

Les fonctions usuelles

Bonjour à tous ! Aujourd'hui, nous nous retrouvons pour vous apprendre à utiliser les fonctions usuelles dont vous vous servirez les 99,9% du temps dans vos jeux. Ces fonctions permettent de savoir sur quelle touche l'utilisateur appuie, de choisir précisément l'endroit où vous voulez afficher un texte ou une variable, et beaucoup d'autres choses !

Envie d'en savoir plus ?

Introduction :

Les fonctions que nous allons voir ici se trouvent toutes dans le menu de commandes de votre calculatrice, avec les instructions telles que « ? », « If » ou encore « ▲ ». De plus, elles sont toutes compatibles avec les anciennes générations de Graph, telles que les Graph 35, 35+, 65, et bien entendu les plus récentes : Graph 35+USB, 75, 85, 95 (SD). Vous pouvez donc les utiliser sans problèmes dans tous vos programmes.



Attention !

Ces fonctions n'ont toutefois pas la même rapidité en fonction de l'âge de la calculatrice !

Un résumé des calculatrices compatibles :

Liste des calculatrices compatibles :

- Graph 35
- Graph 35+
- Graph 65
- Graph 35+ USB
- Graph 75
- Graph 85 (avec ou sans mise à jour)
- Graph 95 (SD)
- Fx-CG 10/20 (Prizm)

Nous essayerons, tout au long de ce tutoriel, de vous apprendre à bien connaître les petites astuces qui rendent votre code moins lourd, et plus lisible. N'hésitez donc pas à revoir quelques passages que vous n'auriez que peu ou pas du tout compris : votre maîtrise du Basic Casio n'en sera que meilleure. De plus, ayez de préférence de quoi tester les codes proposés. Votre Graph favorite ou un émulateur suffira.

Mais tout de suite, passons aux choses plus sérieuses.

Une première fonction, « Locate » :

Pour écrire du texte, vous avez appris précédemment comment utiliser les « " ». Mais cette méthode n'est pas très pratique lorsqu'il s'agit d'afficher une phrase au milieu de l'écran. En effet, pour obtenir le résultat suivant :

```
HELLO WORLD
```

Nous pouvons être tentés d'écrire :

```
" "  
" "  
" "  
" HELLO WORLD"
```

Or, bien que peu pratique, cette syntaxe est longue, lourde à écrire, et bien peu compréhensible. En plus, imaginez que nous ayons envie de faire monter et descendre le texte : ce ne sera pas du tout pratique...

C'est là qu'intervient le Locate. Il est généralement considéré comme la fonction la plus utilisée en Basic Casio. Pour y accéder et l'insérer dans vos futurs programmes, tapez [SHIFT] / [VARS] (PRGM) / [F6] / [F4] (I/O) / [F1] (Lcte).

Cette fonction attend trois arguments : le premier est la position du texte sur l'axe des X (il y a 21 colonnes disponibles, donc un nombre entier de 1 à 21 compris), le second la position sur les Y (7 lignes dont de 1 à 7), et le texte, entre guillemets.

Voici ce que donne l'exemple de tout à l'heure avec un Locate :

```
Locate 6, 4, "HELLO WORLD"
```

Et ça marche ! En une ligne, nous obtenons ce qui nous prenait quatre lignes.

En prime, voici quelques astuces pour mieux utiliser cette fonction :

- Vous pouvez afficher le contenu d'une variable, en remplaçant le texte entre guillemets par le nom de la variable.
- Il est possible d'utiliser des variables déterminer la position du texte
- Toutefois, il ne revient pas automatiquement à la ligne si la phrase est trop longue...

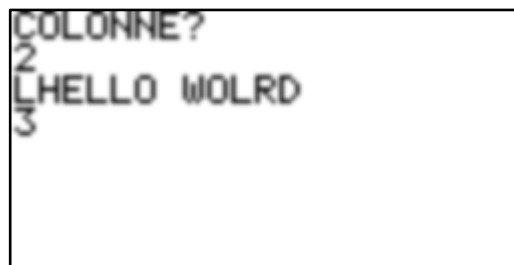
Un exemple d'utilisation, afficher un texte là où l'utilisateur le demande :

```
"Colonne" ? → X  
"Ligne" ? → Y  
Locate X,Y, "HELLO WORLD"
```

Si vous avez essayé, et je vous conseille fort de le faire, vous vous apercevrez de plusieurs « bugs ».

NB : Dans le langage informatique, un bug est une erreur qui n'interromps pas le programme, mais qui peut gêner la présentation ou la jouabilité d'un jeu. Une erreur qui arrête l'exécution du programme avant la fin est une erreur de syntaxe. Ces deux choses sont totalement différentes, bien qu'une erreur de syntaxe peut être avoir comme origine un argument qui est modifié dans le programme, comme une variable décimale, par exemple. Ne vous inquiétez pas, nous y reviendrons.

Le premier bug, le plus visible, est que vous devriez obtenir, si vous rentrez comme nombres 2 et 3, l'écran suivant :



```
COLONNE?  
2 HELLO WOLRD  
3
```

Le texte est superposé à celui qui était avant. Pour corriger ceci, nous allons utiliser une deuxième fonction, « ClrText ». Elle efface l'intégralité de l'écran. Vous la trouverez ici : [SHIFT] / [VARS] (PRGM) / [F6] / [F1] (CLR) / [F1] (Text).

