



Le DrawStat et les cours sur Casio



Vous vous apprêtez à lire un tutoriel rédigé par un membre de ce site. Malgré tout le soin que ce membre a pu apporter au tutoriel, nous ne pouvons pas garantir que les informations contenues sur cette page sont exactes à 100%. Merci de garder cela en tête lorsque vous lirez cette page ;o)

Auteur : pylaterreur

Créé le : Aujourd'hui à 13:32:53

Modifié le : Il y a 5 min



Noter et commenter ce tutoriel

Imprimer ce tutoriel

Editer ce tutoriel

Le tuto qui suit porte sur le BASIC Casio, et plus particulièrement sur le DrawStat. Le DrawStat est le moyen le plus rapide pour tracer des lignes et des points sur les Graph 35+, 65 et 85.

Pour comprendre ce tuto, il est impératif d'apprendre les bases du DrawStat, par exemple grâce à [ce tuto sur Planete Casio](#). Vous devez pouvoir le comprendre sans problèmes avant de continuer.

Il vous faut aussi des bases en programmation Casio : savoir utiliser les variables, les conditions, les boucles, ainsi que les listes, et savoir que l'utilisation des Labels (Lbl et Goto) est crée des bugs, et se révèle particulièrement lente (c'est pour ça qu'on en utilisera pas dans ce tuto 😊).



Je programme uniquement sur Graph 85 SD, donc s'il y a un bug sur Graph 35+ ou 65, faites-le moi savoir par le biais des commentaires ou des MP. 😊

A vos calculatrices 😊 !

Sommaire du chapitre :



[Astuces pour le DrawStat](#)

[Des cours...](#)

[du DrawStat...](#)

[Les deux en même temps](#)

[Q.C.M.](#)

Astuces pour le DrawStat

Des pointillés !

Le DrawStat peut être utilisé en mode Scatter, il trace alors des points non reliés les uns aux autres. Tous les bouts de code de cette partie "pointillés" doivent être précédés de :

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. ViewWindow 1,127,0,1,63,0
2. AxesOff
3. LabelOff
4. BG-None
5. FuncOff
6. S-WindMan
7. S-Gph1 DrawOn,Scatter,List 1,List 2,1,Dot
8. S-Gph2 DrawOff
9. S-Gph3 DrawOff

```

Pour créer des pointillés, on utilisera la fonction Seq(qui se trouve dans [OPTION]=>[F1]=>[F5]. Pour avoir une liste dont les valeurs vont de A à B de C en C, faites Seq(X,X,A,B,C). Les deux premiers paramètres sont à laisser tels quels, je ne sais pas exactement à quoi ils servent 🤔. Le 3ème paramètre est la valeur initiale, le 4ème la valeur finale et le 5ème le pas.

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. Seq(X,X,1,127,2)->List 1
2. Dim List 1->Dim List 2
3. Fill(32,List 2)
4. DrawStat

```

Cette méthode équivaut à cette seconde, sans la fonction Seq(, mais avec une boucle For :

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. 'Il faut d'abord que la List 1 ait une Dim suffisante pour que le code fonctionne, s
2. 0->X
3. For 1->A To 127 Step 2
4. X+1->X
5. A->List 1[X]
6. Next

```

Préférez la première méthode, car elle est plus rapide et elle prend moins de place dans le programme, et surtout elle est beaucoup plus facile à utiliser (vous n'avez pas à vous préoccuper de la Dim de la List 1). Pour les deux

méthodes, l'écran affiche ceci :



Dessiner en xyLine

Pour le reste de cette sous-partie, configurez votre DrawStat en mode xyLine :

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. ViewWindow 1,127,0,1,63,0
2. AxesOff
3. LabelOff
4. BG-None
5. FuncOff
6. S-WindMan
7. S-Gph1 DrawOn,xyLine,List 1,List 2,1,Dot
8. S-Gph2 DrawOff
9. S-Gph3 DrawOff

```

Cette mini-partie sera très courte, mais vous sera d'une importance capitale pour vos dessins en DrawStat : comment faire deux dessins en xyLine avec une seule utilisation de la fonction DrawStat ?

