

Saé S1.01 : implémentation d'un besoin client

Le jeu de Puissance4

Livrables attendus pour la partie 2 - Programmation

En annexe des spécifications précises vous sont fournies pour la programmation en langage C du jeu de Puissance4. Les livrables que vous devez fournir sont les suivants.

- Un **cahier de tests** où sont rassemblés pour chaque fonction ou procédure tous les cas de test à prévoir.
- Le **code source** commenté qui **devra respecter les spécifications fournies en annexe.**
- Une **documentation** Doxygen de votre code source, au format texte (fichier rtf).
- Une **trace d'exécution de chaque cas de test** prévu dans le cahier de tests. Une trace prendra la forme d'une copie-écran qui montre la réalisation du cas de test.

Il est conseillé de commencer à programmer le jeu de Puissance4 dès le début de cette période car c'est sans doute ce qui vous demandera la plus de temps.

ANNEXE : spécifications pour le programme Puissance4

Le programme doit permettre à deux joueurs humains A et B de faire une partie de Puissance4. La grille de jeu sera représentée par un tableau 2D de caractères. En effet, les pions seront représentés par le caractère 'X' pour le joueur A et par le caractère 'O' pour le joueur B. Les autres cases (les cases "vides") contiendront le caractère "espace". Ce sera toujours le joueur A qui commencera une partie.

Algorithme principal

```
programme Puissance4 c'est
    constante caractère PION_A := 'X';
    constante caractère PION_B := 'O';
    constante caractère VIDE := ' ';
    constante caractère INCONNU := ' ';
    constante entier NBLIG := 6;
    constante entier NBCOL := 7;
    constante entier COLONNE_DEBUT := NBCOL/2;

    type Grille : tableau[0..NBLIG-1][0..NBCOL-1] de caractère;

    procédure initGrille (entF/sortF Grille);
    procédure afficher (entF Grille, entF caractère, entF entier);
    fonction grillePleine (entF Grille) retourne booléen;
    procédure jouer(entF/sortF Grille, entF caractère, sortF entier,
entF/sortF entier);
    fonction choisirColonne (entF Grille, entF caractère, entF entier) retourne
entier;
    fonction trouverLigne (entF Grille, entF entier) retourne entier;
    fonction estVainqueur (entF Grille, entF entier, entF entier) retourne
booléen;
    procédure finDePartie (entF caractère);

Début
    vainqueur : caractère;
    ligne, colonne : entier;
    g : Grille;

    initGrille(entE/sortE g);
    vainqueur := INCONNU;
    afficher(entE g, entE PION_A, entE COLONNE_DEBUT);

    tant que vainqueur==INCONNU ET NON grillePleine(entE g) faire
    |   jouer(entE/sortE g, entE PION_A, sortE ligne, entE/sortE colonne);
    |   afficher(entE g, entE PION_B, entE COLONNE_DEBUT);
    |   si (estVainqueur(entE g, entE ligne, entE colonne)) alors
    |   |   vainqueur := PION_A;
    |   sinon si NON grillePleine(entE g) alors
    |   |   jouer(entE/sortE g, entE PION_B, sortE ligne, entE/sortE colonne);
    |   |   afficher(entE g, entE PION_A, entE COLONNE_DEBUT);
    |   |   si estVainqueur(entE g, entE ligne, entE colonne) alors
    |   |   |   vainqueur := PION_B;
    |   |   finsi
    |   finsi
    finfaire
    finDePartie(vainqueur);
fin
```

Liste des fonctions/procédures à programmer

procédure *initGrille*

Initialise la grille en affectant la constante VIDE à chacun de ses éléments.

paramètres

g (E/S) : Grille, représente la grille de jeu

procédure *afficher*

Réalise l’affichage à l’écran du contenu de la grille avec les pions déjà joués. Cette procédure affiche aussi, au-dessus de la grille, le prochain pion à tomber : il sera affiché au-dessus de la colonne dont le numéro est donné en paramètre. Cette procédure commencera par effacer l’écran.

Les correcteurs seront sensibles à la qualité de l’affichage et à la fidélité à votre maquette

paramètres

g (E) : Grille, représente la grille de jeu

pion (E) : caractère, représente le pion à afficher au-dessus de la grille

colonne (E) : entier, représente l’indice de la colonne au-dessus de laquelle le pion doit être affiché

fonction *grillePleine*

Teste si toutes les cases de la grille sont occupées ou non.

paramètres

g (E) : Grille, représente la grille de jeu

résultat

booléen : VRAI si toutes les cases de la grille sont occupées par les pions, FAUX sinon

procédure *jouer*

permet à un joueur de jouer son pion. La procédure fait appel à *choisirColonne*, afin que le joueur indique la colonne dans laquelle il veut jouer ; puis fait appel à *trouverLigne* pour définir la case où ajouter le pion.

paramètres

g (E/S) : Grille, représente la grille de jeu

pion (E) : caractère, correspond au pion à jouer

ligne (S) : entier, correspond à la ligne où est tombé le pion

colonne (S) : entier, correspond à la colonne où est tombé le pion

fonction *choisirColonne*

Un joueur voit son pion au-dessus de la grille et cette fonction doit lui permettre de "déplacer" son pion d'une colonne vers la gauche (par la touche 'q') ou d'une colonne vers la droite (par la touche 'd'). Après chaque déplacement, la grille est réaffichée. Le joueur peut finalement choisir la colonne où il souhaite faire tomber son pion (par la touche ESPACE).

paramètres

g (E) : Grille, représente la grille de jeu

pion (E) : caractère, représente le pion à tester

colonne (E) : colonne de départ (celle au-dessus de laquelle se trouve le pion initialement)

résultat

entier : indice de la colonne choisie par le joueur

fonction *trouverLigne*

Consiste à trouver la première case non occupée de la colonne. Si la colonne est pleine, la fonction retourne -1.

paramètres

g (E) : Grille, représente la grille de jeu

colonne (E) : entier, indice de la colonne dans laquelle le pion doit tomber

résultat

entier : indice de la ligne où le pion devra être ajouté ou -1 si la colonne est pleine

fonction *estVainqueur*

Indique si le pion situé dans la case repérée par les paramètres *ligne* et *colonne* a gagné la partie, c'est-à-dire s'il y a une ligne, une colonne ou une diagonale formée d'au moins 4 de ses pions (la ligne et la colonne passées en paramètres correspondent à la case où le joueur vient de jouer, c'est-à-dire la case à partir de laquelle il faut rechercher 4 pions successifs identiques).

paramètres

g (E) : Grille, représente la grille de jeu

ligne (E) et **colonne** (E) : entiers représentant la case à partir de laquelle rechercher une série de 4 successifs identiques

résultat

booléen : VRAI s'il y a une telle ligne de 4 à partir de la case indiquée, FAUX sinon

procédure *finDePartie*

Affiche le résultat d'une partie lorsqu'elle est terminée.

paramètres

pion (E) : caractère qui représente le pion gagnant (PION_A ou PION_B) ou bien VIDE si match nul