



Lektion 6: Zeichnen

Anwendung: Manuelle Animation

Schreib ein Programm, das ein Objekt mit Hilfe der Pfeiltasten am Schirm bewegen lässt (interaktive Animation). Das ist die Grundlage vieler Videospiele.

Lernziele:

- Tastatureingaben mit **getKey(1)** auswerten
- Ändern von Koordinaten abhängig von Eingaben über die Tastatur
- Löschen und Neuzeichnen einer Form

In der vorigen Übung hast du einen Punkt automatisch über den Bildschirm bewegen lassen und mit **getKey(0)** das Programm beendet. Jetzt wirst du ein Programm schreiben, mit dem der Benutzer den Punkt (oder eine andere Form) mit den Pfeiltasten bewegt. Dazu wirst du **getKey(1)** verwenden; es hält den Prozess an und wartet auf die nächste Tastatureingabe. Nur die Pfeiltasten („hoch“, „tief“, „rechts“, „links“) und die `esc`-Taste sind dafür wichtig.

Der Beginn: Zeichnen des Punkts

1. Beginn ein neues Programm **anim_pkt2**.
2. **VERWENDE NICHT SetWindow**.
Das voreingestellte Zeichenfenster ist (0...317) x (0...211).
3. Der Startpunkt liegt nahe beim Zentrum des Schirms (159,106):

```
x := 159
y := 106
PlotXY x, y, 1
```

Die Hauptschleife

4. Die Hauptschleife läuft bis die `esc`-Taste gedrückt wird


```

k := getKey(1)      warten auf Tastendruck (beginnen)
While k ≠ "esc"    so lange bis esc
    Schleifenkörper
    k := getKey(1)    warten auf Tastendruck (fortsetzen)
EndWhile
EndPrgm

```

Hinweis:

getKey(1) hält den Programmfluss an und wartet auf den nächsten Tastendruck, dann wird das Programm fortgesetzt. Dieses Programm berücksichtigt die vier Pfeiltasten und die ESC-Taste. Alle anderen Tasten bleiben unberücksichtigt.

Die Pfeiltasten

5. Die vier Pfeiltasten heißen "up", "down", "left" und "right". Mit **k** übernehmen wir die eingegebene Richtung und ändern entsprechend **x** oder **y**:


```

If k = "left"
    x := x - 1    (1 ist angenommen; könnte größer sein)

```

 Erinnere dich, das ist die einfachste Form der **If**-Anweisung (weder **Then** noch **Else**). Ergänze die drei anderen **If**-Anweisungen für die restlichen Pfeiltasten.



Neuzeichnen des Punkts

1. Zeichne den Punkt nach den vier **If**-Anweisungen mit den neuen Koordinaten.

PlotXY x, y, 1

```

1.1 1.2 *Dok RAD 7/14
anim_pkt2
k:=getKey(1)
While k≠"esc"
  If k="left": x:=x-1
  If k="right": x:=x+1
  If k="up": y:=y-1
  If k="down": y:=y+1
  PlotXY x,y,1
k:=getKey(1)

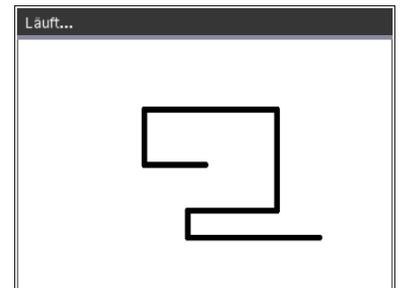
```

Hinweis: Wenn `esc` das Programm nicht abbricht (es läuft in einer Endlosschleife) dann drück mehrmals die `fn-on`-Taste um den Programmablauf zu unterbrechen.

Programmtest

Verwend die Pfeiltasten und beende das Programm mit `esc`. Wenn alles richtig ist, bewegt sich der Punkt ... und hinterlässt eine Spur.

Was geschieht, wenn der Punkt den Schirmrand erreicht? Jetzt verlässt er den Zeichenbereich. Nun gibt es mehrere Möglichkeiten: einfach stehen bleiben und nicht weiter gehen, irgendwohin gehen oder wie üblich, auf die entgegengesetzte Seite des Schirms wechseln. Das wird auch als *Rundum*-Methode bezeichnet.



Projektabschluss

In Übung 3 hast du gelernt, wie man einen Punkt „ausradiert“. Ergänze das Programm so, dass sich der Punkt nur scheinbar bewegt.

