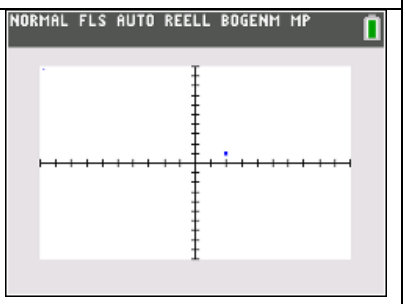
Lektion 5: Grafik	Übung 2: Punkte und Pixel
In der zweiten Übung von Lektion 5 geht es um das Zeichnen von Punkten und den Unterschied zwischen den Befehlen **Punkt-Ein(** und **Pxl-Ein(**.	**Lernziele:** • Verwendung der Befehle zum Zeichnen von Punkten und Pixeln. • Entwicklung von Formeln zur Nutzung von Grafik in Programmen.

Der Punkt - Befehl

Man findet ihn im Menü [DRAW] PKTE. **Punkt-Ein(x,y)** zeichnet einen Punkt unter Berücksichtigung der FENSTER-Einstellungen. Der Punkt wird „eingeschaltet". **Punkt-Ein(2,1)** „schaltet" also im ersten Quadranten bei P(2/1) einen Bildschirmpunkt ein. **Punkt-Ein(** hat noch optional die Argumente Zeichen und Farbe (vergl. mit der *Hilfe*, die man durch Drücken von [+] bekommt, während man den Befehl in [DRAW] auswählt): **Punk-Ein(x,y,Zeichen,Farbe)**.
Punkt-Ein(100,100) zeichnet einen Punkt selbst wenn er außerhalb des aktuellen Grafikfensters ist.

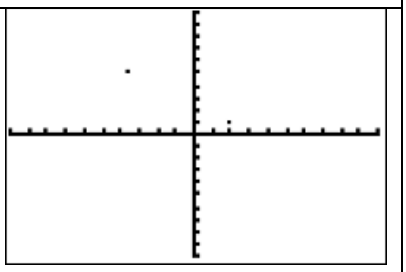


Der Pixel - Befehl

Pxl-Ein(verwendet die Bildschirm-Pixel völlig unabhängig von den Fenstereinstellungen. **Pxl-Ein(2,3)** zeichnet einen winzigen Punkt in die obere linke Ecke in die Reihe 2 Spalte 3. Die Koordinaten sind nicht in der üblichen Reihenfolge (x/y) angeordnet sondern wirken vertauscht, denn es kommt erst die Zeilennummer, dann die Spaltennummer. **Pxl-Ein(** hat nur ein optionales Argument: **…,Farbe): Pxl-Ein(Zeile,Spalte,Farbe)**. Es gibt noch die Befehle **Pxl-Aus(, Pxl-Ändern(** und **pxl-Test(**, die hier aber nicht weiter behandelt werden sollen.

Pixel

Abhängig vom Taschenrechnertyp der TI-84 Familie ist die Anzahl von Spalten und Zeilen unterschiedlich. Der TI-84 Plus hat 96 Spalten und 64 Zeilen, der TI-84 Plus CE-T hat 265 Spalten und 165 Zeilen. Zeilen verlaufen in x-Richtung und Spalten in y-Richtung. Bei einem aufgeteilten Display ändert sich die Anzahl der Zeilen und Spalten entsprechend. Beim TI-84 Plus werden die untere Zeile und die rechte Spalte nicht bei der Graphik verwendet, damit durch die dann ungerade Anzahl sichergestellt ist, dass es einen Bildmittelpunkt gibt. Das Bild zeigt den Grafikbildschirm eines TI-84 Plus mit einem Punkt bei Zeile 15 Spalte 30 (**Pxl-Ein(15,30)**). Wegen der anderen Auflösung sind die Pixel dicker und besser zu sehen. Der Pixel in der linken oberen Ecke hat stets die „Koordinaten" (0/0). **Pxl-Ein(0,0)** schaltet also diesen Punkt bei Zeile 0 und Spalte 0 ein. Die Zeilen- und Spaltennummerierung beginnt stets mit 0, nicht 1! Die untere rechte Ecke hat beim TI-84 Plus die Koordinaten (63/95), beim TI-84 Plus CE-T (165/265).

