

Duncan Cadoret 1E2

SAE 1.01 Implémentation d'un besoin client

LE PUISSANCE 4

Sommaire :

Page 2 : Pseudo-code

Page 4 : rôle procédures et fonctions

Page 9 : Ordre d'exécution des fonctions

```

constante chaîne PION_R := "●" ;
constante chaîne PION_J := "○" ;
constante entier CASE_VIDE := 0 ; // valeur d'une case non remplie
constante entier JOUEUR1 := 1 ; // valeur d'une case remplie par joueur1
constante entier JOUEUR2 := 2 ; // valeur d'une case remplie par joueur2
constante entier BORNE_MIN := 1 ;
constante entier BORNE_MAX_HORIZ := 7 ; // nombre de cases à l'horizontale du
tableau
constante entier BORNE_MAX_VERTIC := 6 ; // nombre de cases à la verticale du
tableau
constante entier ALIGN_VICTOIRE := 4 ; // nombre de pion à aligner pour gagner
constante entier TOUR_NUL := 42 ; // nombre de jeton qu'il faut pour remplir
le tableau

type t_tab curseur : [BORNE_MIN .. BORNE_MAX_HORIZ] de entier; // tableau de
déplacement du curseur
type t_tab jeu : [BORNE_MIN .. BORNE_MAX_HORIZ, BORNE_MIN .. BORNE_MAX_VERTIC]
de entier; // tableau a 2 dimensions du jeu

procedure initialiserCurseur(sortF tab : t_tab curseur);
procedure initialiserTab(sortF tab : t_tab jeu);
procedure demandePionJ1(sortF pionJ1, sortF pionJ2);
procédure afficherCurseur(entF tab, entF curs, entF pion);
procédure afficherTableau(entF tab, entF pion1, entF pion2);
fonction instruction() delivre fonction; // fonction qui execute soit une
fonction soit une procedure
fonction verifHoriz(entF tab) delivre booléen ;
fonction verifVertic(entF tab) delivre booléen ;
fonction verifdiagGD(entF tab) delivre booléen;
fonction verifdiagDG(entF tab) delivre booléen;
procédure afficherGagnant(entF/sortF joueur1,entF/sortF joueur2);

programme puissance4 c'est
début
    pionJ1, pionJ2 : chaîne;
    tour : entier;
    gagne : booléen;
    matchNull : booléen;
    curseur : entier;
    tabCurseur : t_tab curseur;
    tabDeJeu : t_tab jeu;
    gagne := FAUX;
    matchNull := FAUX;
    tour := 0;
    curseur := 4;
    tabCurseur : t_tab curseur;

```

```

tabDeJeu : t_tab_jeu;

initialiserCurseur(sortE tabCurseur);
initialiserTab(sortE tabDeJeu);
demandePionJ1(sortE pionJ1, sortE pionJ2);
tant que (NON(gagne) ET NON(matchNull)) faire
    afficherCurseur(entE tabCurseur, entE curseur, entE pionJ1);
    afficherTableau(entE tabDeJeu, entE pionJ1, entE pionJ2);
    instruction();
    verifHoriz(entE tabDeJeu) ;
    verifVertic(entE tabDeJeu) ;
    verifdiagGD(entE tabDeJeu) ;
    verifdiagDG(entE tabDeJeu) ;
    tour := tour+1 ;
    egalite(entE tour) ;

    afficherCurseur(entE tabCurseur, entE curseur, entE pionJ2);
    afficherTableau(entE tabDeJeu, entE pionJ1, entE pionJ2);
    instruction();
    verifHoriz(entE tabDeJeu) ;
    verifVertic(entE tabDeJeu) ;
    verifdiagGD(entE tabDeJeu) ;
    verifdiagDG(entE tabDeJeu) ;
    tour := tour+1 ;
    egalite(entE tour) ;
finfaire
afficherTableau(entE tabDeJeu, entE pionJ1, entE pionJ2);
si (gagne) alors
    afficheGagnant(entE joueur1, entE joueur2);
sinon
    écrireEcran("Egalité");
finsi
fin

