Achtung, dies ist ein alter Text, der mit Programmieren am Computer (in diesem Falle am Mac, mit HyperCard) zu tun hat.

Er wurde auf Wunsch jener in die Textschublade gestellt, die nach VON NULL AHNUNG ZU ETWAS EDV wissen wollten, ob es noch Artikel zu Programjieren von mir aus der damaligen Zeit gäbe.

Es scheint nur noch diesen zu geben. Ich hatte damals 3 Programmier-Handbücher geschrieben, aber weder die Computer (C64 und der erste „FatMac") noch die Software exisiteren heute (20 Jahre später) noch (Damals hatten wir am Mac Betriebs-System 3.4).

Also, wer mit programmieren nichts an Hut hat, sollte diesen Text bitte

NICHT LESEN.

Alle anderen, bitte umblättern

HyperCard for the rest of us

Vera F. Birkenbihl

Am besten klären Sie gleich ab, ob Sie zur (direkten) Zielgruppe gehören: Stellen Sie sich eine **Linie** vor - das **Spektrum der HyperCard-Benutzer**. (Abb. 1) Nun sitzen am einen Ende die Nur-Anwender (bis Level 3 und, etwas zögerlich manchmal bis level 4, also *Authoring*) während am anderen Ende die "Hacker" sind, die in der Regel schon vor ihrer Mac-Zeit programmiert haben, oft sogar in mehreren Sprachen. Dieser Typ fragt nicht: **Wie programmiere ich** mit HyperTalk?, er will nur wissen: Wie programmiere ich **mit HyperTalk?**

	Author & Leser dieser Serie:	
Nur-Anwender	wir anderen: THE REST OF US	Mac-Hacker Programmierer

Aber es gibt noch eine **dritte Gruppe.** Das sind all die *anderen*, der sog. "**rest of us**". Sie möchten zwar mehr als "nur anwenden", aber Sie sind keine Programmierer. Und Sie tun sich (anfangs) ein wenig schwer.

Denn das Märchen, daß jeder in Minuten HyperTalk lernen würde, glaubt heute niemand mehr.

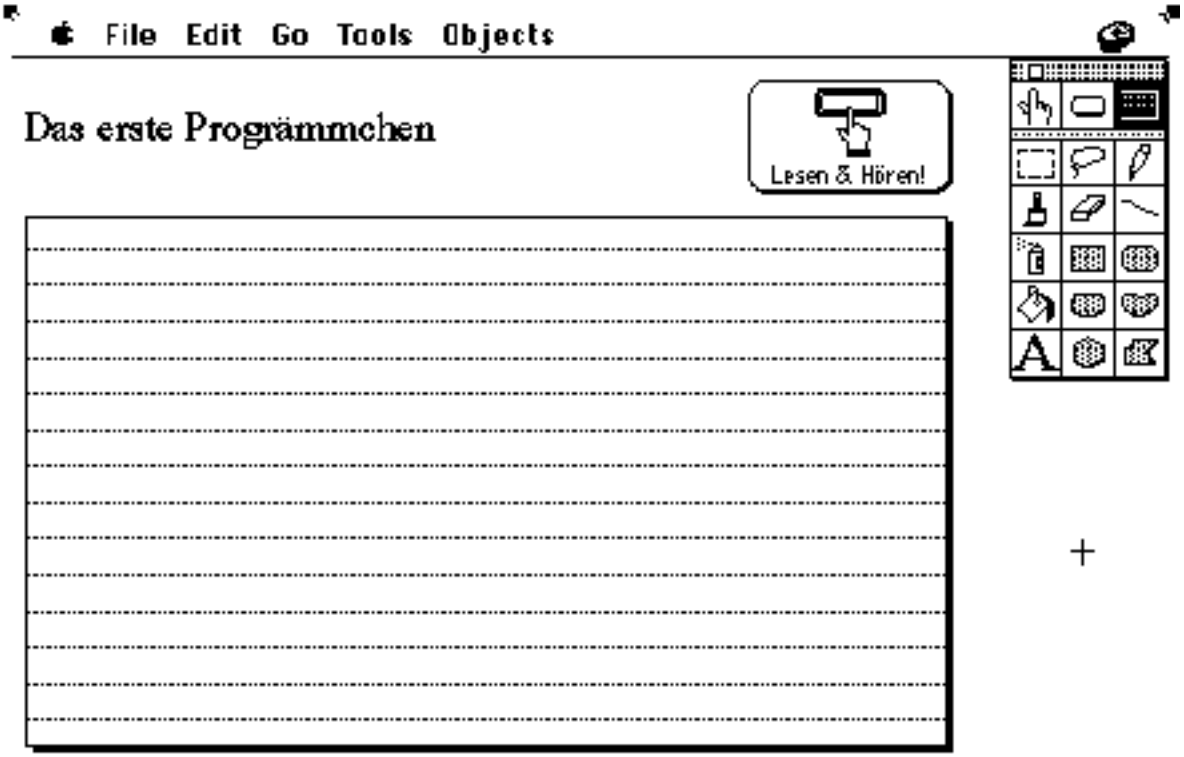
Deshalb sucht man Literatur, aber die Apple-Anleitung geht nicht weit genug; das Buch von Kitza (vgl. MUM 6/87) wurde sofort (zu Recht) gleich doppelt "verrissen" und das 720 Seiten starke 1,2kg schwere Buch von Danny Goodman möchte man auch nicht andauernd auf dem Schoß jonglieren, um wieder mal darin zu wühlen! Übrigens weiß dieser Profi, daß HyperTalk nicht ganz so einfach ist: Mindestens 25 mal rät er den Lesern "don't be afraid", womit er jedoch das Gegenteil bewirken und zunächst unbefangene Leser langsam aber sicher doch "afraid" machen könnte! (Damit kein falscher Eindruck entsteht: Das Buch ist *hervorragend* und jedem, der sich doch "echt" mit HyperTalk auseinandersetzen möchte, unbedingt zu empfehlen.)