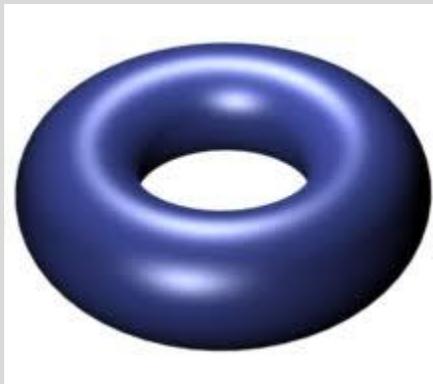
Ergänze den Code um vier **If**-Anweisungen, um das Randproblem zu lösen:

```

If x > 317
  x:= ?

```

Hinweis: „Toral“ bezieht sich auf eine Bewegung auf einem Torus (eine „Donut“-Form ohne Enden), in der dich „hoch“ zurück auf den Boden und „links“ auf deine rechte Seite führt. Jeder „gerade“ Weg windet sich um den Torus.





Kannst du eine ganze Gruppe von Formen bewegen?

Bis jetzt haben wir nur einen Punkt manuell bewegt. Überleg, den Punkt durch einige *Formen* zu ersetzen. Eine wirksame Möglichkeit ist es, dafür ein eigenes Programm (ein Unterprogramm) zu schreiben. Schreib im aktuellen Problem ein neues Programm – **formzeich(x,y)** – und verwende dann in **anim_pkt2** die Anweisung

formzeich(x, y) anstelle von **PlotXY x,y,1**.

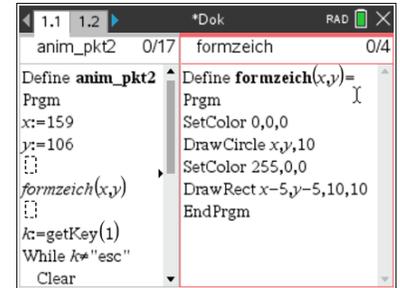
Rechts kannst du ein Beispiel sehen.

Zeichne in **formzeich(x, y)** einige Formen und verwend **(x,y)** als Bezugspunkt. Dein Objekt kann aus mehreren Teilen in unterschiedlichen Farben bestehen.

Überleg, die **Clear**-Anweisung (**menu > Draw > Clear**) zum Löschen der Figuren einzusetzen. **Clear** ohne Argumente leert den Schirm. Um nur einen rechteckigen Teil zu löschen, nimm:

Clear x, y, Breite, Höhe (ähnlich wie **DrawRect**)

Viel Spaß dabei!



Hinweis: **Clear** färbt den Bereich weiß. Bei geänderter Hintergrundfarbe verwend eine große **FillRect**-Anweisung und nimm **FillRect** nochmals, um die Hintergrundfarbe zu löschen.

Mathematische Herausforderung: setz im **formzeich()**-Beispiel von oben die Ecken des Rechtecks auf den Kreis (das Rechteck soll dem Kreis eingeschrieben sein). Welche Werte sind dann in der **DrawRect** -Anweisung anzugeben?

Hier ist eine mögliche Lösung (ohne Berücksichtigung der Schirmränder):

```
Define anim_pkt2()=
Prgm
x := 159
y := 106
PlotXY x, y, 1
k := getKey(1)
While k ≠ "esc"
  SetColor 255, 255, 255
  PlotXY x,y,1
  If k = "left"
    x := x - 1
  If k = "right"
    x := x + 1
  If k = "up"
    y := y - 1
  If k = "down"
    y := y + 1
  SetColor 0, 0, 0
  PlotXY x, y, 1
```



```
k := getKey(1)
EndWhile
EndPrgm
```

Füg vier weitere **If**-Anweisungen zur Berücksichtigung der Schirmränder nach Abfrage der Pfeiltasten ein:

```
If x < 0
  x := 0          (um anzuhalten; nimm x := 317 für den Rundherum-Modus)
If x > 317
  x := 317       (oder x := 0 ...)
...etc.
```

Es gibt aber elegantere Methoden.

Wenn du Zugang zu einem TI-Innovator Rover hast, dann versuch, die Rover-Bewegungen mit den Bewegungen des Punkts am Schirm gleichzuschalten (etwa mit Send "RV TO XY ...").