

## PSEUDO-CODE PUISSANCE 4

programme jeu c'est

// déclaration des constantes

type t\_grille = tableau[0..5, 0..6] de entier;

// entêtes des fonctions et procédures

procedure prenoms(sortF prenom1 : chaine(30), sortF prenom2 : chaine(30));

procedure regles();

procedure couleurs(entF couleur1 : chaine(10), entF couleur2 : chaine(10));

procédure qui\_commence(entF/sortF prenom1 : chaine(30), entF/sortF prenom2 : chaine(30));

procédure aff\_grille(entF grille : t\_grille, entF couleur1 : chaine(10), entF couleur2 : chaine(10));

procédure remplir\_grille(entF joueur : entier, entF/sortF grille : t\_grille, sortF colonne : entier, sortF ligne : entier);

procédure coup\_gagnant(entF joueur : entier, entF colonne : entier, entF ligne : entier, sortF gagnant : booléen);

fonction grille\_pleine(entF grille : t\_grille) délivre booléen;

début // code du programme principal : appel des fonctions et procédures

    prenom1, prenom2 : chaine(30);

    rep : chaine(4);

    couleur1, couleur2 : chaine(10);

    colonne : entier;

    ligne : entier;

    gagnant : booléen;

    rejouer : chaine(4);

    grille : t\_grille;

    grille := ((0, 0, 0, 0, 0, 0, 0, 0),

            (0, 0, 0, 0, 0, 0, 0, 0),

            (0, 0, 0, 0, 0, 0, 0, 0),

            (0, 0, 0, 0, 0, 0, 0, 0),

            (0, 0, 0, 0, 0, 0, 0, 0),

            (0, 0, 0, 0, 0, 0, 0, 0));

    ecrireEcran("BIENVENUE");

```

prenoms(sortE prenom1, sortE prenom2);

ecrireEcran("Voulez-vous jouer ? (OUI / NON)");
lireClavier(rep);

tant que ((rep != "OUI") OU (rep != "NON")) faire
    ecrireEcran("erreur de saisie");
    ecrireEcran("Voulez-vous jouer ? (OUI / NON)");
    lireClavier(rep);
finfaire

si (rep == "OUI") alors
    regles();

    rejouer := "OUI";

    tant que (rejouer == "OUI") faire
        qui_commence(entE/sortE prenom1, entE/sortE prenom2);
        couleurs(entE prenom1, entE prenom2, sortE couleur1, sortE couleur2);
        ecrireEcran("C'EST PARTI !");

        tant que ((grille_pleine(entE grille) != vrai) OU (gagnant != vrai)) faire
            aff_grille(entE grille, entE couleur1, entE couleur2);
            remplir_grille(entE prenom1, entE/sortE grille, sortE colonne, sortE ligne);
            coup_gagnant(entE prenom1, entE colonne, entE ligne, sortE gagnant);
            si (gagnant == vrai) alors
                ecrireEcran(prenom1, "a gagné !");
                ecrireEcran("La partie est finie !");
            sinon
                aff_grille(entE grille, entE couleur1, entE couleur2);
                remplir_grille(entE prenom2, entE/sortE grille, sortE colonne, sortE ligne);
                coup_gagnant(entE prenom2, entE colonne, entE ligne, sortE gagnant);
                si (gagnant == vrai) alors
                    ecrireEcran(prenom2, "a gagné !");
                    ecrireEcran("La partie est finie !");
                finsi
            finsi
        finfaire

        ecrireEcran("Voulez vous rejouer ? (OUI / NON) ");
        lireClavier(rejouer);

    finfaire
fin

ecrireEcran("A BIENTOT !");

fin

```

**procédure** *prenoms*

demande à l'utilisateur de saisir un premier prénom, puis celui de son adversaire

**paramètres :**

prenom1 (sortie) : chaîne de caractères (30 max), le premier prénom demandé

prenom2 (sortie) : chaîne de caractères (30 max), le second prénom demandé

**résultat :**

Les prénoms sont stockés respectivement dans les variables prenom1 et prenom2

**procédure** *regles*

demande aux utilisateurs s'ils veulent connaître les règles du puissance 4

**paramètres :**

aucun

**résultat :**

si l'utilisateur répond 'OUI' alors les règles sont affichées

**procédure** *couleurs*

propose aux utilisateurs de choisir une couleur entre jaune et rouge

**paramètres :**

couleur1 (entrée) : chaîne de caractères (max 10), une des 2 couleurs proposés

couleur2 (entrée) : chaîne de caractères (max 10), une des 2 couleurs proposés

**résultat :**

associe une couleur à un joueur

**procédure** *qui\_commence*

demande aux utilisateurs : "qui veut commencer ?"

**paramètres :**

prenom1 (entrée/sortie) : chaîne de caractères (30 max)

prenom2 (entrée/sortie) : chaîne de caractères (30 max)

**résultat :**

le prénom du joueur qui souhaite commencer sera stocké dans la variable prenom1 et l'autre dans prenom2

**procédure** *aff\_grille*

affiche la grille de jeu, ainsi que le message demandant au joueur concerné de choisir une colonne où déposer sa pièce

**paramètres :**

grille (entrée) : tableau correspondant à la grille de jeu

couleur1 (entrée) : chaîne de caractères (max 10), une des 2 couleurs

couleur2 (entrée) : chaîne de caractères (max 10), une des 2 couleurs

**résultat :**

affiche la grille de puissance 4 dans l'état actuel de la partie (vide si c'est au début, puis remplie au fur et à mesure de la partie)

**procédure** *remplir\_grille*

permet de remplir la grille à chaque coup joué

**paramètres :**

joueur (entrée) : entier (1 ou 2), permet de remplir le tableau selon le joueur et ses coups joués

grille (entrée/sortie) : tableau correspondant à la grille de jeu

colonne (sortie) : entier, renvoie la colonne dans laquelle le joueur a déposé sa pièce

ligne (sortie) : entier, renvoie la ligne dans laquelle le joueur a déposé sa pièce

**résultat :**

remplit le tableau 'grille' selon les coups joués

**procédure** *coup\_gagnant*

vérifie si le coup joué est un coup gagnant

**paramètres :**

joueur (entrée) : entier (1 ou 2), selon s'il s'agit du joueur 1 ou 2

colonne (entrée) : entier, la colonne dans laquelle le joueur a déposé sa pièce

ligne (entrée) : entier, la ligne dans laquelle le joueur a déposé sa pièce

gagnant (sortie) : booléen, vrai si le coup est gagnant, faux sinon

**résultat :**

renvoie vrai ou faux

**fonction** *grille\_pleine*

vérifie si la grille est pleine selon l'avancement de la partie

**paramètres :**

grille (entrée) : tableau correspondant à la grille du jeu dans l'état actuel de la partie

**résultat :**

retourne un booléen, vrai si la grille est pleine, faux sinon