



Lektion 6: Zeichnen

Übung 3: Zeichnen eines Punkts

In dieser Übung wirst du ein Programm schreiben, um einen Punkt zu zeichnen, diesen automatisch zu bewegen und dabei seine Koordinaten anzuzeigen. Mit `[esc]` wird das Programm abgebrochen.

Lernziele:

- Zeichnen eines Punkts
- Diesen Punkt automatisch bewegen
- Anzeigen seiner Koordinaten
- Mit `getKey(0)` auf einen Tastendruck warten

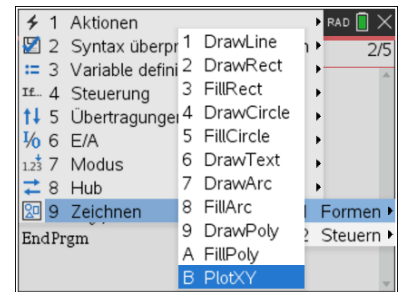
Zeichnen von Punkten

Bis jetzt hast du Formen (Gerade, Rechteck, Kreis, Bogen) gezeichnet. Es gibt ein weiteres Werkzeug. **PlotXY** lässt dich einen Punkt mit seinen Koordinaten (x,y) setzen.

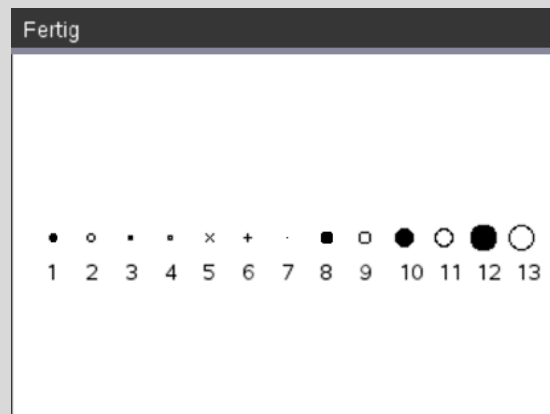
Die Syntax dieser Anweisung ist: **PlotXY** x, y, *Stil*

x, y sind die Koordinaten des Punkts, auf Basis der aktuellen Fenstereinstellung.

Stil ist ein Wert von 1 bis 13 und bestimmt Größe/Form des Punkts. Er ist anzugeben.



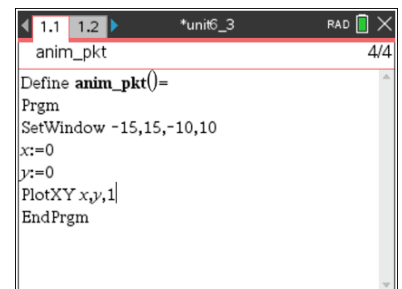
Hinweis: **DrawText** wird später erklärt. Die **DrawPoly** und **FillPoly**-Anweisungen werden in dieser Übung nicht behandelt. **Plot XY** hat 13 *Stile* für die Form eines Punkts.



Zeichnen eines Punkts

1. Beginn ein neues Programm und nenn es **anim_pkt**.
2. Stell angenehme Fensterwerte ein: **SetWindow -15, 15, -10, 10**
3. Beginn mit dem Zeichnen eines Punkts in den Koordinatenursprung:
 - x:=0**
 - y:=0**
 - PlotXY x, y, 1**

PlotXY (zu finden über **menu > Zeichnen > Formen**) zeichnet (x,y) im Stil 1. Wir verwenden die Variablen **x** und **y**, da sie im Programm geändert werden, um den Punkt zu bewegen. Probier einige andere Stile (1 -13 für den dritten Parameter).





Die Hauptschleife

Wir werden die **getKey(0)**-Anweisung (*menu > I/O*) zur Abfrage der **[esc]** Taste für den Programmabbruch verwenden:

4. Füg hinter die **PlotXY**-Anweisung ein:

k:=” ” (nichts zwischen den Anführungszeichen)

While k#”esc”

(Schleifenkörper)

k:=getKey(0)

EndWhile

Hinweis:

getKey(0) übernimmt die letzte Tastatureingabe, ohne das Programm zu unterbrechen. In diesem Programm wird die Schleife (und damit das Programm) mit **[esc]** beendet.

```
unit6_3 RAD
"anim_pkt" erfolg. gespeichert
SetWindow -15,15,-10,10
x:=0
y:=0
PlotXY x,y,1
k:=""
While k#”esc”
  k:=getKey(0)
EndWhile
EndPrgm
```

Bewegen des Punkts

Die Schleife wird den Punkt durch Veränderung der x-Koordinate sich *scheinbar* bewegen lassen. Doch vorerst muss er von seiner aktuellen Stelle entfernt werden. Wir „löschen“ ihn, indem wir ihn in der Hintergrundfarbe (weiß) zeichnen ...