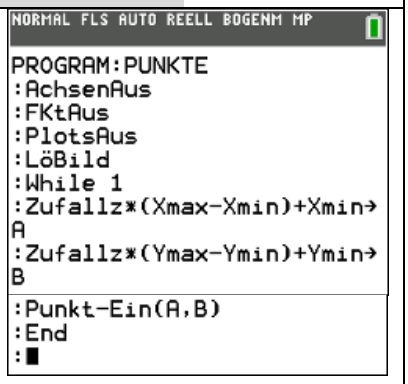


Hinweis: Pixelbezogene Befehle können etwas verwirrend sein, weil die Reihenfolge bei den Koordinaten umgekehrt ist: erst y (Zeilennummer), dann x (Spaltennummer). Außerdem verläuft die Nummerierung von oben nach unten: Zeile 0 ist oben und Zeile 165 (bzw. 63 beim TI-84 Plus) ist unten. Die gleiche Orientierung findet man aber auch beim Befehl **Output(** . Der Befehl **Text(**, der zur Ausgabe von Text auf dem Grafikbildschirm verwendet wird (in einer späteren Übung), nutzt ebenfalls Pixel und nicht Punkte zur Positionierung.
Der Befehl **pxl-Test(** wird in **If** – Befehlen oder Schleifen verwendet um abzufragen ob ein Pixel gesetzt ist. **pxl-Test(0,0)** ist wahr, wenn der Pixel (0/0) gesetzt ist. Das kann auch bei

Pixeln sein, die die Farbe weiß haben und dadurch nicht sichtbar sind!
Auf einem TI-84 Plus haben Pixel und Punkte dieselbe Größe. Auf einem TI-84 Plus CE-T sind Punkte größer als Pixel.

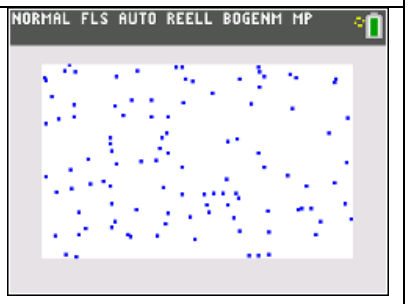
Programmieren mit Punkten

Nun soll ein Programm geschrieben werden, dass auf dem GRAPH-Bildschirm zufällig Punkte setzt. Das Programm verwendet eine unendliche Schleife, so dass man auf [ON] drücken muss, um das Programm zu beenden.
Dazu wird ein ***Algorithmus*** (Formel) benötigt, um einen Punkt innerhalb der Fenstergrenzen zufällig auszuwählen.
Die Funktion **Zufallz** aus dem Menü [MATH] WAHRS erzeugt eine Zufallszahl zwischen 0 und 1. **Zufallz·(Xmax-Xmin)** erzeugt eine Zufallszahl zwischen 0 und **Xmax - Xmin**. Addiert man noch **Xmin**, so erhält man eine Zufallszahl zwischen **Xmin** und **Xmax**: **Zufallz·(Xmax-Xmin)+Xmin** .
Auf gleiche Weise lässt sich eine Formel für die y-Koordinate erstellen.
Das Beispiel zeigt, wie wichtig **Mathematik** für das Programmieren ist!



Hinweise:

AchsenAus *findet man bei [FORMAT].*
FnOff *findet man bei [VARS] Y-VARS Ein/Aus.*
PlotsOff *findet man im Menü [STAT PLOT].*
ClDraw *ist im Menü [DRAW].*
While 1 *erzeugt eine unendliche Schleife, da der Wert* wahr *im Taschenrechner durch die Zahl 1 dargestellt wird.*
*Die Zufallszahlen werden in den Variablen **A** und **B** gespeichert.*
Mit [ON] wird das Programm abgebrochen.
Das Programm läuft auf allen Taschenrechnern der TI-84-Familie.



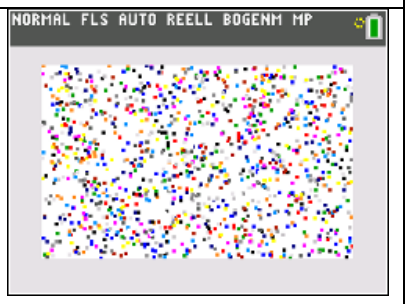
Erweiterung: Farbe

Beim **TI-84 Plus CE-T** kann man noch eine zufällig ausgewählte Farbe dem Befehl **Punkt-Ein(** hinzufügen. Die niedrigste Farbzahl ist 10 und die höchste 24.

Dazu muss ein Befehl geschaffen werden, der eine zufällige natürliche Zahl im Bereich von 10 bis 24 erzeugt und als drittes Argument **Punkt-Ein(** hinzufügt. Diese Zahl kann durch den Befehl **zufInt(** erzeugt werden. Der Befehl muss vor **Punkt-Ein(A,B,C)** (Änderung beachten!) eingefügt werden:

<Farbgenerator> → **C**

*Hinweis: **Punkt-Ein** hat zwei optionale Argumente: Zeichen und Farbe. Zeichen kann die Werte von 1 bis 4 annehmen, Farbe von 10 bis 24. Hat das dritte Argument also einen Wert von 1 bis 4, so wird es als Zeichenattribut interpretiert. Liegt der Wert zwischen 20 und 24, so ist es eine Farbe. Andere Werte erzeugen eine Fehlermeldung.*



Lösung: zufInt(10,24) →C