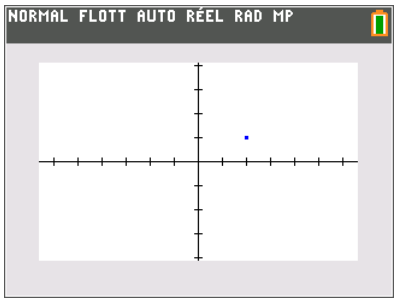
Dans cette seconde leçon de l'Unité 5 vous allez étudier comment tracer des points et la différence entre les commandes Pt-Aff et Pxl-Aff.

Objectifs :

- Utiliser les instructions de tracé des points et pixels
- Développer des formules pour utiliser des graphiques dans des programmes.

Commandes utilisant les Points



Voir le menu POINTS de [dessin].

Pt-Aff(x,y) trace un point selon les paramètres de la fenêtre graphique. Ceci signifie 'allume le point'. **Pt-Aff(2,1)** trace le point dans le premier quadrant dans l'écran de droite. **Pt-Aff(x,y,style,couleur)** a deux arguments optionnels : style (1 à 4) et couleur. Voir l'aide syntaxique en appuyant sur [+] lorsqu'on a sélectionné la commande dans le menu de [dessin].

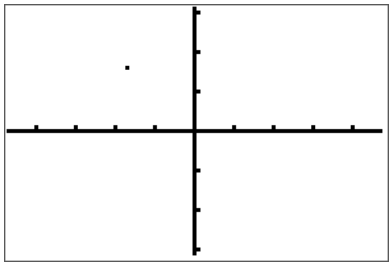
Pt-Aff(100,100) tracera un point même s'il est en dehors de la partie actuellement visible de la fenêtre.

Commandes utilisant les Pixels

Pxl-Aff(utilise les pixels de l'écran et ignore les réglages de la fenêtre. **Pxl-Aff(2,3)** trace le minuscule pixel dans le coin supérieur gauche de ce même écran à la ligne 2, colonne 3. C'est un peu dur à voir ! Les coordonnées ne sont pas dans l'ordre usuel (x,y), ils sont à l'envers car ils font référence à (n°ligne, n°colonne). **Pxl-Aff(x,y,couleur)** n'a qu'un argument optionnel : la couleur. Il y a aussi des commandes correspondantes **-NAff(,-Changer(et -Test(** que nous n'étudierons pas ici.

Pixels

Suivant votre calculatrice, votre écran a un nombre de colonnes et de lignes de pixels différent. Exemples : TI-84 Plus, 96 colonnes x 64 lignes, TI-83 Premium CE 265 colonnes x 165 lignes. Les lignes sont horizontales et les colonnes sont verticales. Les paramétrages du partage d'écran affectent le nombre de lignes et/ou de colonnes dépendant du paramétrage. L'écran graphique de la **TI-84 Plus** n'utilise pas la colonne la plus à droite et la ligne la plus basse pour la *représentation graphique*, il y a ainsi un nombre impair de points dans la zone graphique. Ceci assure l'existence d'un point 'central' (origine). Le pixel supérieur gauche est (0,0). **Pxl-Aff(0,0)** active le pixel en colonne 0, ligne 0. Les lignes et les colonnes sont numérotées en partant de zéro et non de un.



*Cet écran graphique de **TI-84 Plus** a des pixels plus grand, ils sont donc plus facile à voir. Cet écran montre le résultat de **Pxl-Aff(15,30)**. Le pixel est en ligne 15, colonne 30. Le pixel le plus bas à droite est en (63,95).*

Indication : Les commandes relatives aux pixels peuvent induire en erreur car l'ordre des coordonnées est inversé : la valeur de *y* (n° de ligne) est en premier et la valeur de *x* (n° de colonne) est en second. Également, la direction de l'axe des *y* est inversée : la ligne n°0 est en haut et la ligne n°165 (ou 63 sur une TI-84 Plus) est en bas de l'écran. C'est similaire à l'orientation de la grille de l'écran de calcul lorsqu'on utilise l'instruction **Output**((n°ligne, n°colonne, avec ligne 1 en haut). L'instruction **Texte(** qui permet d'écrire du texte dans l'écran graphique (vu dans une autre leçon) utilise aussi les pixels pour positionner le texte

et non les points.
Les instructions **-Test(** sont utilisées comme conditions dans une instruction **If** ou dans des boucles pour voir si un point ou un pixel est 'allumé'. Pt-Test(0,0) sera vrai quand le point (0,0) est 'allumé'.
Notez qu'un point peut être 'allumé' et blanc, et ainsi ne pas apparaître à l'écran !

