

Verspielte Graphik für harmlose Hobbyisten

Helmut Schierer

DSK-517,518

\$38_SHOW.pas

Das Programm "\$38_SHOW.pas" ist mit BP 7.0, Real-Mode, erstellt. Es wird nur der Interrupt 10h verwendet, keine Unit Graph, keine BGI-Treiber. Für VGA 1024 x 768 x 256 (Mode 38h); getestet mit TSENG ET 4000, 1MB auf einem DX-486, 33 MHz. Der praktische Sinn des Programms ist es, einer vorhandenen Graphik durch Farbmanipulation ein anderes Aussehen zu verleihen.

Es bietet also die überaus nützliche Möglichkeit, Graphiken, welche in stundenlanger Rechenarbeit erstellt wurden und durch ungeschickt gewählte Farbgebung nicht ganz so wurden, wie man sie wollte, nachträglich in ihrer Farbgebung so zu modifizieren, daß sie das erstrebte Aussehen erhalten.

Zur Bereitstellung von Test-Bildern liest das Programm geeignet gespeicherte Bilder mit zugehöriger Palette (s. "MANDELVB.pas"). Zur Erinnerung: 256-Farben-Graphik heißt, daß gleichzeitig 256 Paletteneinträge über ihre "Farb"-Nummer (0..255) angesprochen werden können. Im Mode 38h dürfen die Rot|Grün|Blau-Intensitäten die Werte 0..63 annehmen.

Kurzbeschreibung

Im Programm werden die globalen Variablen MyPal und MyPalOrg vom Typ Pal_256=array [0..256,0..2] of byte als 256-Farben-Paletten verwendet. Der Eintrag mit dem Index 256 dient dabei als Hilfsgröße.

Die Indizes 0..2 stehen für R G B.

Graphik-Routinen

Die Procedure vm(mode: byte) setzt den Video-Mode (38h=Graphik, 03h=Text).

Die Procedure px(x, y: integer; fb: byte) entspricht 'putpixel' an der Stelle x,y des Schirms in der Farbe fb=Paletteneintrag-Nummer.

Die Procedure SetCol orBlock(VAR Colors) setzt 256=AnzPal Farbbregister ab Register 0 mit den in der Tabelle Colors (=MyPal [0..255]) enthaltenen Werten. In BX wird die Palettennummer, in CX die Anzahl der zu setzenden Einträge übergeben. Achtung: Colors muß untypisiert sein; es dient zur Ermittlung von Segment und Offset des ersten zu setzenden Farbeintrags (Rot des ersten Paletteneintrags).

In der Procedure AlterOnePal sehen Sie das Setzen eines einzelnen Paletteneintrags.

Alle übrigen Routinen bedienen sich des move-Befehls als flinker Methode zum Datenschaukeln. Das Setzen der neuen Farben wird dann mit SetCol orBlock erreicht.

Programmbedienung

Aufruf: \$38_SHOW [pic-filename]. Wird kein Dateiname angegeben, so wird C:\X als default verwendet. Unmittelbar nach dem Programmstart wird die folgende, manchmal vielleicht etwas kryptische, Informationsseite angezeigt:

```
Shows $38-Pic "C:\X" or Paramstr; THEN type
ESC: hal t=> Abbruch, Ende Programm auf
'#': AlterOnePal => alternante Pal (Input #, R,G,B <return>)
'@': RestorePal => Paletten UND Bild -> Original
'P': RestorePic => NUR Bild -> Original, MyPal bleibt
'p': SetCol orBlock(MyPal) => Bild -> Bild mit aktuellem MyPal
'n': RotateOnePal => Pal (#) -> Pal (#-1), #=1..256, 1 Mal
'N': RotatePal => RotateOnePal, 256 Mal
'M': RotatePal Stop=> RotatePal, weiter mit <key> oder <ESC>
'c': ConfuseOne=> Pal (0,1)->Pal (0,0)..Pal (256,0)->Pal (255,2)
'c': Confuse=> ConfuseOne 256*3 Mal
'X': ConfuseStop=> ConfuseOne, weiter mit <key> oder <ESC>
'0'..'2': Change(ch)> Pal (#,0)->Pal (#,1)->Pal (#,2)->Pal (#,0)
'r', 'g', 'b': RotateOneCol or(ch)> 1 Farbe: Pal (#)->Pal (#-1), 1 Mal
'R', 'G', 'B': RotateCol or(ch)> RotateOneCol or, 256 Mal
^r, ^g, ^b: RotateCol orStop(ch)> RotateCol or, weiter mit <key> oder <ESC>
'j', 'k', 'l': IntensOneCol or(ch)> fb=(fb+1) mod 64, 1 Mal
'J', 'K', 'L': IntensCol or(ch)> IntensOneCol or, 64 Mal
^j, ^k, ^l: IntensCol orStop(ch)> IntensCol or, weiter mit <key> oder <ESC>
'i': IntensOnePal => alle 3 Farben: fb=(fb+1) mod 64, 1 Mal
'I': IntensPal => IntensOnePal, 64 Mal
^i: IntensPal Stop=> IntensPal, weiter mit <key> oder <ESC>
```

