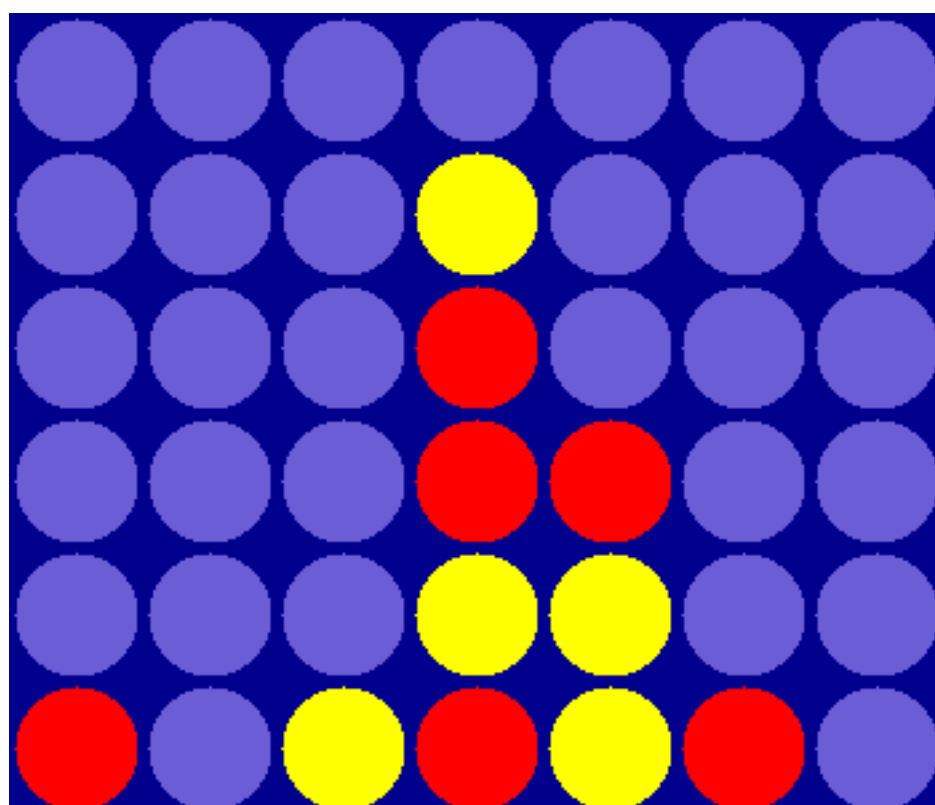


# Projet de Programmation : Puissance 4

Borges Daniel ( [dborges@etudiant.univ-mlv.fr](mailto:dborges@etudiant.univ-mlv.fr) )

24 mai 2005



Au joueur Jaune de jouer

A R Q

# 1 Description générale

Puissance 4 est un jeu à deux joueurs composé d'une grille dans laquelle les joueurs font tomber, chacun à leur tour, des pions dans les colonnes, remplissant ainsi la case libre la plus basse de la colonne correspondante. Le gagnant est le premier joueur qui arrive à aligner 4 pions de sa couleur dans n'importe quelle direction.

## 2 Manuel d'utilisation

### 2.1 Comment jouer

Le jeu commence par afficher une grille de cases vides. Pour ajouter un pion dans une colonne, il suffit de cliquer sur la colonne correspondante. Quand 4 pions sont alignés, le jeu s'arrête et affiche un message indiquant qui est le joueur qui vient de gagner.

En dessous de la grille se trouve une barre de status indiquant :

- Quel est le joueur qui doit actuellement jouer. Exemple "Au joueur Rouge de jouer".
- Un bouton vert marqué d'un "A" (pour Annuler). En cliquant sur ce bouton, on annule le dernier coup joué, permettant ainsi de le rejouer.
- Un bouton jaune marqué d'un "R" (pour Recommencer). Ce bouton lance une nouvelle partie.
- Un bouton rouge marqué d'un "Q" (pour Quitter). Ce bouton quitte le programme.

Pour ces deux derniers boutons, une confirmation sera requise. Par exemple, en cliquant sur le bouton Recommencer, un message demandant si l'on souhaite réellement recommencer s'affichera. Pour confirmer il suffira de cliquer sur le bouton vert marqué d'un "O" (Oui), pour infirmer, de cliquer sur le bouton rouge marqué d'un "N" (Non).

### 2.2 Options de lancement

Le programme possède plusieurs options de lancement. Pour afficher la liste de toutes les options, il suffit de lancer le programme avec l'option "-a" :

```
$ ./puissance\_4 -a
```

Option valides :

- d x : adresse du serveur X
- h x : hauteur de la grille de jeu
- l x : largeur de la grille de jeu
- p x : nombre de pions à aligner pour gagner

-t x : taille ( en pixels ) des cases de la grille  
-a : cette page d'aide

## 3 Manuel du programmeur

### 3.1 Type “Donnees”

Dans `puissance_4.h` est défini un type 'Donnees'. C'est une structure contenant toutes les informations nécessaires au bon fonctionnement du jeu.

