Algorithme Pendu

**Fonction** principale()

//**Var** banque[][] **en caractère**

**Var** paysPourTableau[50][250], capitalePourTableau[50][250] **en caractère**

**Var** indexPays, indexCapitale, continent, compteur, tailleFichier <- 50 **en entier**

**Var** pays[250], capitaleADeviner[250], capitale[250] **en caractère**

**Ouvrir** fichier « fichierDePays.txt » **en** **lecture**

**VérifierSi** fichierDePays.txt **existe**

**Pour** compteur **allant de** 0 **à** tailleFichier

paysPourTableau[n°compteur] <- premier mot de la ligne n°compteur de

« fichierDePays »

capitalePourTableau[n°compteur] <- deuxième mot de la ligne n°compteur de « fichierDePays »

**Fin Pour**

**Faire**

continent <- menu()

**Selon** continent

**Cas** 1

indexPays <- genererNombreAleatoire(0 et 9)

pays <- paysPourTableau[indexPays]

capitaleADeviner <- capitalePourTableau[indexPays]

**Pour** compteur **allant** **de** 0 **à** 50

capitale[compteur] <- capitalPourTableau[compteur]

**Fin Pour**

**Cas** 2

**Cas** 3

**Cas** 4

**Cas** 5

**TantQue** continent <> 6

**Enlever** capitaleADeviner **de** capitale

devinerCapitale(pays, capitaleADeviner, capitale)

**Fin Fonction**

**Fonction** menu()

**Var** continent **en entier**

**Faire**

**Mettre curseur à** 33, 2 **Ecrire** « JEU DU PENDU »

**Mettre curseur à** 32, 3 **Ecrire** « MENU PRINCIPAL »

**Mettre curseur à** 14, 8 **Ecrire** « 1. Africa »

**Mettre curseur à** 14, 9 **Ecrire** « 2. America »

**Mettre curseur à** 14, 10 **Ecrire** « 3. Asia »

**Mettre curseur à** 14, 11 **Ecrire** « 4. Europe »

**Mettre curseur à** 14, 12 **Ecrire** « 5. Oceania »

**Mettre curseur à** 14, 13 **Ecrire** « 6. Quitter »

**Mettre curseur à** 14, 15 **Ecrire** « Veuillez choisir le continent : »

**lire** continent

**TantQue** continent **<>** 1 **ou** 2 **ou** 3 **ou** 4 **ou** 5 **ou ­**6

**FinFonction**

**Fonction** genererNombreAleatoire(**var** minimum **en entier**, **var**  maximum  **en entier**)

**Var** nbrAleatoire **en entier**

nbrAleatoire <- Générer un nombre aléatoire entre minimum et maximum

//((**Générer un nombre)** modulo(maximum – minimum + 1)) + minimum

**Retourne** nbrAleatoire

**FinFonction**

**Fonction** devinerCapitale(pays, capitaleADeviner, capitale[])

potence(0)

**Var** compteur **en entier**

**Var** indice = 0 **en entier**

**Var** listeCapitale[] **en caractère**

**Var** tableauAleatoire[] **en entier**

tableauAleatoire[0] <- genererNombreAleatoire(0 et 49)

**Pour** compteur **allant de** 1 **à** 8

**TantQue** test = True

**Pour** indice **allant de** 0 **à** compteur

tableauAleatoire[compteur] <- genererNombreAleatoire(0 et 49)

**Si** tableauAleatoire[indice] = tableauAleatoire[indice-1]

Test <- True

**Sinon**

Test <- False

**FinTantQue**

listeCapitale [compteur] <- capitale[tableauAleatoire[compteur]]

**Enlever** listeCapitale [compteur] **de** capitale

**Fin Pour**

**Var** insertionAleatoire <- genererNombreAleatoire(0 et 9) **en entier**

listeCapitale [9] <- listeCapitale [insertionAleatoire]

listeCapitale [insertionAleatoire] <- capitaleADeviner

**Pour** compteur **allant de** 0 **à** 10

**Mettre curseur où il faut Afficher** listeCapitale[compteur]

**Fin Pour**

potence(0)

**Var** nombreEssai = 0 **en entier**

**Tant que** nombreEssai< 2 **faire**

**Var** choixUtilisateur **en entier**

choixUtilisateur <- **Valeur Saisie**

**Si** listeCapitale[choixUtilisateur] = capitaleADeviner

**Afficher** messageDeVictoire

**Fin Tant que**

**Fin Si**

nombreEssai <- nombreEssai + 1

potence(nombreEssai)

**Fin Fonction**

**Fonction** basePotence()

**Afficher** JEU DU PENDU

**Mettre curseur à** 26, 4 **Afficher** \*\*\*\*\*\*\*\*\*\*\*\*\*

**Pour** conteur <- 12

**Mettre curseur à** 26, 5+conteur **Afficher** \*

**Mettre curseur à** 22, 18 **Afficher** \*\*\*\*\*\*\*\*\*

**Fonction** potence(**var** etatPotence **en entier, var** listeCapital[] **en caractère**)

**Effacer** console

**Selon** etatPotence

**Cas** 0

basePotence()

**Cas** 1

basePotence()

**Mettre curseur à** 36, 5 **Afficher** \*\*\*\*\*

**Mettre curseur à** 36, 6 **Afficher** \* \*

**Mettre curseur à** 36, 7 **Afficher** \*\*\*\*\*

**Mettre curseur à** 38, 8 **Afficher** \*

**Mettre curseur à** 35, 9 **Afficher** \*\*\*\*\*\*\*

**Mettre curseur à** 36, 10 **Afficher** \* \*

**Mettre curseur à** 36, 11 **Afficher** \* \*

**Cas** 2

basePotence()

**Mettre curseur à** 36, 5 **Afficher** \*\*\*\*\*

**Mettre curseur à** 36, 6 **Afficher** \* \*

**Mettre curseur à** 36, 7 **Afficher** \*\*\*\*\*

**Mettre curseur à** 38, 8 **Afficher** \*

**Mettre curseur à** 35, 9 **Afficher** \*\*\*\*\*\*\*

**Mettre curseur à** 36, 10 **Afficher** \* \*

**Mettre curseur à** 36, 11 **Afficher** \* \*

**Afficher** messageDeDefaite

**Ecrire** Pour continuer à jouer, appuyez sur une touche

**Si une touche est appuyée**

Menu()

**Fin Si**

**FinFonction**

void gotoxy(int *colonne*, int *ligne*){  
 *// Initialise les coordonnées.* COORD *coord* = {*colonne*, *ligne*};  
  
 *// Place le curseur à l'endroit désiré à l'écran.* SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), *coord*);  
}