Le DrawStat

Bonjour à tous ! Nous nous retrouvons pour aborder une fonction, certes assez com pliquée à utiliser, mais infiniment utile : le DrawStat (dessin de graphe statistique). Tout d'abord nous allons le découvrir dans son utilisation mathématique qui pourra vous être utile dans le cadre scolaire, et ensuite nous verrons son utilisation dans le cadre de la programmation de jeux. C'est en effet la fonction de dessin la plus puissante de votre calculatrice.

Utilisation statistique

Partons d'un problème : vous voulez représenter par une courbe l'évolution de la population de cochons volants en Loulouxie entre 2050 et 2060. Vous avez un tableau de données :

Année	2050	2052	2055	2057	2060
Individus	142	153	136	140	132

Pour commencer faisons ça à partir du menu *Statistiques*. Nous allons rentrer nos données dans les listes 1 et 2 :



Ensuite, il nous faut représenter la courbe de la population en fonction de l'année. Ce qui veut dire que la List 1 contient nos abscisses, et la List 2 nos ordonnées. Nous allons donc paramétrer un graphique. Pour cela rendez-vous dans les réglages de graphes : [F1] (GRPH) / [F6] (SET). Nous allons ensuite régler les paramètres pour nous mettre en mode *xyline*, avec la List 1 en X et la List 2 en Y, comme indiqué ci-dessous :



Ne vous souciez pas des paramètres *Frequency* et *Mark Type*, il faut juste veiller à ce que la fréquence soit toujours à 1 et que le marquage soit bien un point.

Nous allons maintenant afficher le graphique que nous venons de paramétrer. Pour ce la nous allons quitter la page de paramétrage et sélectionner le graphe à afficher : [EXIT] / [F1] (GPH1). Normalement vous devriez avoir un truc qui ressemble à ça :



Si vous n'avez rien, c'est peut-être que votre fenêtre d'affichage est réglée en manuel plutôt qu'en automatique. Pour changer cela faites [SHIFT] / [MENU] / [F1] (Auto).

Maintenant nous allons faire la même chose dans le menu *Programmes*, avec encore plus de souplesse, mais un peu plus de difficulté. Créons un nouveau programme, que nous allons appeler « DRAWSTAT ».



Nous allons, dans notre programme, entrer les données dans la liste, régler nous-mêmes la fenêtre graphique, paramétrer le graphe et afficher. Ne vous inquiétez pas, je vais tout expliquer pas à pas.

Voilà déjà les fonctions que nous allons utiliser et leur positionnement dans les sous-menus de la calculatrice (n'oubliez pas de bien ressortir à chaque fois du sous-menu précédent, et si vous ne trouvez pas, cherchez dans le catalogue : [SHIFT] / [4]) :

- S-WindMan : [SHIFT] / [F6] / [F6] / [F3] (S-WIN) / [F2] (Man)
- *ClearGraph* : [SHIFT] / [VARS] / [F6] / [F1] (CLR) / [F2] (Grph)
- ViewWindow : [SHIFT] / [F3] (V-Win) / [F1] (V-Win)
- *S-Gph1* : [F4] (MENU) / [F1] (STAT) / [F2] (GRPH) / [F1] (GPH1)
- *DrawOn* : [F4] (MENU) / [F1] (STAT) / [F1] (DRAW) / [F1] (On)
- Dot : [F4] (MENU) / [F1] (STAT) / [F4] (TYPE) / [F1] (MARK) / [F3] (.)
- DrawStat : [SHIFT] / [VARS] / [F6] / [F2] (DISP) / [F1] (Stat)
- xyline : [F4] (MENU) / [F1] (STAT) / [F2] (GRPH) / [F5] (xyline)

Et maintenant, construisons ce programme :

S-WindMan // activation du réglage manuel de la fenêtre graphique {2050,2052,2055,2057,2060}→List 1 // entrée des abscisses dans la List 1 {142,153,136,140,132}→List 2 // entrée des ordonnées dans la List 2 ClearGraph // effacement de l'écran graphique ViewWindow 2050,2060,0,130,160,0 // réglage adapté de la fenêtre graphique * S-Gph1 DrawOn,xyline,List 1,List 2,1,Dot // definition du graphe ** DrawStat // affichage

* nous avons fait que l'écran affiche en abscisse de 2050 à 2060 et en ordonnée de 130 à 160, en réglant l'échelle des axes à 0 pour ne pas être embêtés par l'affichage de l'échelle à gauche et en bas de l'écran.

