

# Puissance 4

## Objectifs

- Manipuler les types utilisateurs ;
- Comprendre, spécifier et définir des sous-programmes ;
- Raffiner, toujours !

## 1 Consignes

Le travail sera réalisé en équipes de 4 personnes en utilisant un document partagé entre les membres de l'équipe et l'enseignant.

Voici l'échéancier à suivre :

1. 10 minutes : **lecture individuelle** de la totalité du sujet,
  2. 15 minutes : réalisation de l'**exercice 1**. Après une réflexion individuelle de 5 minutes, les idées individuelles sont mises en commun et discutées pour aboutir à une solution commune qui sera adoptée par toute l'équipe pour la suite du travail.
  3. 30 minutes : **spécification des sous-programmes** (exercices 2 à 9)
    - 5 minutes : les équipiers se distribuent les exercices de 2 à 9. Les exercices 3, 6, 8 et 9 doivent être traités par des équipiers différents (si la taille de l'équipe le permet).
    - 10 minutes : chaque membre de l'équipe écrit la spécification des sous-programmes qui lui ont été attribués et identifie les tests qu'il faudrait faire.
    - 15 minutes : présentation des spécifications aux autres équipiers et discussions pour les améliorer.
  4. 45 minutes : **raffinage des sous-programmes**
    - 15 minutes : Chaque équipier écrit les premiers niveaux de raffinages des sous-programmes qui lui ont été attribués.
    - 30 minutes : Présentation des raffinages aux autres équipiers et discussions pour les améliorer.
- On utilisera la grille d'évaluation des raffinages pour évaluer la qualité des raffinages. Les 4 premières colonnes doivent être copiées dans votre document et complétés (une seule grille pour l'ensemble des raffinages).
5. 5 minutes : Répondre à la question « était-il logique de définir les types au début ? ».

## 2 Puissance 4

Le jeu « Puissance 4 » est proposé par MB Jeux. Sur la boîte, on peut lire les indications suivantes.

Puissance 4 est un jeu de stratégie verticale passionnant et plein d'astuce. C'est un jeu facile à apprendre et amusant à jouer. Ses règles sont simples. Chaque joueur essaie de faire une rangée de quatre pions dans le cadre – horizontalement, verticalement ou diagonalement – tout en essayant d'empêcher son adversaire de faire de même. Croyez-moi ce n'est pas aussi facile que cela en a l'air ! La position verticale du jeu demande au joueur beaucoup de concentration et de réflexion.

### **Montage :**

1. Suivez les instructions illustrées pour l'assemblage des parties en plastiques (non reproduites ici).
2. Placez le jeu entre les joueurs.
3. Chaque joueur choisit une couleur de pions.

### **But du jeu :**

Être le premier joueur à placer quatre de ses pions sur une ligne horizontale, verticale ou diagonale continue.

### **Règles du jeu :**

1. Choisissez le premier joueur. Le joueur qui commence la première partie sera le deuxième joueur au cours de la deuxième partie.
2. À tour de rôle, chaque joueur fait tomber un de ses pions dans une des fentes au sommet de la grille.
3. Le jeu continue jusqu'à ce qu'un des joueurs ait un alignement continu de quatre pions de sa couleur. L'alignement peut être vertical, horizontal ou en diagonale.
4. Pour vider la grille, poussez la barre de retenue qui se trouve au bas de celle-ci et les pions tomberont. Vous pouvez maintenant commencer la partie suivante.

### **Exercice 1 : Préparation du jeu**

Le jeu « puissance 4 » se joue sur une grille verticale de six lignes et sept colonnes avec 21 pions rouges et 21 pions jaunes.

Définir les types nécessaires pour représenter la grille de jeu.

### **Exercice 2 : Créer une grille**

Pour commencer chaque partie, il faut commencer par créer une grille. Écrire un sous-programme qui réalise cette opération.

### **Exercice 3 : Afficher la grille**

Plusieurs affichages sont possibles (figure 1). C'est l'affichage 1 qui sera réalisé mais les autres

pourraient l'être aussi (le joueur pourrait choisir l'affichage qu'il souhaite). Il fait apparaître les numéros de colonnes pour expliciter la convention à l'utilisateur. Les jetons de couleur rouge sont représentés par un astérisque (\*) et les jetons jaunes par la lettre o minuscule. Dans le reste du sujet, l'affichage 3 sera utilisé pour faciliter la localisation d'une case grâce aux numéros de lignes en plus des numéros de colonnes.

