

OPL Workshop 8 - Zeichnen mit Grips

Ulrich Krinzner, 11/1999

Nachdem wir bereits über einen großen Teil von OPL-Grafik berichtet haben, nehmen wir uns heute das eher einfache Thema "Zeichnen" vor.

Wollen Sie in Ihren OPL-Programmen Grafiken benutzen? Dann haben Sie im wesentlichen zwei Möglichkeiten. Entweder, Sie laden bestehende Grafiken dazu oder Sie erzeugen Sie unmittelbar durch entsprechende Befehle.

Die erste Art verwenden Sie, wenn es sich um komplexe Gebilde handelt, die vielleicht sogar Graustufenbilder sind - das ist aber nicht unser heutiges Thema. Die zweite kommt in Frage, wenn es eher um kleine schwarz/weiße Skizzen oder optische Trenner auf dem Bildschirm geht.

Einfach, aber schnell

Zu diesem Zwecke verfügt OPL über eine Reihe von einfachen Befehlen, mit denen sich Linien und leere als auch gefüllte Rechtecke erzeugen lassen. An den Geräten mit EPOC 32 sind außerdem auch Kreise und Ellipsen per Befehl verfügbar. Diese Ausstattung ist spartanisch und doch lassen sich "gehobenere" Projekte verwirklichen.

Wir schrieben in Ausgabe 13 im Marktplatz über das Adventure Game "Haunted House" (Download von unserer WebSite www.palmtop-pro.com/ möglich) - dessen Grafiken werden nahezu vollständig durch diese einfachen Befehle während des Programmlaufes gezeichnet. OPL-Grafiken bauen also recht schnell auf.

Langweilig wird es, wenn man eigene Routinen verwenden muss, um Flächen zu füllen, die nicht durch die Standardbefehle erzeugt werden, z.B. ein Fünfeck. Darauf richtet man sich bei seiner Planung einfach ein und vermeidet diese Art von Darstellung. Bringen Sie etwas Fantasie auf, denken Sie einfach quer, Sie werden sich wundern, was trotzdem alles geht!

Das Prinzip

Sehen wir uns die Befehle an. Zuerst muss, wie inzwischen schon häufiger geübt, der Cursor positioniert werden. Da es sich um Grafikbefehle handelt, kommt nur "gAT x%,y%" in Frage. Um es noch einmal ins Bewußtsein zu bringen: "gAT" positioniert pixelgenau und vor allen Dingen absolut, x% beschreibt die horizontalen, y% die vertikalen Position auf dem Bildschirm.

Erinnern Sie sich: Der Ursprung 0,0 des Bildschirm-Koordinatensystems liegt links oben. Positive x-Werte liegen rechts davon, negative links. Relative horizontale Bewegung gehen vorzeichenabhängig dementsprechend in die gleichen Richtungen. Positives y geht zählt nach unten, negatives nach oben, sowohl absolut, als auch relativ.

Nach der Cursorpositionierung schließt sich der Grafikbefehl an.

Wollen Sie eine Linie ziehen, geben Sie dem entsprechenden Befehl nur noch die x/y-Koordinaten des Endpunktes in Pixel mit:

```
gAT 10,10 : gLINETO 50,50
```

Absolut und relativ

Stellen Sie sich vor, Sie wollen eine größere Skizze mehrfach an verschiedenen Positionen auf dem Bildschirm verwenden. Daher legen Sie die Skizze in ein Unterprogramm ab, das Sie dann immer wieder aufrufen. Gut gedacht, aber durch die Verwendung absoluter Koordinaten haben Sie schlechte Karten!

Hier hilft OPL mit Befehlen, deren Parameter als relative Koordinaten ausgewertet werden. Die Parameter sagen also nur noch, um wieviel Pixel versetzt gezeichnet werden soll. Damit wird ein Unterprogramm dann tatsächlich effektiv wirksam.

```
gAT 10,10 : gLINEBY 40,40
```

Eine solche Relativbewegung ist auch für den Cursor möglich. Er wird verschoben, ohne dabei zu zeichnen:

```
gMOVE -40,-40
```

Sowohl nach "gLINETO x%,y%" als auch nach "gLINEBY dx%,dy%" nimmt der Cursor den Endpunkt der Linie als neue Position an. "gMOVE dx%,dy%" bringt ihn hier in unserem Beispiel also wieder an die vorherige Position zurück.