** le graphe est défini grâce à un enchaînement de paramètres :

Graphe, activé ou non, mode, liste des X, liste des Y, fréquence, marquage Mais bon, pour ne pas nous embrouiller nous allons nous contenter de ceux -là.

Allez, testez ce code, normalement vous devriez obtenir :



Félicitations, vous venez de vous servir avec succès du DrawStat dans un programme, pour tracer une courbe statistique ! Maintenant nous allons voir son utilité dans le cadre de la programmation d'un jeu. Prêts ? On attaque !

Utilisation pour le dessin

Maintenant qu'on sait dessiner un graphique qui relie tous les points dont on a rentré les abscisses dans une liste et les ordonnées dans une autre, pourquoi ne pas nous servir de ça pour afficher des images rapidement ?

Nous allons commencer par l'affichage d'une image statique, et ensuite vous apprendrez à déplacer une image sur l'écran de votre calculatrice.

Tentons d'afficher une flèche, un dessin assez simple. Pour cela nous allons la décomposer en points à relier, et récupérer les coordonnées des points. Nous allons prendre un repère qui a pour origine le point en haut à gauche et pour échelle 1 pixel.



On obtient ainsi un tableau avec les abscisses et ordonnées de nos points. Il faut les écrire dans un certain ordre pour que les points soient bien reliés. Nous allons partir d'un point à gauche, tourner dans le sens des aiguilles d'une montre et revenir à ce point :

Abscisses	0	9	9	15	9	9	0	0
Ordonnées	6	6	2	8	14	10	10	6

Pour afficher la flèche il va falloir reprendre l'algorithme que nous avions programmé dans la première partie, en remplaçant les données des listes par les données ci-dessus, et en changeant les réglages de la fenêtre graphique. Nous allons prendre une fenêtre de 0 à 126 en abscisse et de 62 à 0 en ordonnée, puisqu'en Basic notre écran graphique mesure 127 pixels par 63, et qu'on veut que l'origine soit en haut à gauche.

Nous allons nous servir d'une seule fonction supplémentaire : AxesOff, pour supprimer les axes, dont on n'a pas besoin pour notre dessin. Elle se trouve dans : [SHIFT] / [MENU] / [F4] (AXES) / [F2] (Off).

Cela donne donc :

AxesOff // suppression des axes S-WindMan // activation du réglage manuel de la fenêtre graphique {0,9,9,15,9,9,0,0}→List 1 // entrée des abscisses dans la List 1 {6,6,2,8,14,10,10,6}→List 2 // entrée des ordonnées dans la List 2 ClearGraph // effacement de l'écran graphique ViewWindow 0,126,0,62,0,0 // réglage adapté de la fenêtre graphique S-Gph1 DrawOn,xyline,List 1,List 2,1,Dot // definition du graphe DrawStat // affichage

Et normalement vous devriez obtenir :



Vous y êtes ? Challenge suivant : sans changer les coordonnées de la flèche ni la fenêtre graphique, l'afficher à un autre endroit sur l'écran. Je vous dis tout de suite la technique : on peut ajouter un nombre à toutes les valeurs d'une liste, juste en faisant :

List numéro + nombre → List numéro

Maintenant nous allons nous servir de cette technique pour afficher notre flèche à partir du point de coordonnées (42 ; 26). Modifions notre algorithme :