Sur une TI-84 Plus les pixels et les points sont de même taille. Sur la TI-83 Premium CE les points sont plus grands que les pixels.

Programmation avec des Points

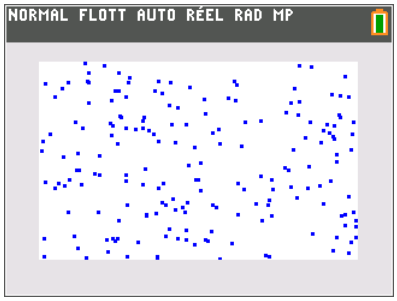
Écrivons un programme qui rempli de façon aléatoire l'écran GRAPHIQUE avec des POINTS. Ce programme aura une infinité de boucles aussi il faut appuyer sur la touche [on] pour interrompre le programme.

Nous aurons besoin d'un ***algorithme*** (formule) pour obtenir un point aléatoire dans l'écran GRAPHIQUE dans les limites de la fenêtre :
NbrAléat se trouve dans le menu [math] PROB et génère un nombre aléatoire entre 0 et 1. **NbrAléat * (Xmax-Xmin)** génère un nombre aléatoire entre 0 et Xmax-Xmin, ainsi nous ajouterons **Xmin**.
NbrAléat * (Xmax-Xmin)+Xmin génère un nombre aléatoire entre **Xmin** et **Xmax.**
…et nous écrivons une formule similaire pour les ordonnées Y.
C'est pour cela que les **mathématiques** sont si importantes pour programmer !

Note :
AxesNAff *est dans l'écran* [format].
FoncNAff *est dans le menu* [var] **VAR Y Aff/NAff**
GraphNAff *est dans le menu* [graph stats].
EffDess *est dans le menu* [dessin].
While 1 *crée une boucle infinie puisque* ***1*** *représente Vrai dans la calculatrice.*
Les valeurs aléatoires sont stockées dans **A** *et* **B** *pour être tracées*.
Pensez à appuyer sur [on] pour interrompre le programme.
Ce programme fonctionne de la même façon sur toute calculatrice de la famille TI-84 Plus !

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:FILLPTS
:GraphNAff
:EffDess
:While 1
:NbrAléat*(Xmax-Xmin)+Xmin
→A
:NbrAléat*(Ymax-Ymin)+Ymin
→B
:Pt-Aff(A,B)
:End
```

Ce programme fonctionne sur toutes les calculatrices TI-83 et TI-84.



Amélioration du programme FILLPTS avec la Couleur

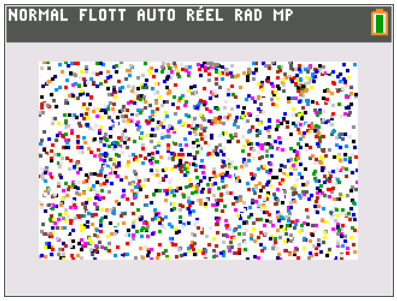
Sur la **TI-83 Premium CE** vous pouvez ajouter une couleur aléatoire à l'instruction **Pt-Aff(**. Le numéro de couleur le plus petit est 10 et le plus grand 24.

Écrivez une instruction qui génère un nombre aléatoire entre 10 et 24 en utilisant **nbrAléatEnt(** et ajoutez-la comme *troisième* argument à l'instruction **Pt-Aff(**.

Ajoutez l'instruction suivante avant **Pt-Aff(A,B).**
Changer **Pt-Aff(A,B)** en **Pt-Aff(A,B,C).**

<votre générateur de couleur> → **C**

Note : Il y a deux arguments optionnels dans **Pt-Aff(** *: le style et la couleur. Le style est un entier compris entre 1 et 4, la couleur entre 10 et 24. Si le troisième*



Avec des points en couleur !

argument est un entier compris entre 1 et 4 c'est un style. S'il est compris entre 10 et 24 c'est une couleur. Toute autre valeur provoque une erreur.

Réponse : nbrAléatEnt(10,24)→C