Was möchte ich Ihnen nun anbieten? Antwort: Tips für Einsteiger *von einem Einsteiger.* Eben darum beginne ich diesen Beitrag *heute*, **an meinem dritten HyperCard-Tag**; denn jetzt weiß ich noch, was mich beschäftigt, *während* ich einsteige. Ich gehe davon aus, daß Sie sich bereits mit HyperCard vertraut gemacht haben; Sie haben den Info- und den Help-Stack angesehen; vielleicht auch schon einige Artikel gelesen, und jetzt wollen Sie ins **Scripting** einsteigen.

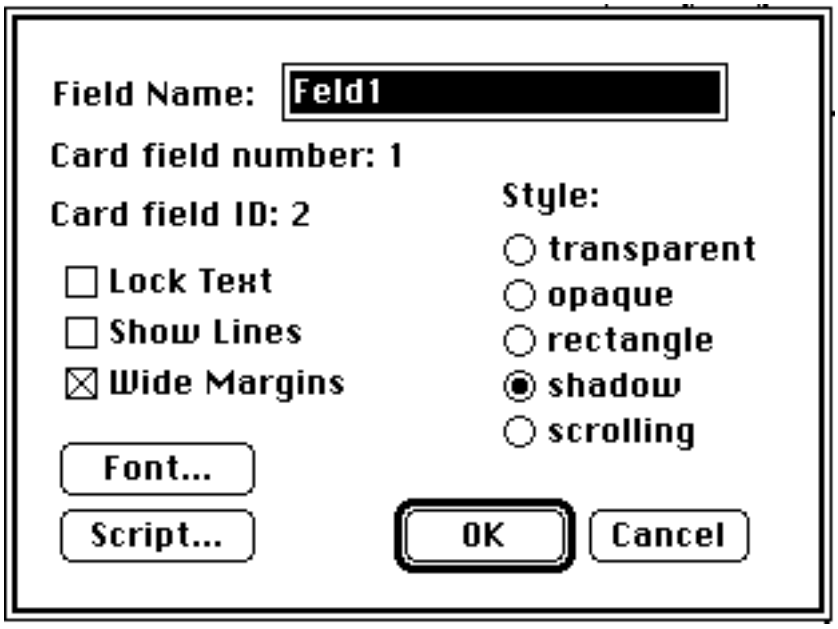
Das erste Prográmmchen

Das nachfolgende Mini-Programm soll einige erste aber wesentliche Aspekte von HyperTalk *demonstrieren.* Die Noten für die Musik entnahm ich Danny Goodman's Buch: "The Complete HyperCard Book" (S. 473). Schaffen Sie einen *New Stack* in welchem sich vorläufig *keinerlei* Background befindet. Dann gehen Sie auf die erste Karte und

erstellen ein Card Field, das in etwa so (Abb. 2) aussehen soll (wobei wir den Button anschließend kreieren werden):



Dieses Feld benennen und definieren Sie wie in Abb. 3, damit nachher alles in etwa so abläuft, wie bei mir, wobei ich als Font *Times* 18 gewählt habe):



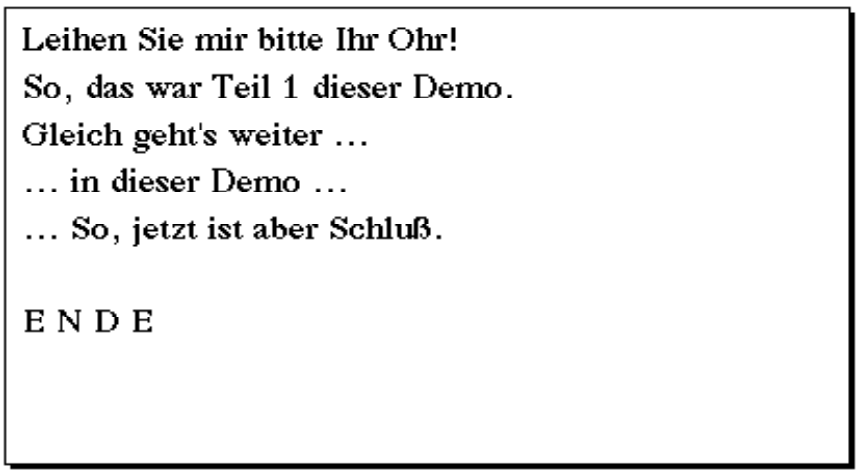
Wie Sie sehen ist **SHOW LINES nicht angekreuzt**, es **wirkt** nur in der Abbildung 3 so, weil alle Felder so dargestellt werden, solange das Field-Tool aktiviert ist!