AxesOff // suppression des axes S-WindMan // activation du réglage manuel de la fenêtre graphique {0,9,9,15,9,9,0,0}→List 1 // entrée des abscisses dans la List 1 {6,6,2,8,14,10,10,6}→List 2 // entrée des ordonnées dans la List 2 List 1+42→List 1 // on ajoute 42 à toute la List 1 des abscisses List 2+26→List 2 // on ajoute 26 à toute la List 2 des ordonnées ClearGraph // effacement de l'écran graphique ViewWindow 0,126,0,62,0,0 // réglage adapté de la fenêtre graphique S-Gph1 DrawOn,xyline,List 1,List 2,1,Dot // definition du graphe DrawStat // affichage

Normalement votre calculatrice affiche :



Mais, avec cette technique, on peut déplacer le dessin comme on veut ? Bingo ! On va pouvoir faire déplacer notre dessin par l'utilisateur. Vous avez appris précédemment dans ces cours à vous servir de boucles, de la double flèche pour des conditions avec une seule instruction à exécuter, et de la fonction Getkey, qui permet de savoir quelle touche est pressée. Il suffit donc de nous servir de tout ça pour créer notre algorithme. Pour ne pas réinitialiser le ViewWindow à chaque tour de boucle nous utiliserons Cls et non pas ClearGraph.

Voilà déjà les codes des touches dont vous pourrez avoir besoin :

- haut : 28
- -bas : 37
- gauche : 38
- droite : 27
- [EXIT] : 47

Je vous laisse réfléchir un peu avant de donner la solution...

Voilà donc notre algorithme final :

AxesOff // suppression des axes S-WindMan // activation du réglage manuel de la fenêtre graphique {0,9,9,15,9,9,0,0}→List 1 // entrée des abscisses dans la List 1 {6,6,2,8,14,10,10,6}→List 2 // entrée des ordonnées dans la List 2 ViewWindow 0,126,0,62,0,0 // réglage adapté de la fenêtre graphique S-Gph1 DrawOn,xyline,List 1,List 2,1,Dot // definition du graphe Do Cls // effacement de l'écran graphique DrawStat // affichage Getkey=27⇒List 1+1→List 1 // droite ; on ajoute 1 à toute la List 1 des abscisses Getkey=37⇒List 2+1→List 2 // bas ; on ajoute 1 à toute la List 2 des ordonnées Getkey=38⇒List 1-1→List 1 // gauche ; on enlève 1 à toute la List 1 des abscisses Getkey=28⇒List 2-1→List 2 // haut ; on enlève 1 à toute la List 2 des ordonnées LpWhile Getkey≠47 // tant qu'on n'appuie pas sur [EXIT]

Et voilà ! Votre flèche se déplace selon les flèches appuyées lors de l'exécution. Nous avons terminé notre gros cours sur le DrawStat. Bravo à vous de ne pas vous être découragés ! Nous nous retrouverons bientôt pour créer un jeu avec le DrawStat et les autres fonctions graphiques.

Vous pouvez, si vous êtes intéressés, regarder quelques petites astuces avec l'utilisation du Drawstat, ci-dessous.

Astuces avec le DrawStat

- Vous pouvez vous servir du DrawStat avec des variables. Il faut alors prendre soin de stocker le dessin dans d'autres listes et de rentrer dans les listes d'affichage la somme de vos listes d'enregistrement et de la variable. Par exemple, vous pouvez stocker la flèche dans les List 7 et List 8, sa position dans A et B, et faire List 7+A \rightarrow List1 et List 8+B \rightarrow List 2.

- Si vous voulez faire un menu avec une flèche pour pointer les options, vous pouvez stocker la position de la flèche dans une variable, veiller à ce que la valeur de cette variable reste entre 1 et le nombre d'options, et déduire grâce à cette variable l'ordonnée (menu vertical) ou l'abscisse (menu horizontal) à ajouter à votre List pour que la flèche se trouve face à votre option.

- Vous pouvez agrandir votre objet en multipliant les listes par un nombre (ou une variable bien sûr). Bien sûr les objets peuvent s'en trouver déformés, à vous de bien calculer. Pour rétrécir un objet il suffit de le diviser, ou de le multiplier par un nombre plus petit que 1.