

## **Projet Puissance 4**

Lucie Dubrunfaut – Aude Vieillefon

### Résumé :

Nous avons implémenté un jeu de puissance 4 qui comprend trois modes : 2 joueurs, 1 joueurs et 0 joueurs (démonstration). Le mode 2 joueurs peut se jouer avec l'extension "go", c'est à dire que si un pion est entouré par 4 pions adverses, il devient un des pions du camp adverse.

Le mode 1 joueur permet de jouer contre deux "intelligences artificielles", soit un mode "nobrain" qui place ses pions aléatoirement, soit un mode sournois qui gagne s'il le peut (quand trois de ces pions sont alignés) ou bloque l'adversaire si 3 de ces pions sont alignés, sinon il place ces pions aléatoirement de la même façon que le mode "nobrain". Pour cela, le mode sournois simule le tombé d'un pion (le sien puis celui de son adversaire) pour détecter une victoire au prochain coup.

Le mode à 0 joueur correspond à une démonstration, il est possible de choisir quelles "intelligences artificielles" s'affrontent.

### **Début du Main :**

Avant de commencer une partie le main utilise la fonction "initialiseVide" qui va parcourir le tableau de 7 Colonnes et de 6 Lignes pour insérer dans chaque case la valeur 0 qui sera reconnue comme une case vide.

Ensuite l'utilisateur peut choisir dans quel mode il veut jouer, ce mode doit exister et être 1, 2 ou 3.

### **Première partie : Joueur contre Joueur**

Il s'agit de la fonction "joueurContreJoueur" qui va commencer par demander à l'utilisateur s'il veut utiliser l'extension Go (le joueur doit répondre par 0 ou 1), ensuite elle va proposer au joueur de choisir qui commence la partie entre le Rond (1) et la Croix (2) (ici aussi le choix n'est que 1 ou 2). *Nous avons choisis ces symboles car ils étaient plus illustratifs que J et R sur la grille.*

La fonction fait appeler en premier lieu à une autre fonction "afficheClair" qui va permettre de montrer un aperçu graphique du tableau sur lequel on joue. Cette fonction parcourt le tableau et affiche chaque case en la traduisant selon le langage 0 = vide, 1 = Rond et 2 = Croix.

Ensuite le joueur qui commence doit donner la colonne dans laquelle il joue (sachant que visuellement elles sont numérotées de 1 à 7 tandis que pour la machine elle sont numérotées de 0 à 6), cette colonne ne doit pas être pleine (ceci est vérifié) et exister (pas de choix en dehors des colonnes définies).

Une fois qu'il a fait ce choix, la fonction "position" utilise ce choix ainsi que l'id du joueur en cours pour positionner le jeton dans la colonne souhaitée. Cette fonction se décompose en deux cas, une colonne qui n'est pas vide (ainsi le jeton s'arrête au dessus du dernier jeton) ou une colonne qui est vide (dans ce cas le jeton arrive en bas du tableau). Elle retourne la position dans la colonne du jeton ainsi déposé.

Suite à cela si l'extension Go a été choisie la fonction associée "extensionGo" qui utilise les coordonnées du jeton placé et l'id du joueur qui l'a placé, va vérifier si ce positionnement a induit un changement dans le tableau. Elle vérifie si le pion positionné est au dessus du pion à convertir ou s'il est à sa droite ou s'il est à sa gauche. Dans chaque cas elle va vérifier si cette modification a induit la victoire du joueur ou non.

Pour vérifier s'il y a eu victoire il faut faire appel à la fonction "plateauVictoire" qui prend les coordonnées du jeton transmis et l'id dont on souhaite vérifier la victoire. Elle fonctionne en utilisant des pions fantômes (bas qui ira aussi vers le haut, gauche et droite) qui vont se déplacer

dans toutes les directions à partir du jeton vérifié. Ainsi à chaque déplacement d'un pion fantôme qui aura répondu à l'égalité de l'id vérifié et de celle de la case traité, la valeur du pion fantôme augmente de 1. A La fin de chaque vérification de Colonne / Ligne / Diagonale ascendante / Diagonale descendante, si la valeur des pions est supérieure ou égale à 4 alors c'est une victoire pour l'id vérifié par la fonction.

Retour à la fonction joueur contre joueur après que l'extension Go ai vérifié que son changement (s'il y en a eu un) a pu induire une victoire ou non (si c'est le cas la fonction joueur contre joueur se termine elle aussi). Sinon la fonction vérifie si le coups du joueur en lui meme a pu induire une victoire en utilisant la meme fonction "plateauVictoire". Ensuite il faut vérifier si le tableau est plein par la fonction "plateauPlein" qui va parcourir le tableau du haut vers le bas à la recherche d'une case vide (en sachant que le bas du plateau ne peut pas contenir de case vide si les cases au dessus sont pleines) s'il n'en trouve pas il renvoie une valeur à la fonction joueur contre joueur qui lui indiquera qu'il est plein et alors celle-ci se termine également.

