

Liste des fonctions et procédures

procedure saisiNom

Procédure permettant aux joueurs de saisir leurs noms

paramètres

-joueur_1(Sortie) : chaine, assignation du nom du joueur_1 par le premier utilisateur

-joueur_2(Sortie) : chaine, assignation du nom du joueur_2 par le deuxième utilisateur

procedure placerSonPion

Procédure permettant de placer un pion dans la grille

paramètres

-tab(Entrée/Sortie) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, tableau permettant d'être modifié si le joueur place un pion

-joueurCommence(Entrée) : chaine, permet de savoir qui va mettre le pion dans le jeu

-joueur_1(Entrée) : chaine, nom du joueur_1 qui permet de voir si ce dernier est égale au paramètre 'joueurCommence'

-joueur_2(Entrée) : chaine, nom du joueur_2 qui permet de voir si ce dernier est égale au paramètre 'joueurCommence'

procedure recommencerPartie

Procédure pour que les joueurs puissent recommencer la partie

paramètres

-rejouer(Entrée/Sortie) : chaine, paramètre que le joueur doit remplir soit par une chaine vide ' ' si il veut continuer la partie ou le joueur doit noter 'FIN' si il veut arrêter

procedure grille

Procédure qui affiche la grille de jeu dans la console

paramètres

-tab_joueurs(Entrée) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, utilisation de ce tableau pour afficher dans la console la grille avec les pions

procedure *creationTab*

Procédure permettant d'initialiser le tableau de jeu

paramètres

-tab_joueurs(Sortie) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, initialisation du tableau qui sera donc composé de 0

fonction *commenceAJouer*

Fonction qui permet de savoir lequel des deux joueurs commencera la partie

paramètres

-joueur_1(Entrée) : chaîne, paramètre de vérification pour voir si l'utilisateur a choisi joueur_1 pour commencer la partie

-joueur_2(Entrée) : chaîne, paramètre de vérification pour voir si l'utilisateur a choisi joueur_2 pour commencer la partie

résultat

chaîne, renvoie un des 2 paramètres d'entrée(joueur_1 ou joueur_2)

fonction *donneeCorrecte*

Fonction qui vérifie si le joueur a saisi une donnée correcte entre le minimum de largeur et son maximum et vérifie aussi si la colonne n'est pas pleine lorsque le joueur place son pion

paramètres

-choixJoueur(Entrée) : entier, choix de la colonne où le joueur place son pion

-tab_joueurs(Entrée) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, tableau qui permet de vérifier si la colonne sélectionnée n'est pas remplie totalement

résultat

booléen, renvoie VRAI si le placement de pion de l'utilisateur est correcte, sinon cela renvoie FAUX

fonction *gagner*

Fonction qui renvoie VRAI et affiche un des deux joueurs qui a gagné et le perdant

paramètres

-tab_joueurs(Entrée) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, tableau qui permet de voir si il y a 4 pions qui sont alignés

-joueur_1(Entrée) : chaîne, nom du joueur_1 qui permet de vérifier si il a gagné

-joueur_2(Entrée) : chaîne, nom du joueur_2 qui permet de vérifier si il a gagné

résultat

booléen, si il y a une puissance 4 chez un des deux joueurs, cela renvoie VRAI et cela

affiche un message dans la console, sinon cela renvoie FAUX

fonction *egalite*

Fonction qui vérifie si il n'y a pas d'égalité dans la partie

paramètres

-tab_joueurs(Entrée) : tableau[1.. HAUTEUR, 1.. LARGEUR] d'entier, tableau permettant de voir si ce dernier n'est pas rempli totalement

-joueur_1(Entrée) : chaine, nom du joueur_1 qui permet d'être affiché lorsqu'il y a une égalité

-joueur_2(Entrée) : chaine, nom du joueur_2 qui permet d'être affiché lorsqu'il y a une égalité

résultat

booléen, renvoie VRAI quand le tableau est rempli totalement et FAUX sinon