Rechtecke

Ein Rechteck ist genauso simpel zu zeichnen:

```
gAT 10,10 : gBOX 50,50
```

Dem "gBOX b%,h%"-Befehl gibt man jedoch keine Endkoordinaten, sondern gleich die Breite und die Höhe des Rechtecks mit. Die Cursorposition wird nicht geändert! Versuchen Sie einmal dieses:

```
gAT 10,10 : gBOX 10,50 : gBOX 50,10 : gBOX 25,25
```

Ein gefülltes Rechteck erzeugt man so:

```
gAT 10,10 : gFILL 50,50,gmode%
```

Je nach "gmode%" fällt die Füllung schwarz (0) oder weiß (1) aus. Die "schwarze" Füllung lässt sich auch anders tonen, wenn zusätzlich vor dem Zeichenbefehl noch ein "gGREY hell%" (alle EPOC Varianten) oder "gCOLOR r%,g%,b%" (EPOC32) angewandt wird (vorherige Folgen nachlesen!). Wird für gmode% eine "2" verwendet, wird invertierend gezeichnet. Beispiel:

```
gAT 10,10 : gFILL 50,50,0 : gMOVE 25,25 : gFILL 50,50,2
```

"gFILL" lässt sich auf diese Art auch geschickt zu "rechteckigem Löschen" verwenden. Reines Invertieren kommt auch ohne den dritten Parameter aus:

```
gINVERT breite%,hoehe%.
```

Aber Achtung, die Eckpixel werden nicht mit bearbeitet!

Rund nur mit EPOC32

Was EPOC16 nicht bietet, hat EPOC32 an Bord (wer es braucht, muss eigene Routinen aufstellen und rechnen lassen):

```
gCIRCLE radius% REM nur Umriss
gCIRCLE radius%,fill% REM gefüllt, wenn fill%<>0
gELLIPSE hradius%,vradius%
gELLIPSE hradius%,vradius%,fill% REM gefüllt, wenn fill%<>0
```

Die Position "gAT x%,y%" bilden in allen Fällen den Mittelpunkt. Für die Ellipse ist "hradius%" der Abstand zwischen Mittelpunkt und linkem bzw. rechtem Rand der Ellipse, "vradius%" ist der entsprechende Wert für oberen bzw. unteren Rand.

Möchten Sie auch diese Figuren "weiß" zeichnen oder invertieren, setzen Sie den gewünschten Zeichenmodus mit dem Befehl "gGMode modus%" vor die Kreis- bzw. Ellipsen-Befehle. modus%=1 zeichnet weiß, modus=2 invertiert, modus%=0 stellt den normal zeichnenden Zustand wieder her.

```
gAT 100,100 : gELLIPSE 50,20,1      REM gefüllte Ellipse
gGMOVE 10,10 : gGMode 2 : gELLIPSE 50,20,1
```

Beginnen Sie zu ahnen, mit wie wenig Mitteln man Grafiken effektiv aufbauen kann?

Ersatz für gPOLY

Einen Befehl lassen wir hier unter den Tisch fallen, obwohl er sehr schnell ist. Es geht um "gPOLY array%()". Der Befehl zeichnet mehrere Linien hintereinander, Vielecke werden so recht schnell gezeichnet. Allerdings bezahlt man das mit der Vorbereitung eines Feldes, in dessen Aufbereitung man etwas Grips investieren muss. Für einfache Aufgaben, etwa ein Dreieck oder ein "schräges Rechteck", ein Trapez, stellen wir lieber eigene Unterprogramme auf, z.B.:

```
PROC dreieck:(dx1%,dy1%,dx2%,dy2%)
    LOCAL lx%,ly%
    lx%=gX : ly%=gY      REM akt. Cursorpos. merken
    gLINEBY dx1%,dy1%
    gLINEBY dx2%,dy2%
    gLINETO lx%,ly%
ENDP
```

So ruft man dann das Unterprogramm ab:

```
PROC main:
    gAT 80,50 : dreieck:(30,30,-60,0) : GET
ENDP
```

Rationell programmieren

Wann immer möglich: lassen Sie den Computer für sich arbeiten! Bevor Sie zu selbstgeschriebenen Routinen greifen, sehen Sie sich an, welche Erleichterungen OPL bereits bietet.