`int **grille` : La grille de jeu, une tableau d'entiers à deux dimensions. Chaque case contient soit VIDE ( qui vaut 0 ), soit ROUGE ( qui vaut 1 ), soit JAUNE ( qui vaut 2 ). Ces noms correspondent au contenu de la case.

Cette grille est de largeur “lgrille” et de hauteur “hgrille”, données que l'on retrouve dans `Donnees`. lgrille vaut 7 par défaut et hgrille 6. Ce sont les dimensions du puissance 4 classique.

`int *historique` : l'historique de tous les coups joués. A chaque coup joué, la colonne jouée est notée dans l'historique. L'historique est un tableau d'entiers de taille lgrille x hgrille.

`int x` et `int y` : Ce sont les coordonnées du dernier coup joué.

`int tour` : C'est le numéro du tour actuellement joué, le premier tour est le tour 0. On peut ainsi retrouver la colonne qui a été jouée à n'importe quel moment du jeu par `historique[tour]`.

`int joueur` : Contient le joueur dont c'est le tour actuellement. Cette information sert pour placer les nouveaux pions.

`int puissance` : C'est le nombre de pions à aligner nécessaire pour gagner la partie.

`int tcases` : C'est la taille en pixels de chaque case à l'écran. 50 pixels par défaut.

`char *display` : C'est l'adresse du serveur graphique. Par défaut `Rr display` vaut “:0”.

`MlvType ecran` : C'est la variable contenant les données nécessaires au bon fonctionnement de la `mlvlib` utilisée pour l'affichage.

### 3.2 Initialisation

Au début du programme, 3 fonctions d'initialisation du programme sont appelées. La première option, modifie la structure de jeu pour prendre en compte les paramètres de la ligne de commande.

`alloc_mem` va allouer la quantité de mémoire nécessaire pour la grille de jeu ainsi que pour l'historique. Cette fonction est nécessaire puisqu'il est possible de choisir la taille de la grille au lancement, un simple tableau n'aurait pas suffi.

`init_partie` va remplir le tableau de “case vides” et mettre le tour à 0 pour que la partie commence. Le joueur rouge est aussi mis comme premier joueur. La fenêtre du jeu est créée est appelée “Puissance 4”.

### 3.3 Jeu

Toutes les fonctions de jeu se trouvent dans le fichier `puissance_4.c`.

Suite à l'initialisation, une boucle infinie de la fonction `jeu` est lancée. Celle ci sert à effectuer plusieurs parties sans avoir à relancer le programme.

`jeu` commence par lancer une boucle qui aura deux fonctions, la première d'afficher le plateau, la seconde de demander au joueur dont c'est le tour de jouer. Ce joueur aura plusieurs options : jouer une colonne, quitter, recommencer la partie ou encore annuler le dernier coup joué. Toutes ces actions se font en cliquant sur la fenêtre de jeu.

Si le joueur choisit de jouer, la fonction `jouer` est appelée. Celle ci va ajouter un pion, noter la colonne jouée dans l'historique, passer au tour suivant, noter les coordonnées du coup joué, vérifier qu'il n'y a pas de **puissance 4**. Si c'est le cas, le joueur suivant est mis à 0 et la partie s'arrête. Sinon, on passe au joueur suivant.

La fonction `puissance_4` est chargée de vérifier s'il n'y a pas d'alignement de pions gagnants. Pour cela, elle utilise la fonction `nombre_pions` qui retourne le nombre de pions de la couleur du joueur actuel dans la direction demandée. `puissance_4` fait donc 7 appels à la fonction `nombre_pions` pour vérifier qu'il n'y a pas de puissance 4 dans les 4 directions possibles ( haut-bas, gauche-droite et les diagonales ). A noter que `puissance_4` n'appelle pas `nombre_pions` en lui demandant de vérifier vers le haut puisqu'il n'y a forcément pas de pions dans cette direction, on joue par cette direction.

L'ajout d'un pion dans une colonne se fait en recherchant le pion le plus haut de la colonne, au travers de la fonction `recherche_pion`, et un pion de la couleur du joueur courant est mis sur la case au dessus, si ce n'est pas le haut de la grille.

Dans le cas d'une annulation du dernier coup joué, `annuler` est la fonction appelée : elle décrémente le tour, remet une "case vide" à l'endroit du dernier pion joué à l'aide des coordonnées du dernier pion joué et repasse au joueur précédent.

### 3.4 Entrée/Sortie

Les fonctions d'entrée/sortie sont dans `es.c`.