Erläuterungen

Haben Sie die Angabe eines Dateinamens nicht vergessen oder ist Ihnen C:\X recht, so setzen Sie den weiteren Programmlauf durch Drücken einer beliebigen Taste ungleich <ESC> in Gang, das Bild wird, leider langsam, spaltenweise geladen und gezeigt.

Erst wenn die Beendigung des Ladevorgangs akustisch signalisiert wurde, können Sie eine der vorher gezeigten Auswahlen treffen.

Tip: Eine Hardcopy der Informationsseite stärkt das Gedächtnis.

Eingabemöglichkeiten (i.a. ohne <return>)

- ESC Ende des Programmlaufs bzw. einer aktuellen Schleife (s.u.)
- # gemeint ist hier die jeweils mit <return> beendete Eingabe einer gültigen Farbnummer, gefolgt von den für sie zu setzenden RGB-Anteilen; z.B. 55,0,63,0 (N°55, grün)
- @ MyPal (die verwendete Palette) und das gezeigte Bild werden in den ursprünglichen Zustand versetzt
- P nur das Bild-Original, MyPal (die Tabelle) bleibt unverändert
- p das Bild wird in den Farben des aktuellem MyPal angezeigt. Die im folgenden beschriebenen Änderungen werden nicht nur in MyPal ausgeführt, sondern auch sofort am angezeigten Bild wirksam (^ steht dabei für Strg bzw. ctrl):
- n zyklischer Tausch; jede Farbnummer wird um eins erniedrigt, die ursprüngliche Farbe N° 0 erhält dabei die N° 255
- N rasche Abfolge der obigen Option, 256 Mal, also bis zum Ausgangszustand M: wie 'N', nur schrittweise; weiter mit <key>; <ESC> = Schleifenabbruch
- c pro Farbnummer: Rot erhält die Intensität von Grün, Grün jene von Blau und Blau erhält die Intensität vom Rot der um eins größeren Farbnummer; Blau[255] erhält dabei die Intensität von Rot[0]
- C rasche Abfolge der obigen Option, 256*3 Mal, also bis zum Ausgangszustand X: wie 'C', nur schrittweise; weiter mit <key>; <ESC> = Schleifenabbruch
- 0 in allen Farbnummern werden die Intensitäten von Rot und Grün vertauscht 1: in allen Farbnummern werden die Intensitäten von Grün und Blau vertauscht 2: in allen Farbnummern werden die Intensitäten von Blau und Rot vertauscht
- r|g|b zyklischer Tausch der Intensität von Rot|Grün|Blau in der Palette: z.B.: Rot[0]=Rot[1],...,Rot[254]=Rot[255] und Rot[255]=Rot[0]
- R|G|B rasche Abfolge der obigen Option, 256 Mal, also bis zum Ausgangszustand ^r|^g|^b: wie oben, nur schrittweise; weiter mit <key>; <ESC> = Schleifenabbruch
- j|k|l die Intensität der Farbe Rot|Grün|Blau wird für jede Farbnummer um 1 vergrößert, 64 wird zu 0
- J|K|L rasche Abfolge der obigen Option, 64 Mal, also bis zum Ausgangszustand ^j|^k|^l: wie oben, nur schrittweise; weiter mit <key>; <ESC> = Schleifenabbruch
- i die Intensität der Farben Rot,Grün und Blau wird gleichzeitig für jede Farbnummer um 1 vergrößert, 64 wird zu 0
- I rasche Abfolge der obigen Option, 64 Mal, also bis zum Ausgangszustand
- ^i: wie 'I', nur schrittweise; weiter mit <key>; <ESC> = Schleifenabbruch

Abschließend einige Kleinigkeiten

Die Bilddateien sollten 787.200 Bytes groß sein; besser mit XCOPY kopieren. Beim Betrachten der Bilder Helligkeit und Kontrast des Schirms relativ hoch einstellen, da sonst niedrige Intensitätswerte nicht erkannt werden. Optionen mit "rascher Abfolge" sind nur zur Gewinnung eines ersten Überblicks über die sich ergebenden Möglichkeiten brauchbar (zu rasch), zur genaueren Betrachtung eignen sich nur die "schrittweise" arbeitenden Optionen (diese mit ESC abbrechen und mit einer anderen Option weiterarbeiten erlaubt praktisch alle denkbaren Einfärbungen einer Graphik).