Ensuite si personne n'a gagné et que la plateau n'est pas plein, la fonction inverse les idjoueur pour permettre le tour du second joueur. La fonction recommence alors au niveau de l'affichage du tableau puis la demande de la colonne dans laquelle le joueur souhaite jouer.

## Seconde Partie : Joueur contre Intelligence Artificielle

Il s'agit de la fonction "joueurContreordi" qui va permettre de jouer contre une intelligence artificielle. En premier lieu elle demande donc à l'utilisateur de choisir le niveau de difficulté de son adversaire, soit "nobrain" qui est une l'Intelligence Artificielle 1, soit "facile mais sournois" qui va etre l'Intelligence Artificielle 2 (ici encore on vérifie que la difficulté choisie existe bien). Ensuite on demande au joueur de choisir son symbole (sachant que par défaut on définit que le joueur commencerait toujours la partie, et encore une fois son choix de symbole doit exister dans la liste proposé).

La partie peut commencer, le tour du joueur se déroule comme dans la précédente section c'est à dire sondage du choix de colonne, existence de la colonne, positionnement du jeton, existence d'une victoire induite, plateau plein.

Selon le niveau d'Intelligence Artificielle choisie la fonction exécutée est :

- Difficulté 1 : la fonction "intelligenceArtificielle1" qui va tirer au hasard un nombre entre 0 et 6 (puisque c'est la numérotation de tableau pour l'ordinateur). Bien sur elle vérifie que le choix fait ne soit pas dans une colonne pleine.

- Difficulté 2 : la fonction "Iasournois" qui va se décomposer en 3 possibilités selon leur priorité :

- Peut-elle réaliser un coups gagnant ?

Pour chaque colonne (si celle-ci est pleine elle passe directement à la suivante), elle va simuler une position de son jeton par la fonction "positionPotentielle" qui fonctionne comme la fonction de positionnement sauf qu'elle n'inscrit rien dans le tableau, elle se contente de renvoyer la position dans la colonne si on avait mit le jeton dans celle-ci. Ensuite l'IA sournois va tester si en jouant ce jeton la fonction victoire lui retournerait son id ou si elle lui retourne 0 (valeur de base pour dire que le coups n'est pas gagnant). Si elle lui retourne son id l'IA décide de jouer ce coups et renvoie donc le choix de colonne à la fonction joueur contre ordinateur, sinon elle passe à la colonne suivante.

- Sinon peut-elle empecher l'utilisateur de gagner ?

L'IA n'a trouvé dans le tableau aucune victoire pour elle mais peut-elle contrer une victoire imminente de son adversaire ? Par le meme procédé elle parcourt le tableau de colonne en colonne (si la colonne est pleine elle passe directement à la suivante) et elle test si son adversaire jouait dans cette colonne est-ce qu'il gagnerait ? Elle simule des positions dans chaque colonne puis les test avec la fonction victoire et l'id de son adversaire (au lieu de la sienne précédemment). Si elle trouve

une position potentielle de victoire elle renvoie son choix de colonne à la fonction joueur contre ordinateur, sinon elle passe à la colonne suivante.

- Si elle n'a pu ni trouve de victoire pour elle, ni pour son adversaire  
Elle joue aléatoirement comme l'IA 1 et renvoie donc un choix entre 0 et 6.

Comme pour les joueurs, en fin de tour des IA on vérifie s'il y a une victoire et sinon est-ce que le tableau est plein.

### **Troisième partie : Ordinateur contre Ordinateur**

Il s'agit de la fonction "OrdicontreOrdi" qui va simuler une partie entre deux IA. Elle commence par demander à l'utilisateur de choisir le niveau d'IA du premier joueur ensuite celui du second joueur (ici comme toujours on vérifie que le choix donné existe dans la liste).

Les tours se déroulent comme pour la partie ordinateur de la section précédente selon le niveau de l'IA.

Le premier joueur a pour symbole le rond et le second la croix, entre chaque tour il existe un timer qui va nous permettre de visualiser la partie (3 seconde de "pause" entre chaque joueur) sinon les deux ordi joueraient trop vite.

### **Retour au Main**

Selon la fonction choisie comme mode de jeu, elle va renvoyer un résultat qui sera interprété afin d'être affiché au joueur pour déterminer qui a obtenu la victoire.