

Le Plus ou Moins est un peu simple, vous ne trouvez pas ? Montons la difficulté d'un cran...

Quelques astuces avant de commencer

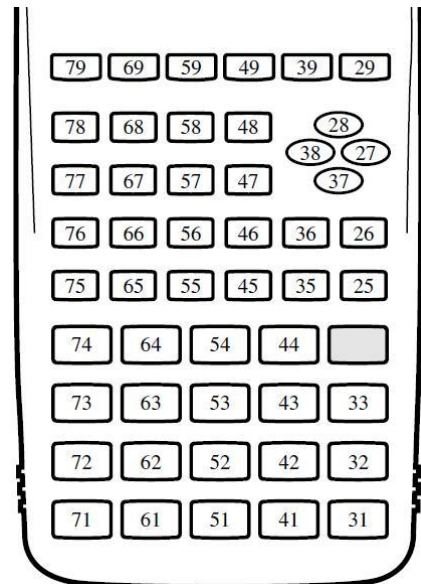
Le Getkey

Pour l'instant, nous ne connaissons qu'un seul moyen pour communiquer avec l'utilisateur : la fonction « ? ». Cependant, elle n'est ni très pratique, ni très esthétique car son appel décale le texte vers le haut. Heureusement, il est possible de détecter quelle touche est appuyée via Getkey : [SHIFT] + [VAR5] (PRGM) puis [F6] + [F4] (I/O) + [F2] (Gtkey).

```
Do
Getkey→B
LpWhile B=0
```

Si aucune touche n'est appuyée, Getkey renvoie 0. L'image ci-contre donne les valeurs de touches. Remarquez que [AC/ON] n'a pas de valeur car elle permet d'interrompre le programme. Par exemple, si j'appuie sur [0], Getkey renvoie 71 et 48 pour [MENU]. Je vous déconseille vivement de mettre le Getkey toute seule dans la boucle, vous gagnerez en réactivité. Je vois trop souvent ce genre de code dans les programmes :

```
Do
Getkey→B
If B=71
Then...
IfEnd
If B=72
Then...
IfEnd
(plusieurs lignes plus loin)
LpWhile B≠47
```



Source : Manuel d'utilisateur Calculatrices Casio

Isolez au maximum vos Getkey de tous calculs inutiles. Certes, vous perdez quelques octets, mais les gains en vitesse d'exécution du programme sont importants !

```
Do
Do
Getkey→B
LpWhile B=0
(votre code)
LpWhile B≠47
```

Allégez vos conditions

```
=====MORPION =====  
IfEnd#  
J→H#  
Isz J#  
J=3⇒1→J#  
IfEnd#  
0→G#  
Lbl Goto ⇒ Isz Dsz Menu
```

Dans l'onglet (PRGM) puis (JUMP) [F3], vous avez ce qu'on appelle la « double flèche ». Elle agit comme un If mais ne peut avoir qu'une seule action.

► Les Graph 100 (+ et USB) n'ont pas cette « double flèche » dans leur menu, même si elles savent l'interpréter. On peut la récupérer d'un programme et

avec le copier-coller [CTRL]+[F1] (Copy) puis [CTRL]+[F2] (Paste).

Les deux codes suivants ont le même effet, mais l'un est plus léger en mémoire.

```
If A=1  
Then 3→B  
IfEnd  
  
A=1=>3→B
```

Isz et Dsz : incrémentez ou décrémentez facilement

Une dernière petite astuce : les fonctions Isz et Dsz. La première incrémente de 1 la variable, l'autre décrémente de 1. Ces fonctions sont disponibles dans l'onglet (PRGM) puis [F3] (JUMP).

```
1→B  
While B<100  
Isz B  
Locate 1,1,B  
WhileEnd
```

Mais comme rien n'est simple, Isz et Dsz ont une fonction cachée, si la variable vaut 0 après l'incrémementation ou la décrémentation, la ligne suivante est ignorée. Faites bien attention à cette particularité, c'est une source récurrente d'erreurs !

```
1→B  
Dsz B  
"BONJOUR"  
"AU REVOIR"
```

« BONJOUR » ne s'affichera pas à l'écran. D'ailleurs, vous comprenez maintenant pourquoi elles se trouvent dans l'onglet (JUMP) car elles permettent d'effectuer un saut selon une condition bien précise.

Le morpion !