Ainsi arrangé, le code ressemble à ceci :

```
"Colonne" ? → X  
"Ligne" ? → Y  
ClrText // on efface tout  
Locate X,Y, "HELLO WORLD"
```

Le deuxième bug, un peu plus masqué, est que si vous ne rentrez pas un nombre entier compris entre 1 et 21 pour les colonnes et entre 1 et 7 pour les lignes, la calculatrice vous affichera une erreur de syntaxe : « Erreur Argument. Appuyez sur [EXIT] ». Pour palier à cela, il suffit d'ajouter une boucle « Faire – Tant Que » (Do – LpWhile) qui ne passe à l'affichage que si les nombres entrés sont corrects. Pour obtenir la partie décimale, nous utiliserons « Frac * » ([OPTN] / [F6] / [F4] (NUM) / [F3] (Frac)). Aussi, pour récupérer la partie entière, c'est la fonction juste à côté ([F2] (Int)). Une fois de plus, voici à quoi ressemblera notre code :

```

Do
"Colonne" ? → X
"Ligne" ? → Y
LpWhile X<1 And X>21 And Y<1 And Y>7 And Frac X≠0 And Frac Y≠0
ClrText // on efface tout
Locate X, Y, "HELLO WORLD"

```

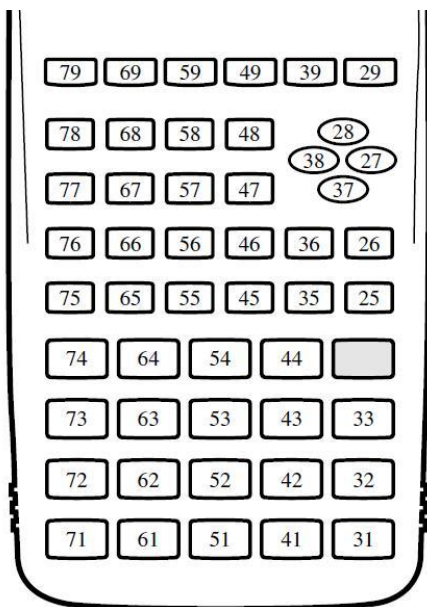
Il n'y a plus aucun souci, le programme tourne jusqu'à ce qu'on lui donne deux bons nombres.

Une deuxième grande fonction, « Getkey » :

Le Getkey est sans doute la deuxième fonction la plus utilisée dans les jeux puisque c'est grâce à elle que nous allons connaître la touche qui est enfoncée par l'utilisateur.

Tout d'abord, vous trouverez le Getkey à côté du Locate, c'est à dire via [SHIFT] / [VARS] (PRGM) / [F6] / [F4] (I/O) / [F2] (Gtkey).

Il s'utilise un peu comme le « ? » : il faut souvent lui ajouter une flèche et une variable, comme ceci : « Getkey → G ». Il retourne une valeur sans stopper le programme : 0 si aucune touche n'est pressée, sinon le code de la touche. Les codes sont donnés par le tableau suivant, mais le petit programme ci-contre vous donnera le code de la touche sur laquelle vous appuierez.



```

While 1 // Equivaut à « While 1=1 »
ClrText
Do
Getkey → A
While A = 0
Locate 10, 4, A
WhileEnd

```

Il est ainsi possible de faire bouger un caractère grâce aux touches fléchées :

NB : Pour une condition qui ne modifie qu'une seule chose, le saut conditionnel (« \Rightarrow ») est bien plus pratique et efficace : « If A=1 Then 1 \rightarrow X IfEnd » est remplacé par « A=1 \Rightarrow 1 \rightarrow X »

Il se trouve ici : [SHIFT] / [VARS] (PRGM) / [F3] (JUMP) / [F3]

```
11 $\rightarrow$ X // la position en X du caractère
4 $\rightarrow$ Y // la position en Y
0 $\rightarrow$ G // la valeur du Getkey
While 1
Do
Locate X,Y,"O" // On affiche le caractère
Getkey $\rightarrow$ G
LpWhile G $\neq$ 28 And G $\neq$ 37 And G $\neq$ 28 And G $\neq$ 27
ClrText // on efface tout
G=27 $\Rightarrow$ X+1 $\rightarrow$ X // on fait bouger le caractère en fonction de la touche enfoncée
G=38 $\Rightarrow$ X-1 $\rightarrow$ X // idem...
G=37 $\Rightarrow$ Y+1 $\rightarrow$ Y // idem...
G=28 $\Rightarrow$ Y-1 $\rightarrow$ Y // idem...
X<1 $\Rightarrow$ 1 $\rightarrow$ X // Si le caractère va sortir de l'écran, on le retient à l'intérieur
X>21 $\Rightarrow$ 21 $\rightarrow$ X // idem...
Y<1 $\Rightarrow$ 1 $\rightarrow$ Y // idem...
Y>7 $\Rightarrow$ 7 $\rightarrow$ Y // idem...
WhileEnd
```

Conclusion :

Nous espérons que cet exemple vous aura inspiré d'autres idées pour vos futurs jeux, mais ce chapitre est maintenant terminé. Les quelques fonctions que nous avons abordées (Locate, ClrText, Int, Frac, Getkey, \Rightarrow) sont bien entendu les plus courantes, mais elles vous serviront beaucoup.

Le prochain cours que vous allez voir illustrera les notions abordées durant ces deux derniers tutoriels en vous proposant de créer un jeu « Question pour un Champion ». Vous allez enfin pouvoir épater vos amis en cours.

A bientôt sur la page Facebook Casio Calculatrices, et sur le site www.planete-casio.fr !