5. Schreibe in den *Schleifenkörper* zwischen **While** and **getKey(0)**:

SetColor 255,255,255 Farbe weiß

PlotXY x, y, 1 lösche aktuellen Punkt

x:=x+1 erhöhe x um 1

SetColor 0, 0, 0 Farbe schwarz

PlotXY x, y, 1 zeichne den nächsten Punkt

Wait 0.5 warte eine halbe Sekunde

Wait findet sich über *menu > Steuerung* und erteilt dem Rechner den Auftrag, den Prozess für diese Anzahl von Sekunden zu unterbrechen. Füh das Programm aus und du wirst den Punkt nach rechts wandern sehen. Er wird dann vom Schirm verschwinden. Drück **[esc]**, um das Programm zu beenden und verbessere es so, dass der Punkt innerhalb der Zeichengrenzen bleibt:

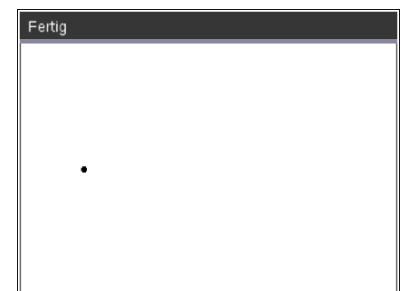
```
*Unit 6 RAD
movepoint 15/16
While k#”esc”
  SetColor 255,255,255
  PlotXY x,y,1
  x:=x+1
  If x>15x:=-15
  SetColor 0,0,0
  PlotXY x,y,1
  Wait 0.5
  k:=getKey(0)
EndWhile
```

6. Füg nach **x := x+1** die folgenden Anweisungen ein:

if x > 15 rechten Schirmrand erreicht?

x := -15 dann kehr zurück an den linken Rand

Herausforderung: Kannst du den Punkt umkehren lassen, so dass er sich nach Erreichen des rechten Rands wieder nach links bewegt – und dann wieder nach rechts ...?





Hinweis: So kehrt man am Rand des Bildschirms um:

Vor der Schleife:

`j := 1` (oder eine andere Zahl ... das ist die Schrittweite)

In der Schleife:

`x := x + j` (anstelle von `x := x + 1`)

`if x > 15 or x < -15` (das sind die Werte für die Schirmränder aus **SetWindow**)

`j := -j`

Anzeigen der Koordinaten

Während der Anzeige des Zeichenschirms sind E/A-Anweisungen wie **Disp** und **Request** nicht verfügbar. Um Text im Zeichenfenster auszugeben müssen wir die Anweisung **DrawText** (über *menu > Zeichnen > DrawText*) verwenden.

Syntax: **DrawText** Objekt1, Objekt2, Objekt3, ...

Objekte sind: Zeichenketten wie "Hello, world!"

Zahlen

Variable

Ausdrücke

Beispiel (ohne Programmierung): Gib im Calculator ein:

drawtext 30, 50, "Der Wert von x ist ", x

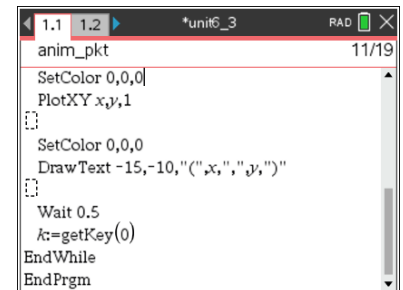
(Dein Wert für x ist möglicherweise ein anderer!)

7. Aber zurück zu unserem Programm. Füg einige Anweisungen zur Anzeige der Koordinaten des bewegten Punkts vor der **Wait**-Anweisung ein.

SetColor 0, 0, 0 *Farbe schwarz*

DrawText -15, -10, "(, x, ", ", y, ")" *zeige (x,y)*

Führ das Programm aus und beobachte, wie die Koordinaten in der linken unteren Ecke der Zeichenfläche angezeigt werden. Leider werden die Werte jeweils über die vorhergehenden geschrieben und damit unleserlich!





8. So geht's: Lösch den Bereich, in dem die Koordinaten angezeigt werden durch Zeichnen eines weißen gefüllten Rechtecks *vor* dem Schreiben der neuen Koordinaten. Füg hinzu:

SetColor 255, 255, 255

Farbe weiß

FillRect -15, -10, 8, 3

überdeckt die Koordinaten

Genau über *SetColor 0, 0, 0*

Farbe schwarz

DrawText -15, -10, "(" , x, " , " , y, ")"

schreibt (x,y)

Starte das Programm nochmals. Besser?

Führ eigene Verbesserungen durch Änderung von Stil und Farbe des Punkts und der Form der Bewegung durch.

```

1.1 1.2 *unit6_3 RAD 17/19
anim_pkt
SetColor 0,0,0
PlotXY x,y,1
SetColor 255,255,255
FillRect -15,-10,6,3
SetColor 0,0,0
DrawText -15,-10,"( , x, " , " , y, )"
Wait 0.5
k:=getKey(0)
EndWhile
EndPrgm

```

Hinweis:

- Die Durchführung einer Animation erfordert das Löschen des alten Bildes und das Zeichnen eines neuen. Manchmal benötigt man dafür zusätzliche lokale Variable, um „alten“ Daten im Auge zu behalten
- Die Darstellung der Koordinaten mit Klammern und Komma erfordert genaue Kleinarbeit mit Anführungszeichen und Kommas. Bleib geduldig! In dieser Anweisung kommen einige Kommas vor.
- **Clear -15,-10, 8, 3** wirkt hier wie **FillRect**: der rechteckige Bereich wird gelöscht.
- Und nun folgt das vollständige Programm:

Define anim_pkt()=

Prgm

```

: SetWindow -15, 15, -10, 10
: x := 0 : y := 0
: PlotXY x, y, 1
: k: = ""
: While k ≠ "esc"
: SetColor 255, 255, 255
: PlotXY x, y, 1
: x := x+1
: If x > 15
: x := -15
: SetColor 0, 0, 0
: PlotXY x, y, 1
: SetColor 255, 255, 255
: FillRect -15, -10, 6, 3
: SetColor 0, 0, 0
: DrawText -15, -10, "(" , x, " , " , y, )"
: Wait 0.5
: k := getKey(0)
:EndWhile
:EndPrgm

```