Vous connaissez tous le jeu du morpion où deux joueurs s'affrontent sur une grille de 3*3 cases et doivent aligner 3 croix ou 3 ronds. Le système de vérification des cases sera la partie la plus difficile. Lorsque vous aurez terminé, continuez sur la page suivante pour une proposition de correction.

```

/* MENU DU DEBUT */
"LE JOUEUR 1 CHOISIT ?"
" 1-Les ronds (O)"
" 2-Les croix (X)"
Do
Getkey
LpWhile Ans≠72 And Ans≠62 //Tant que [1] ou [2] n'est pas appuyée, le programme tourne en
boucle. Si une autre touche est appuyée, la boucle continue car la condition est vraie.
/* INITIALISATION DE LA PARTIE */
If Ans=72 //On stocke qui prend les ronds et les croix
Then {1,2}→List 1
Else {2,1}→List 1
IfEnd
0→Z //Initialisation du compteur de cases prises
1→J
{3,3}→Dim Mat A //Initialisation de la grille du morpion
/* PREPARATION DE L'ECRAN DE JEU */
ClrText
"===== MORPION ====="
Locate 9,3,"7 8 9"
Locate 9,4,"4 5 6"
Locate 9,3,"1 2 3"
Locate 1,7,"Joueur (3 espaces) ( )"
Do //1
Locate 8,7,J //On affiche à qui est le tour
List 1[J]→I
If I=1
Then Locate 11,7,"O"
Else Locate 11,7,"X"
IfEnd
/* LA PARTIE */
Do //2
Do //3
Getkey→B
LpWhile B=0 //Fin 3. Le programme tourne en boucle tant qu'une touche n'est appuyée
0 //Mettre un 0 tout seul attribue la valeur 0 à Ans qui est une variable temporaire
B=72=>1:B=62=>2 :B=52=>3 //Les deux points « : » fonctionnent comme des retours à la ligne,
B=73=>4 :B=63=>5 :B=53=>6 //ils permettent de condenser le code. (PRGM) + [F6] + [F5]
B=74=>7 :B=64=>8 :B=54=>9
LpWhile Ans=0 //Fin 2.Si Ans = 0, cela veut dire que la touche appuyée n'est pas valide. Supprimez
cette vérification et votre programme s'arrêtera à cause d'une erreur de calcul.

----- LE CODE DU PROGRAMME CONTINUE SUR LA PAGE SUIVANTE -----

```

```

Ans→B
1+Int((B-1)/3)→E
1+3Frac((B-1)*3)→F
If Mat A[E,F]=0 //On vérifie que la case soit vide (=0)
Then I→Mat A[E,F]
Isz Z //On incrémente le compteur de cases prises de 1
If I=1 //On affiche le symbole correspondant au joueur dans la case
Then Locate 7+2F,6-E,"O"
Else Locate 7+2F,6-E,"X"
IfEnd
J→H
Isz J //On incrémente de 1 le numéro du joueur (1→2→3)
J=3⇒1→J //Si le numéro du joueur vaut 3, alors c'est au tour du joueur 1 (3→1)
IfEnd
/* VERIFICATION SI UN JOUEUR A GAGNE */
0→G //Si G=0, aucun joueur a gagné, le cas contraire pour G=1
For 1→B To 3 //Lignes horizontales
Mat A[B,1]⇒Mat A[B,1]=Mat A[B,2] And Mat A[B,1]=Mat A[B,3]⇒1→G
Next
For 1→B To 3 //Lignes verticales
Mat A[1,B]⇒Mat A[1,B]=Mat A[2,B] And Mat A[1,B]=Mat A[3,B]⇒1→G
Next
Mat A[2,2]⇒Mat A[1,1]=Mat A[2,2] And Mat A[1,1]=Mat A[3,3] Or Mat A[1,3]=Mat A[2,2]
And Mat A[1,3]=Mat A[3,1]⇒1→G //Les deux diagonales
LpWhile G=0 And Z<9 //Fin 1. Si un joueur a gagné (G=1) ou si toutes les cases sont prises (Z=9),
alors la partie est terminée
/* MESSAGE DE FIN DE PARTIE */
If G=1
Then Locate 1,7,"Le joueur (3 espaces) a gagné !"
Locate 11,7,H
Else Locate 1,7,"(5 espaces) Match nul !"
IfEnd

```

A part le système de vérification des lignes, si vous avez bien suivi le cours, vous ne devriez pas avoir rencontré de grosses difficultés. La semaine prochaine, nous aborderons les chaînes de caractères appelées *String* en anglais grâce auxquelles vous pourrez manipuler les mots et d'autres utilisations plus puissantes.