les conditions sont remplies le joueur dont ce n'est pas le tour voit dans un premier temps le plateau de jeu puis un message de défaite.

Une fois la partie terminée 'l'Host ' ferme la socket d'écoute et le client quitte l'application!;

dans la doc **server.c** sera nommé le serveur de matchmaking le premier client connecté sera nommé l'host le second client sera nommé le client

# File Index

## File List

Here is a list of all documented files with brief descriptions:

<b>client.c (Client de jeu pour Puissance 4 )</b> .....	4
<b>P4.c (Code du jeu Puissance4 )</b> .....	7
<b>server.c (Server de matchmaking pour Puissance 4 )</b> .....	9
<b>shared.h</b> .....	Erreur : source de la référence non trouvée
<b>tableau.c (Génération du tableau du Puissance4 )</b> .....	10

# File Documentation

## client.c File Reference

Client de jeu pour Puissance 4.

```
#include "shared.h"
```

### Functions

int **calculNBJetons** (int, int, int, int)

*Fonction permettant de compter le nombres de jetons alignés dans chaque direction.*

void **clientMode** (char addr[INET\_ADDRSTRLEN])

*Crée une connexion entre le client et l'host à partir de l'adresse fourni par le serveur de matchmaking.*

void **com** (int, struct sockaddr\_in)

*Définie le mode des deux clients et receptionne les infos de connexion à l'host.*

void **creationTableau** ()

*Fonction permettant de créer la grille de jeu.*

void **game** (int, int, int)

*Déroulement du jeu.*

int **grillePleine** ()

*Fonction permettant de vérifier si on peut encore jouer un coup si la grille est pleine fin de partie.*

int **jouable** (int)

*Fonction permettant de vérifier si on peut jouer un coup sans superposition de jeton.*

int **partieGagnante** (int, int)

*Fonction permettant de vérifier les conditions de victoire.*

int **saisirCoup** ()

*Demande au joueur dans quelle colonne il veut jouer et vérifie la possibilité de jouer.*

void **serverMode** ()

*Transforme le client en host.*

int **main** (int c, char \*v[])

---

### Detailed Description

Client de jeu pour Puissance 4.

#### Author:

Lucas Dorczynski & Guilleme Benoit

#### Date:

decembre 2019

#### Version:

1.0

---

### Function Documentation

int **calculNBJetons** (int *c*, int *l*, int *dirV*, int *dirH*)

Fonction permettant de compter le nombres de jetons alignés dans chaque direction.

**Version:**

1.0

0

**void clientMode (char *addr*[INET\_ADDRSTRLEN])**

Crée une connexion entre le client et l'host à partir de l'adresse fourni par le serveur de matchmaking.

**Version:**

1.0

**void com (int *sd*, struct sockaddr\_in *svc*)**

Définit le mode des deux clients et receptionne les infos de connexion à l'host.

**Remarks:**

Le port de l'host n'est pas libre, il est fixé à PORT\_SVC+1

**Version:**

1.0

**void creationTableau ()**

Fonction permettant de créer la grille de jeu.

**Version:**

1.0

1

**void game (int *joueur*, int *mode*, int *sock*)**

Déroulement du jeu.

**Version:**

1.0

2

**int grillePleine ()**

Fonction permettant de vérifier si on peut encore jouer un coup si la grille est pleine fin de partie.



**Version:**

1.0

3

**int jouable (int x)**

Fonction permettant de vérifier si on peut jouer un coup sans superposition de jeton.

**Version:**

1.0

**int partieGagnante (int c, int l)**

Fonction permettant de vérifier les conditions de victoire.

**Version:**

1.0

**Remarks:**

la première condition vérifie les jetons horizontalement, la seconde verticalement, la troisième diagonalement ( haut-droit vers bas gauche) la quatrième diagonalement (haut gauche- bas droite)

4

**int saisirCoup ()**

Demande au joueur dans quelle colonne il veut jouer et vérifie la possibilité de jouer.