Vous savez que le mode xyLine dessine des points et les relie entre eux. Essayons de tracer deux traits parallèles, en tapant, par exemple, ceci :

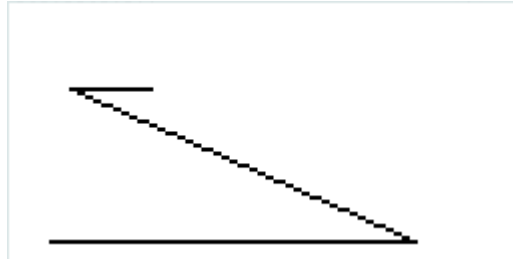
Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. {10,101,15,35}->List 1
2. {5,5,43,43}->List 2
3. DrawStat

```

Comme on pouvait s'y attendre, cela donne un mauvais rendu : les deux traits sont reliés :



Pour éviter cela, il faut utiliser une valeur intermédiaire entre les coordonnées des deux traits. Cette valeur intermédiaire peut être égale à 0 pour la Graph 35+, mais cette astuce ne marche pas sur la Graph 85. Pour cette dernière, utilisez une valeur strictement supérieure à 127, et utilisez plutôt une variable :

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. 128->D
2. {10,101,D,15,35}->List 1
3. {5,5,D,43,43}->List 2
4. DrawStat

```

Ce qui donne bien ce que l'on voulait :



Des cours...

Là, on attaque une partie essentielle, dans cette partie vous apprendrez comment créer un programme de cours léger et pratique, qui autorise à faire page précédente et page suivante, à créer un menu, etc.

Dans cette sous-partie, on ne touchera pas au DrawStat, mais on reprendra de plus belle dans la suivante 😊.

La structure du programme est... une boucle !

Pour ma part, j'ai choisis une boucle While, mais ça revient au même avec un LpWhile (c'est plus embêtant à faire avec un For, par contre, je vous le déconseille). On va aussi utiliser un seul Getkey pour savoir sur quelle touche l'utilisateur a appuyé, pour savoir si on doit aller à la page précédente ou à la page suivante ou pour quitter le programme, et des conditions pour afficher la page que l'on désire.

Code : Autre - Afficher / masquer les numéros de ligne

```

1. ViewWindow 1,127,0,1,63,0
2. 0->A~Z
3. 2->B
4. 'B+1 = nombre de pages dans le programme de cours
5. While G!=47
6. 0->G
7. Cls
8. If A=0
9. Then Text 1,1,"Page 1"
10. Else If A=1
11. Then Text 1,1,"Page 2"
12. Else If A=2
13. Then Text 1,1,"Page 3"
14. Text 10,5,"Ceci est la dernière page"
15. IfEnd
16. IfEnd
17. IfEnd
18. '(Tant que Getkey est différent de [->] ET que Getkey est différent de [EXIT] ET (ta
19. Do
20. Getkey->G
21. LpWhile G!=25 And G!=47 And (G!=38 Or A=0) And (G!=27 Or A>=B)
22. G=38=>A-1->A
23. G=27=>A+1->A
24. If G=25
25. 'Tant que la page demandée est avant la première ou qu'elle est après la dernière
26. Then Do
27. ""
28. Locate 1,1,"QUELLE PAGE (1~"
29. Locate 16,1,B
30. Locate 17+Int log B,1,")"
31. "ALORS "?->A
32. Rep A->A
33. ClrText
34. LpWhile (A<0 Or A>B) Or Frac A!=0
35. IfEnd
36. WhileEnd
37. Cls
38. Text 1,1,"Termine !"

```

du DrawStat...