Écrire un sous-programme qui affiche la grille conformément à l'affichage 1.

	1	2	3	4	5	6	7
*	6		*		6		
0	5		0		5		
0 0 *	4	0 0	*	4			
* 0 0 0	3	*	0	0 0	3		
* 0 * 0 *	2	*	0 *	0 *	2		
* 0 * * 0 0 *	1	*	0 *	*	0 0 *	1	
1 2 3 4 5 6 7	1	2	3	4	5	6	7
affichage 1	affichage 3						
	0	1	2	3	4	5	6
*	F	X		F			
0	E	0		E			
0 0 *	D	0 0	X	D			
* 0 0 0	C	X 0	0 0	C			
* 0 * 0 *	B	X 0 X	0 X	B			
* 0 * * 0 0 *	A	X 0 X X	0 0 X	A			
-----	0	1	2	3	4	5	6
affichage 2	affichage 4	affichage 5					
	1	2	3	4	5	6	7
	- -   - -   - -   - -   - -   - -   - -						
6	X	6					
	- -   - -   - -   - -   - -   - -   - -						
5	0	5					
	- -   - -   - -   - -   - -   - -   - -						
4	0   0     X	4					
	- -   - -   - -   - -   - -   - -   - -						
3	X   0     0   0	3					
	- -   - -   - -   - -   - -   - -   - -						
2	X   0   X     0   X   2	2					
	- -   - -   - -   - -   - -   - -   - -						
1	X   0   X   X   0   0   X   1	1					
	+ - - + - - + - - + - - + - - + - - + - - +						
	1	2	3	4	5	6	7

FIGURE 1 – Affichage de la grille

#### **Exercice 4 : Décider si un coup est possible**

Écrire un sous-programme qui indique s'il est possible de jouer dans une colonne donnée.

**Exemple :** Sur l'exemple de la figure 1, il est possible de jouer les colonnes 1, 2, 3, 5, 6 ou 7. Il n'est pas possible de jouer la colonne 4.

## Exercice 5 : Lâcher un jeton

Lorsqu'un joueur joue, il lâche un jeton de sa couleur au dessus d'une colonne de la grille. En raison de la force gravitationnelle, le jeton tombe dans la grille jusqu'à ce qu'il soit arrêté par un autre jeton ou le fond de la grille.

Écrire un sous-programme qui réalise cette opération.

**Exemple :** Partant de la figure 1, si les rouges jouent la colonne 5, le nouvel état de la grille est celui donné par la figure 2.

	1	2	3	4	5	6	7
6		*			6		
5		0			5		
4		0	0	*	4		
3		*	0	0	0	3	
2		*	0	*	*	0	*
1	*	0	*	*	0	0	*
	1	2	3	4	5	6	7

FIGURE 2 – Les rouges ont joué la colonne 5

**Exercice 6 : Compter les jetons alignés**

Écrire un sous-programme qui indique quelle est la longueur de l'alignement le plus long qui inclut un jeton donné (repéré par son numéro de colonne et son numéro de ligne). Les alignements doivent être cherchés dans toutes les directions (horizontale, verticale ou en diagonale).

Ce sous-programme sera ensuite utilisé pour déterminer la fin d'une partie (alignement  $\geq 4$ ) ou aider l'ordinateur à choisir la colonne où jouer (exercice 8).

Sur l'exemple de la figure 2, la case (1,1) correspond à un alignement de trois jetons en diagonale ; la case (7,1) correspond à un alignement de deux jetons verticalement ; la case (6,2) correspond à trois jetons alignés verticalement ou en diagonale...

**Exercice 7 : Conseiller une colonne où jouer**

Écrire un sous-programme qui étant donné une grille propose une colonne où jouer. Le numéro de la colonne sera choisi aléatoirement.

**Exercice 8 : Améliorer le conseil**

On conseillera la colonne qui permet l'alignement le plus long. Si un coup gagnant existe, la colonne conseillée conduira donc à la victoire.

**Exemple :** Partant de la figure 2, les jaunes joueront en colonne 5. Ceci provoque un alignement de 4 pions et donc la victoire.

**Exercice 9 : Jouer une partie**

Écrire un programme qui fait jouer un joueur humain contre l'ordinateur. Le joueur humain a les jetons jaunes et commence la partie.