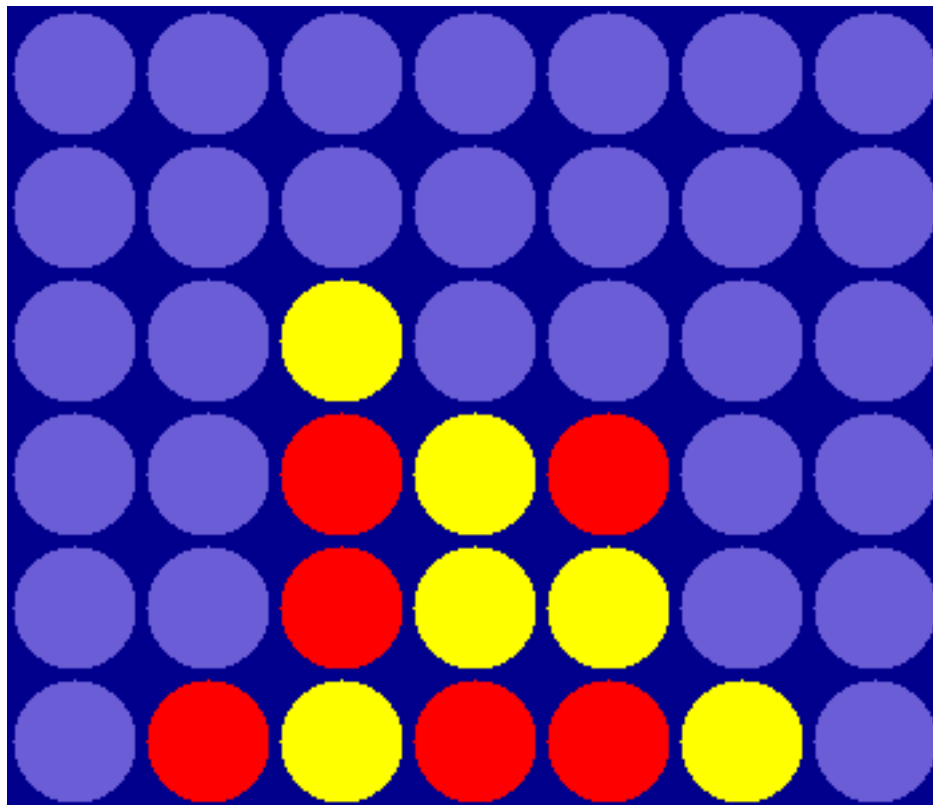
La fonction `entree` sert à déterminer l'action demandée par le joueur en cliquant sur l'écran. Un appel à la fonction `mouse_wait` de la `MLvLib` permet de déterminer où le joueur a cliqué. La fonction retournera la colonne sur lequel le joueur a joué ou -1 si le joueur a cliqué sur l'icone quitter, -2 pour celle de recommencer et -3 pour celle d'annulation.

Pour rejouer une partie, la fonction `rejouer` demande d'abord une confirmation avec la fonction `affichage_question` et appelle ensuite la fonction `reponse` pour connaître la réponse. Si la réponse est oui, la partie est réinitialisée : on détruit la fenêtre et on appelle à nouveau `init_partie`.

La fonction `quitter` a un fonctionnement similaire avec le résultat attendu.

La fonction `reponse` fait un appel à la fonction `mouse_wait` et détermine ainsi sur quelle icône le joueur a cliqué.

La fonction `options` utilise la fonction `getopt` ( disponible dans `getopt.h` ) pour mettre à jour la structure de jeu en fonction des paramètres passées à la ligne de commande.



Le joueur Jaune à gagné

### 3.5 Affichage

Les fonctions d’affichage sont dans `affichage.c`.

L’affichage est réalisé par la fonction `affichage` qui à chaque fois qu’on l’appelle, affiche les pions, la barre de status et met à jour l’affichage avec `display_window`.

`affichage_pions` est la fonction utilisée pour afficher tous les pions si la partie est au tour 0 ( début du jeu ) ou le dernier pion joué dans les autres cas. Pour afficher chaque pion, `affichage_pion` calcule le rayon du pion en fonction de la taille de la case, 90% de la taille de la case, calcule l’abscisse et l’ordonnée du pion et dessine un cercle de couleur avec la fonction `draw_filled_circle`.

`affichage_status` affiche la barre de status, c’est à dire : un message indiquant quel est le joueur qui doit actuellement jouer et les 3 icones “A”, “R”, “Q”, respectivement annuler, recommencer et quitter.

`affichage_message` affiche le message passé en argument dans la barre de status.

`affichage_bouton` affiche un bouton à la position demandée, en partant de la droite. Exemple, pour le bouton R de recommencer, sa position est 2.

`affichage_question` affiche le message passé en argument ainsi que 2 icones O et N pour oui et non.

`gagner` affiche un message indiquant le joueur qui vient de gagner la partie et attend que le joueur clique au hasard sur l’écran pour relancer l’exécution du programme.

`aide` affiche sur la sortie standard un message indiquant les options du programme.