MANDELVB.PAS

Der Hauptzweck des Programms "MANDELVB.pas" ist die Erstellung von Testbildern für das Programm "\$38_SHOW.pas". Bedingungen

s.d., auch die globale Variable MyPal sowie die Procedures vm, px und SetCol or Block sollen nicht neuerlich beschrieben werden.

Das "Apfelmännchen" (bzw. Ausschnitte daraus) wurde als Beispiel gewählt, weil es sich gut zur Demonstration der Effizienz des Programms \$38_SHOW.pas eignet: nach geschickt gewählten Farbänderungen könnte man glauben, ein völlig anderes Bild vor sich zu haben.

Die vorliegende Programmversion ist trotz ihrer ca. 200 Zeilen als eher rudimentär zu bezeichnen, da ein Großteil des Codes der Bereitstellung von Auswahlmöglichkeiten gewidmet ist: 7 Farbpaletten und mehr als 30 verschiedene Ausschnittangaben stehen zur Auswahl (dazu im Hauptprogramm in `init_mandl` (`BildNo`) und `init_graph` (`PaletteNo`) ändern). Ein unschätzbare Vorteil des Programms ist die Möglichkeit, sich durch überschreiben der typisierten Konstanten `xstep` und `ystep` relativ rasch eine Vorschau auf die Graphik verschaffen zu können, welche nicht gespeichert wird. Sehr rasch (1% Rechenzeit bei 10|10 statt 1|1) erkennt man, ob der Ausschnitt den Vorstellungen entspricht.

Zum "Apfelmännchen" selbst

Es handelt sich um die graphische Darstellung der Mandelbrot-Menge mit Hilfe des Escape-Time-Algorithmus.

Die Mandelbrotmenge ist die Menge aller komplexen Zahlen C , für welche der Grenzwert der iterierten Gleichungen der quadratischen Funktion $z=z^2+C$ mit dem Startwert $z=(0|0)$ endlich ist.

Verständlicher ausgedrückt

Jedem Punkt des Bildschirms werden die Koordinaten $(x|y)$ eines Punktes $C=x+j.y$ der Gaußschen Zahlenebene zugeschrieben. Mit $z=(0|0)$ beginnend berechnet man $z=z^2+C$ und zählt mit (Variable `i` in der Funktion `iter`), wie oft man ein neues z berechnet. Der Abbruch der Berechnungen erfolgt, wenn entweder eine vorgegebene Größe (im Programm `rq`) von z^2 überschritten wird oder wenn die Anzahl an Berechnungen eine gleichfalls vorgegebene Größe (`MaxIter`) überschreitet. Aufgrund der Größe `i`, des sogenannten Fluchtindex, ermittelt man die Farbe, in welcher der Punkt C dargestellt wird. Je größer `rq` und `MaxIter`, desto differenzierter ist die Graphik und desto länger dauert ihre Erstellung. Ungeduldige rechnen etwa mit `rq=4` und `MaxIter=15` (vor allem bei nur 16 Farben sehr beliebt).

Mit ESC kann nach jeder Berechnung des Fluchtindex der Programmablauf abgebrochen werden.

OLYRI NGE. pas (Olympische Ringe, VGA)

Dieses Programm ist die Antwort auf die periodisch zum gegebenen Anlaß gestellte Frage "Wie programmiert man die olympischen Ringe?". Peinlicherweise weiß zwar jeder, daß es sie gibt, aber beim Versuch, ihre Farben zu ermitteln, tauchen die ersten Widersprüche auf, und wenn man gar über die Anordnung der Ringe und ihre Überschneidungen Bescheid wissen will, erzielt man selten Einigkeit.

Also: SO sehen sie aus und SO kann man sie programmieren.

Der Ablauf im Detail

Hintergrund Hellgrau (sonst sieht man den schwarzen Ring sehr schlecht) Feststellung: die Procedure `FillBox` muß man selbst programmieren, die Unit `GRAPH.TPU` stellt aber `FillPoly` zur Verfügung.

Die 3 oberen Kreisringe zeichnen und mittels der Procedure `Paint` (Marke Eigenbau aus `SetFillStyle` und `FillPoly`) füllen.

Feststellung: mit Schwarz klappts nicht so richtig, also pfuschen. Die 2 unteren Kreisringe zeichnen, mittels `Paint` füllen. Überschneidungen richtigstellen: der linke obere Kreisring ist in seinem rechten senkrechten Teil sichtbar, der rechte obere in seinem linken unteren Teil. Fertig. (Details s. Kommentare in der Source).