**Version:**

1.0

5

**void serverMode ()**

Transforme le client en host.

**Version:**

1.0

6

## P4.c File Reference

Code du jeu Puissance4.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

### Macros

```
#define LIG 6
#define COL 7
```

### Functions

```
int calculNBJetons (int, int, int, int)
void creationTableau ()
```

*Fonction permettant de créer la grille de jeu.*

```
int grillePleine ()
int jouable (int)
int partieGagnante (int, int)
int saisirCoup ()
int main ()
```

### Variables

```
int i
int j
char tab [COL][LIG]
int remplissage [7]
int choixC
```

---

### Detailed Description

Code du jeu Puissance4.

Fonction permettant de placer un jeton.

Fonction permettant de vérifier les conditions de victoire.

Fonction permettant de vérifier si on peut jouer un coup pas de superposition de jeton.

Fonction permettant de vérifier si on peut encore jouer un coup si la grille est pleine fin de partie.

Fonction permettant de créer la grille de jeu.

Fonction permettant de calculer les jetons qui se suivent dans toutes les directions.

### Author:

Lucas Dorczynski & Guilleme Benoit

### Date:

decembre 2019

### Version:

1.0

### Remarks:

Ce fichier est un fichier de travail -> il n'est pas importé ou utilisé dans le projet final  
-> fonction init

### Version:

1.0

→ fonction créationTableau

**Version:**

1.0

→ fonction grillePleine

**Version:**

1.0

→ fonction jouable

**Version:**

1.0

→ fonction partieGagnante

**Version:**

1.0

**Remarks:**

la première condition vérifie les jetons horizontalement, la seconde verticalement, la troisième diagonalement ( haut-droit vers bas gauche) la quatrième diagonalement (haut gauche- bas droite)

→ fonction SaisirCoup

**Version:**

1.0

**Remarks:**

appel la fonction **jouable()**

---

## Function Documentation

### void creationTableau ()

Fonction permettant de créer la grille de jeu.

**Version:**

1.0

## server.c File Reference

Server de matchmaking pour Puissance 4.

```
#include "shared.h"
```

### Functions

```
void com (int, struct sockaddr_in, int, struct sockaddr_in)
```

*Définie le mode des deux clients, leur envoie les infos de connexion client -> host.*

```
int main ()
```

---

### Detailed Description

Server de matchmaking pour Puissance 4.

### Author:

Lucas Dorczynski & Guilleme Benoit

### Date:

decembre 2019

### Version:

1.0

---

### Function Documentation

```
void com (int sd, struct sockaddr_in clt, int sd2, struct sockaddr_in clt2)
```

Définie le mode des deux clients, leur envoie les infos de connexion client -> host.

### Remarks:

OK est fixé à "1", Ok2 à "2" et ACK à "-1", voir **shared.h**

Le port de l'host n'est pas libre, il est fixé à PORT\_SVC+1

### Version:

1.0

## tableau.c File Reference

Génération du tableau du Puissance4.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

### Macros

```
#define LIG 16
#define COL 15
```

### Functions

```
void creationTableau ()
    Fonction permettant de créer la grille de jeu.
void demande ()
void placementJeton (choixL, choixC)
int main ()
```

### Variables

```
int i
int j
char tab [COL][LIG]
int choixL
int choixC
```

---

### Detailed Description

Génération du tableau du Puissance4.

### Author:

Lucas Dorczynski & Guilleme Benoit

### Date:

decembre 2019

### Version:

1.0

### Remarks:

Ce fichier est un fichier de travail -> il n'est pas importé ou utilisé dans le projet final  
Deprecated, n'est pas docé : les fonctions utilisées sont docé dans les autres fichiers

---

### Function Documentation

#### void **creationTableau** ()

Fonction permettant de créer la grille de jeu.

#### Version:

1.0

**void placementJeton (choixL , choixC )**

Condition

# **Index**

INDE