

Chapitre 9 : Le mouvement

Avant de commencer à lire ce chapitre, assurez vous que vous maîtrisez bien les premiers.

Nous allons donc commencer par faire avancer un mot :

```
Cls
For 10→A to 60
Text 30,A,"Hello!!"
Next
```

Ce programme affiche un message qui défile de la gauche vers la droite. Mais comme vous pouvez le voir, il laisse derrière lui une traînée. Nous allons donc ajouter un rafraîchissement qui va effacer cette traînée derrière le message.

```
Cls
For 10→A to 60
Text 30,A," Hello!!"
Next
```

C'est juste le petit espace qui est placé avant le message qui va se charger d'effacer la traînée. Le message défile de la gauche vers la droite donc ce petit espace efface ce qui reste derrière ce message.

On peut aussi faire bouger des lignes

```
ViewWindow 1,127,0,63,1,0
20→A
10→B
Do
Isz A
Isz B
F-Line A,B,A-10,B+10
LpWhile B<50
```

« Isz » sert à incrémenter. Isz A correspond à $A+1 \rightarrow A$

Dans cette exemple, la ligne se déplace de haut en bas et de gauche a droite.

Chaque coordonnée de la ligne est exprimée en fonction de la variable qui augmente et donc fait bouger.

Cet exemple peut vous être utile lorsque vous avez affiché du texte, des lignes etc. et que votre écran est tellement chargé que lorsque vous voulez faire bouger quelque chose vous ne pouvez pas faire appel au rafraîchissement partiel sous peine d'effacer un élément présent au passage de cette chose en mouvement.

Nous allons donc reprendre l'exemple précédent , faire passer la ligne au dessus d'un texte sans effacer celui-ci et utiliser un rafraîchissement.

```
ViewWindow 1,127,0,63,1,0
Text 30,30,"Hello"
StoPict 1
20→A
10→B
Do
Cls
RclPict 1
Isz A
Isz B
F-Line A,B,A-10,B+10
LpWhile B<50
```

On affiche le texte et on enregistre une « photo » de l'écran grâce à la fonction « StoPict » Et a chaque passage de la ligne, on efface l'écran avec la commande « Cls » et on recharge notre photo de l'écran avec la commande « RclPict » pour ravoir notre situation initiale (écran vierge + texte).

Note : N'utilisez pas des lettres comme X ou Y pour coordonnées pour faire bouger des choses car la calculatrice s'en sert et donc votre programme a beaucoup de chance de bugger avec ces lettres.

Le dernier exemple devrait vous inspirer pour vos futur programmes : nous allons faire bouger une lettre avec les touches de direction.
Il faut savoir avant tout que la calculatrice dessine plus vite des lettres que des lignes.

```
61→A
31→B
Lbl 1
Cls
Text B,A,"O"
Lbl 2
0→K
Do
Getkey→K
LpWhile K=0
K≠28 And K≠38 And K≠37 And K≠27⇒Goto 2
Text B,A," "
If K=28
Then B-5→B
B<1⇒1→B
Goto 1
IfEnd
If K=37
Then B+5→B
B>56⇒56→B
Goto 1
```

```

IfEnd
If K=38
Then A-5→A
A<1⇒1→A
Goto 1
IfEnd
If K=27
Then A+5→A
A>121⇒121→A
Goto 1
IfEnd

```

Ce programme permet de déplacer une lettre avec les touches de direction et j'ai pris la peine d'ajouter des limites c'est-à-dire que le curseur ne peut pas sortir de l'écran.

```

ViewWindow 1,127,0,63,1,0
20→A
Text 20,24,"Nouveau Jeu"
Text 27,24,"Aide"
Text 34,24,"Quitter"
StoPict 1
Lbl 1
F-Line 20,A+1,20,A+3
PlotOn 21,A+2
Lbl 2
0→K
Do
GetKey→K
LpWhile K=0
K≠28 And K≠37 And K≠31⇒Goto 2
Cls
RclPict 1
If K=28
Then A-7→A
A<20⇒34→A
Goto 1
IfEnd
If K=37
Then A+7→A
A>34⇒20→A
Goto 1
IfEnd
K=31 And A=20⇒Text 1,1,"Menu : Nouveau Jeu"
K=31 And A=27⇒Text 1,1,"Menu : Aide"
K=31 And A=34⇒Text 1,1,"Menu : Quitter"

```

Ce dernier exemple vous montre comment faire un menu et connaître la position du curseur lorsque vous avez appuyé sur [EXE].