UNKRAUT_. pas (EGA/VGA)

Weit entfernt von Lindemayerpflanzen läßt das Programm dennoch Jahre (4=Anzahl Wachstumsperioden) hindurch das Unkraut sprießen. Es handelt sich um ein Beispiel zur rekursiven Programmierung, das vom Thema her keinerlei Fachkenntnisse irgendeines Wissenszweiges benötigt. Der Programmablauf bietet jedoch die Möglichkeit, den Ablauf einer Rekursion optisch zu verfolgen und eventuelle Unklarheiten zu

beseitigen. Der Recordtyp wird verwendet, einer raschen Umsetzung in eine OOP-Version steht nichts im Wege.

Gegeben: zwei Pflanzenteile (als Ast mit Knoten und als Blüte bezeichnet) und einer Wachstumsvorschrift für jeden einzelnen Wachstumschritt:

Blüte -> Ast + Blüte links + Ast + Blüte rechts + Blüte geradeaus ;
Ast -> Ast + Ast.

Gesucht: Darstellung der Wachstumsstadien.

Lösung : Die rekursive Procedure WACHSEN.

Die Source ist so ausreichend kommentiert, daß eine weitere Beschreibung nicht notwendig erscheint.

KILIMAND. bas

in QBASIC, VGA 320 x 200 x 256, gefunden als SNOWFALL.BAS in der Zeitschrift DOS 5'95, Seite 153.

Jeder, der mühsam aus der Zeitung abtippt und dann zusehen muß, wie Schneeflocken unter 45° nach links unten schwarze Löcher in die Berge bohren, fragt sich, ob das so sein soll. Das Durchlesen der Beschreibung läßt anderes vermuten, eine eher entsprechende Version, garniert mit Osterreminiszenzen, wird hiemit zur Verfügung gestellt; auch Neulinge im Programmieren haben ein Anrecht auf Erfolgserlebnisse. Kommentare im Programm. Wer schickt uns die VB-Version ?

Die übrigen QBASIC-Files haben eines gemeinsam: Graphikprogramme, welche „unmögliche Körper“ darstellen.

Sie sind wirklich nur zum erstellen und ansehen der Bilder geeignet, es sei denn, man hat in QBASIC Schwierigkeiten mit der Graphik.

Nicht allgemein bekannt ist vielleicht nur die flinkere Möglichkeit, 256-Farben-Paletten zu erstellen, welche die 320x200-Graphiken verwenden: nicht PALETTE verwenden, sondern die Ports direkt ansprechen, wie z.B. in 1-BALK. bas :

OUT &H3C8, i: OUT &H3C9, r: OUT &H3C9, g: OUT &H3C9, b

Alle Programme

Disk 1 enthält alle Programme, Disk 2 nur \$38-SHOW, MANDELVB und 5 Bilder

Disk 517

```

| $38_DEMO.EXE          Demo zum Ergebnis von $38_SHOW
| MANDLOOV.PAL          dazu benötigtes file
| MANDLOOV.SCR          "-"
+---REAL_BP7.O
|   $38_SHOW.EXE       1024 x 768 x 256  ansehen und die
|   $38_SHOW.PAS              Farben ändern
|   MANDELVB.EXE           "-"          "Apfelmännchen"
|   MANDELVB.PAS              erstellen und speichern
+---ZUSATZL.PRG
|   EGAVGA7.OBJ
|   FONTS_1.TPU
|   OLYRI NGE.EXE
|   OLYRI NGE.PAS        Olympische Ringe '94
|   UNKRAUT_.EXE
|   UNKRAUT_.PAS        rekursiv wachsendes Unkraut
+---QBASIC
|   KILIMAND.BAS        DOS 5'96: Schneefall aufhängig gemacht
|                       (fast) nur zum Ansehen:
|                       Sortiment "unmöglicher" Körper
|   1-BALK.BAS
|   123-BALK.BAS
|   4-BALK.BAS
|   ARCHITKT.BAS
|   HEIDEN.BAS
|   MONUMENT.BAS
|   RAHMEN.BAS
|   SCHACH.BAS
|   TRIANGEL.BAS
    
```

Disk 518

\$38.exe ist ein selbstentpackendes Archiv. Entpacken: in gewünschte Ziel-Directory wechseln und dort A:\$38 eingeben.

```

Inhalt:  MANDELVB.pas & exe ... zeichnen und speichern von
         "Apfelmännchen" und Teilansichten; 1024 x 768 x 256
         $38_SHOW.pas & exe ansehen derselben,
         Farben (fast) beliebig variieren

         URMANDL.PIC
         MANDL001.PIC
         MANDL002.PIC
         MANDL009.PIC
         MANDL00V.PIC ... 5 Testbilder dazu
    
```

Speicherplatzbedarf : 4 MB ☐