```

procedure initialiserCurseur

crée un tableau rempli de 0 l'element vide

paramètres

tab (sortie) : tableau, où se déplacera le curseur

résultat

tableau

procedure initialiserTab

crée un tableau rempli de 0 l'element vide

paramètres

tab (sortie) : tableau, où se déplacera seront positionné les pions

résultat

tableau

procedure demandePionJ1

demande au joueur 1 la couleur du pion qu'il souhaite, le deuxième joueur aura par défaut l'autre pion

paramètres

couleurPionJ1 (sortie) : chaine, couleur du pion du joueur n°1

couleurPionJ2 (sortie) : chaine, couleur du pion du joueur n°2

résultat

chaine, couleur des pions pour le rendu tableau

procedure afficherCurseur

Affiche le tableau de déplacement du curseur

Paramètres

Tab (entrée) : tableau du curseur

Curs (entrée) : indice de positionnement du curseur

Pion (entrée) : couleur du pion utilisé

Résultat

Chaine, mise en forme du tableau curseur

procedure afficherTableau

Affiche le tableau du puissance4

Paramètres

Tab (entrée) : tableau du curseur

Curs (entrée) : indice de positionnement du curseur

Pion1 (entrée) : couleur du pion utilisé

Pion2 (entrée) : couleur du pion utilisé

Résultat

Chaine, mise en forme du tableau curseur

fonction instruction

demande au joueur sur quelle touche il souhaite appuyer et selon cette touche cela exécutera une autre fonction

paramètres

aucun

résultat

fonction, exécute celle-ci qui correspondra à l'action demandée

procédure aDroite

modifie la position du curseur dans le tab curseur

paramètres

curseur (sortie), incrémente curseur de 1 pour le faire avancer d'une position

résultat

entier, correspondant à l'indice de positionnement du pion

fonction, on réexécute la fonction instruction

procédure aGauche

modifie la position du curseur dans le tab curseur

paramètres

curseur (sortie), décrémente curseur de 1 pour le faire reculer d'une position

résultat

entier, correspondant à l'indice de positionnement du pion

fonction, on réexécute la fonction instruction

fonction restart

relance le jeu puissance 4 à 0

paramètres

aucun

résultat

fonction, exécute le main() pour relancer le jeu

procédure help

affiche une fenêtre avec les différentes actions pouvant être faites

paramètres

aucun

résultat

chaîne, avec les différentes actions

fonction, on réexécute la fonction instruction

fonction chute

indique l'indice du point de chute du pion

paramètres

curseur (entrée, avoir la colonne du point de chute

tab entrée, pour savoir si un pion se trouve sur cette colonne,  
si oui on remonte d'une ligne, sinon le pion va tout en bas

joueur (entrée), savoir quel joueur joue

résultat

tableau, tabDeJeu, où l'on aura ajouter le chiffre du joueur là  
où il a joué.

fonction verifHoriz

indique si une condition de victoire est remplie à l'horizontale

paramètres

tab(entrée), savoir si 4 pions sont alignés à l'horizontale

résultat

booléen, renvoie Vrai

fonction verifVertic

indique si une condition de victoire est remplie à la verticale

paramètres

tab(entrée), savoir si 4 pions sont alignés à la verticale

résultat

booléen, renvoie Vrai

fonction verifdiagGD

indique si une condition de victoire est remplie sur une des diagonales

paramètres

tab(entrée), savoir si 4 pions sont alignés sur une des diagonales

résultat

booléen, renvoie Vrai

fonction verifdiagDG

indique si une condition de victoire est remplie sur une des diagonales

paramètres

tab(entrée), savoir si 4 pions sont alignés sur une des diagonales

résultat

booléen, renvoie Vrai

fonction egalite

indique si le match est nul

paramètres

tour (entrée) (négligeable), si tour atteint 42 alors il y a (si pas victoire) égalité

résultat

booléen, renvoie Vrai



## Ordre d'exécution des fonctions

- ➔ initialiserCurseur
- ➔ initialiserTab
- ➔ demandePionJ1
- ➔ afficherCurseur
- ➔ afficherTableau
- ➔ instruction
  - o aDroite
    - instruction
  - o aGauche
    - instruction
  - o restart
    - instruction
  - o help
    - instruction
  - o chute
    - afficherTableau
    - verifHoriz
    - verifVertic
    - verifDiagGD
    - verifDiagDG
    - egalite
- ➔ restart