Wollen Sie rechteckigen einen Bereich mit einem wiederkehrenden Muster füllen, können Sie das mit "gPATT bitmapId%,breite%,hoehe%,modus%" bewerkstelligen. Das Muster beziehen Sie aus einer bestehenden Bitmap-Grafik oder Sie erzeugen sie wie hier "on the go". Der Inhalt der Bitmap-Grafik wird auf eine Fläche von "breite%*hoehe%" ins aktuelle Fenster übertragen. "mode%" ist wie so häufig der Zeichenmodus

```
PROC tapete:
    GLOBAL bm%, win%
    win%=gIDENTITY : gAT 80,80
    bm%=gCREATEBIT(10,10)      REM Muster erzeugen
    gBOX 10,10 : gLINEBY 5,7 : gLINEBY 5,-7
    gUSE win% : REM zurück zum Hauptfenster
    gAT 10,10 : gPATT bm%,100,100,0
    REM gAT 120,20 : gCOPY bm%,0,0,10,10,0
    GET
ENDP
```

Eine Füllung können Sie auch erreichen, ohne dass Sie eine andere Grafikdatei zur Hand haben. Dann müssen Sie allerdings mit einem vorgefertigten Grau-Muster vorlieb nehmen. Benutzen Sie als ersten Parameter, also sonst normalerweise die Bitmap-ID, eine „-1“:

```
gPATT -1,40,40,0
```

Wenn Sie im letzten Beispiel das REM in der Zeile vor dem GET entfernen, lernen Sie eine weitere Methode kennen, aus einer bestehen Bitmap-Grafik Nutzen zu ziehen:

```
gCOPY bm%,x%,y%,breite%,hoehe%,modus%
```

An die aktuelle Position im aktuellen Fenster wird aus dem geöffneten oder gerade erstellten Grafikfenster mit der ID "bm%" ein rechteckiger Bereich kopiert, der bei den Koordinaten "x%" und "y%" beginnt und dessen weitere Abmaße durch "breite%" und "hoehe%" gegeben sind. Ob setzend, löschend oder invertierend kopiert wird, legt wie gehabt "modus%" fest.

Wenn Sie EPOC32 benutzen, ziehen Sie zusätzlich Vorteil aus dem Befehl:

```
gSETPENWIDTH breite%
```

Damit wird die Linienstärke eingestellt. Linien, die mehr als ein Pixel breit sind, sind somit schnell gezeichnet.

Durch die "Farbart" oder durch Aktivieren des löschenden Zeichnens erzielen Sie interessante Effekte:

```
gAT 100,100
gSETPENWIDTH 50 : gLINEBY 50,0 : gLINEBY 0,50
gSETPENWIDTH 46 : gMODE 1 : gLINEBY 0,-50 : gLINEBY -50,0
```

Als letztes stellen wir zur Anregung noch ein Beispiel vor, das zeigt, mit wie wenig Aufwand man recht komplexe Figuren erzeugen kann.

```
PROC kachel:
  GLOBAL vz%, n%
  gAT gWIDTH/2,gHEIGHT/2
  vz%=1 : n%=1
  DO
    gLINEBY vz%*n%,vz%*n%
    gLINEBY vz%*n%,-vz%*n%
    vz%=vz%*-1
    n%=n%+2
    PAUSE 5
  UNTIL n%>gHEIGHT/2
  GET
ENDP
```

Nun sind Sie wieder gefragt. Experimentieren Sie fleißig.