Le DrawStat est beaucoup plus rapide pour tracer des traits que des F-Line 🤖. Il est aussi beaucoup plus pratique à utiliser pour celui qui veut créer des symboles qui se retrouveront en grand nombre dans son programme, comme dans un cours de mathématiques avec des flèches vectorielles. Avant de découvrir le DrawStat, je "m'amusais" à tracer les flèches une par une, ce qui prenait une place énorme dans mon programme, et rendait le code illisible 😞.



Dans cette troisième sous-partie, configurez le DrawStat en mode xyLine au lieu de Scatter, comme cela était pour la première sous-partie

Tout d'abord, je vous donne quelques idées de symboles :

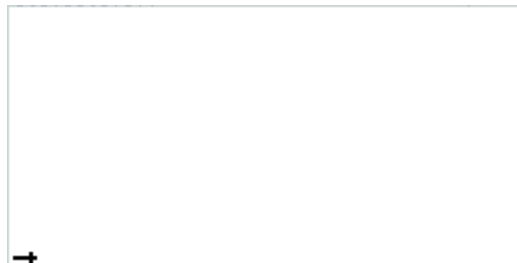
Une flèche de vecteur

Pour dessiner une flèche, on fait :

Code : Autre - Afficher / masquer les numéros de ligne

```
1. {1,6,5,5}->List 1
2. {2,2,3,1}->List 2
3. DrawStat
```

Ce qui nous donne ceci :



Il suffit alors de faire des additions sur les List 1 et List 2 pour déplacer la flèche

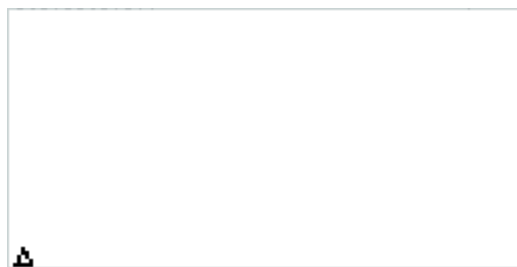
Un delta

Pour faire un "delta", on fait

Code : Autre - Afficher / masquer les numéros de ligne

```
1. {1,3,5,1}->List 1
2. {1,5,1,1}->List 2
3. DrawStat
```

Ce qui fait, en pas très joli 😞 :

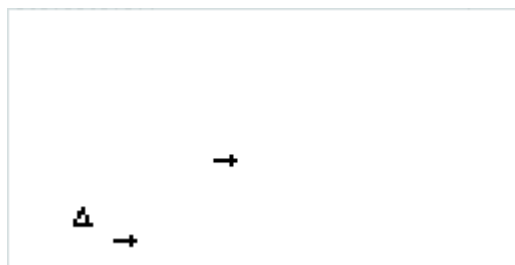


Je vais vous donner une méthode qui vous permettra de créer vos symboles et de les utiliser comme vous les voulez. On utilisera pour cela la List Ans, qui donnera le type de symbole à utiliser (flèche vectorielle, "delta", "infini", etc), son abscisse et son ordonnée. Ensuite, dans une boucle, on analyse la List Ans et on affiche ce que les 3 paramètres ont décidé.

Code : Autre - Afficher / masquer les numéros de ligne

```
1. {1,25,5,2,15,10,1,50,25}
2. For 1->X To Dim List Ans Step 3
3. If List Ans[X]=1
4. 'Si on demande une fleche
5. Then {1,6,5,5}+List Ans[X+1]->List 1
6. {2,2,3,1}+List Ans[X+2]->List 2
7. Else If List Ans[X]=2
8. 'Si on demande un delta
9. Then {1,3,5,1}+List Ans[X+1]->List 1
10. {1,5,1,1}+List Ans[X+2]->List 2
11. IfEnd
12. IfEnd
13. DrawStat
14. Next
```

Ce code nous affiche ceci :



Les deux en même temps

Vous avez désormais tous les éléments pour "construire" votre programme de cours en intégrant le DrawStat (2ème et 3ème sous-parties réunies) 😊

Je vais vous aider un peu : avant d'entrer dans la boucle For qui va tracer les éléments, vérifiez s'il y a quelque chose à dessiner, sinon il y aura un Dim Error (s'il n'y a pas de List Ans et que vous faites List Ans[X], ça foire 😞).

Le programme se décomposera grosso-modo alors en :

Début *boucle principale*

Page en fonction d'une variable

Boucle *DrawStat* s'il on veut afficher quelque chose

Boucle *Getkey*

Fin *boucle principale*

Voici le code commenté :

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. ViewWindow 1,127,0,1,63,0
2. 0->A~Z
3. 2->B
4. 'B+1 = nombre de pages dans le programme de cours
5. While G!=47
6. {0}
7. 'Très important : List Ans[1] vaut maintenant 0, ce qui veut dire que seules les pag
8. 0->G
9. Cls
10. If A=0
11. Then Text 1,1,"Page 1"
12. Else If A=1
13. Then Text 1,1,"Page 2"
14. Else If A=2
15. Then Text 1,1,"Page 3"
16. Text 10,5,"Ceci est la derniere page"
17. IfEnd
18. IfEnd
19. IfEnd
20. 'Si List Ans[1]!=0, ce qui empêche d'avoir une Dim Error si on n'affiche rien
21. If List Ans[1]
22. Then For 1->X To Dim List Ans Step 3
23. If List Ans[X]=1
24. 'Si on demande une fleche
25. Then {1,6,5,5}+List Ans[X+1]->List 1
26. {2,2,3,1}+List Ans[X+2]->List 2
27. Else If List Ans[X]=2
28. 'Si on demande un delta
29. Then {1,3,5,1}+List Ans[X+1]->List 1
30. {1,5,1,1}+List Ans[X+2]->List 2
31. IfEnd
32. IfEnd
33. DrawStat
34. Next
35. IfEnd
36. '(Tant que Getkey est différent de [->] ET que Getkey est différent de [EXIT] ET (ta
37. Do
38. Getkey->G
39. LpWhile G!=25 And G!=47 And (G!=38 Or A=0) And (G!=27 Or A>=B)
40. G=38=>A-1->A
41. G=27=>A+1->A
42. If G=25
43. 'Tant que la page demandée est avant la première ou qu'elle est après la dernière
44. Then Do
45. ""
46. Locate 1,1,"QUELLE PAGE (1~"
47. Locate 16,1,B
48. Locate 17+Int log B,1,")"
49. "ALORS "?->A
50. Rep A->A
51. ClrText
52. LpWhile (A<0 Or A>B) Or Frac A!=0
53. IfEnd
54. WhileEnd
55. Cls
56. Text 1,1,"Termine !"

```

Q.C.M.

Quel est le code à préférer parmi les deux suivants ?

- Premier code

Code : Autre - [Afficher / masquer les numéros de ligne](#)

```

1. Lbl 1
2. ?->A
3. A=215=>Goto 2
4. Goto 1
5. Lbl 2
6. "Mot de passe decouvert"

```

- Deuxième code

Code : Autre - Afficher / masquer les numéros de ligne

```

1. Do
2. ?->A
3. LpWhile A!=215
4. "Mot de passe decouvert"

```

- Le premier
 Le deuxième

Code : Autre - Afficher / masquer les numéros de ligne

```

1. {1,2,3}
2. {2,3,1}+List Ans

```

Que vaut List Ans[2] ?

- 5
 2
 3
 6



Grâce à ce tuto, si vous l'avez bien lu et si je l'ai bien écrit, vous avez appris comment faire des symboles pour vos cours de mathématiques (ou de physique-chimie), et ce de manière à avoir un programme léger et rapide, chose importante, car les 64Ko de mémoire et le processeur de la calculatrice sont vite essouffés 😞. Désormais, vous n'avez plus d'excuse pour faire des programmes moches ou lents ou qui bugent 🤪.

Si vous avez des suggestions ou des questions, dirigez-vous vers les commentaires 😊.



Auteur : pylaterreur

Noter et commenter ce tutoriel

Imprimer ce tutoriel

Editer mon tutoriel