Als nächstes klicken Sie auf SCRIPT und dann schreiben Sie bitte das folgende Programm für diese **Karte** (Abb. 4). Bitte beachten Sie, daß Zeilen mit zwei Bindestrichen Kommentare enthalten, die Sie selbstverständlich nicht eintippen müssen. Irgendwo habe ich gelesen, *ein* Bindestrich genüge auch, aber das stimmt nur bedingt! Manchmal ist es ok, manchmal aber kommt dann die Fehlermeldung, daß HyperCard den Befehl "-" nicht verstehen könne. Also doch lieber zwei Bindestriche!

So daß das Karten-Programm selbst aus nur drei Zeilen besteht; die erste: **on OpenCard**, die dritte mit **end OpenCard**. Bitte registrieren Sie auch, daß es bei HyperTalk vollkommen egal ist, ob Sie Groß- oder Kleinbuchstaben verwenden; aber längere Ausdrücke lesen sich besser, wenn man die hier vorgestellte Form verwendet.

Später könnten Sie das Text-Feld auch *transparent* machen, dann wird die Schrift bei Programm-Ablauf (nach Klicken des Buttons) wie *magisch* auf dem *blütenweißen* Background erscheinen; aber während des Schreibens und bei den ersten Probeläufen ist ein gut sichtbares Feld besser für den Überblick!!

So in etwa (Abb. 5) wird das Feld sich bei Programmablauf zeigen, wobei Sie bitte beachten wollen, daß die Schrift Zeile für Zeile **nacheinander** erscheint, **während** Musik ertönen wird!



Wir werden jetzt einen Button schaffen, der sowohl diese Zeilen als auch die erwähnte Musik produziert. Wie Sie den Button optisch gestalten wollen, ist egal, aber das Script hierzu sollte in etwa wie folgt sein, wobei Bemerkungen (wie gehabt) von zwei Bindestrichen eingeleitet werden):

```
-- Das erste Prograemmchen: Lesen & Hören: Button-Script
-- Vera F. Birkenbihl am 3. Tag mit HyperCard (6. Feb. 1989)
on mouseUp
  -- Als erstes lassen wir die MENUBAR verschwinden (Achtung:
  -- am Ende wieder einsetzen! - Siehe Ende des Programmes)
HIDE MENUBAR
  -- Jetzt löschen wir etwaigen Text (vom letzten Durchgang, wenn
  -- wir das Prograemmchen bereits ein- (mehr-)mals laufen ließen...
put " " into line 7 of card field "Feld1"
put " " into line 5 of card field "Feld1"
put " " into line 4 of card field "Feld1"
put " " into line 3 of card field "Feld1"
put " " into line 2 of card field "Feld1"
put " " into line 1 of card field "Feld1"
  -- Und jetzt erst lassen wir das bereits "saubere" Feld1
  -- auf der Card erscheinen!
show card field "Feld1"
beep -- falls Sie später Warn-Beeps in Ihre Stacks einbauen wollen...
wait 30 -- kurze Wartepause
  -- Es folgen die Befehle für Text im Textfeld plus "Musik"
put "Leihen Sie mir bitte Ihr Ohr!" into line 1 of card field "Feld1"
play "boing" tempo 130 "g# f f# a#"
  -- Achtung: Lange Strings (die Noten, oben) aus Gründen der Optik
  -- auf zwei Zeilen verteilt, damit die ganze Zeile am Bildschirm
  -- zu sehen ist; was die Arbeit erleichtert!
  -- Übrigens: Ohne Tempoangabe gilt (default) Tempo 200
play "boing" tempo 130 "b d#5t c#5e c c5# bb4e. be d5t d#e g#4"
play "boing" tempo 130 "bb4t be f gt g#e c#"
play "boing" tempo 130 "et fe f# at a#e c# et f#e a3# c4t c#e f#3w"
    repeat while the sound is not "done"
   -- die letzte Zeile bewirkt, daß die folgende Textzeile erst dann erscheint,
   -- wenn die Musik verklungen ist! Einer REPEAT-Klausel folgt immer
   -- ein END, s. nächste Programmzeile:
    end repeat
put "So, das war Teil 1 dieser Demo." into line 2 of card field "Feld1"
  --  Im folgenden können Sie den Musikteil von oben KOPIEREN, sofern Sie nachträglich
  --  das Instrument und das Tempo verändern.
play "harpsichord" tempo 180 "g# f f# a# b d#5t c#5e c c5# bb4e."
play "harpsichord" tempo 180 "be d5t d#e g#4"
play "harpsichord" tempo 180 "bb4t be f gt g#e c#"
play "harpsichord" "et fe f# at a#e c# et f#e a3# c4t c#e f#3w"
```

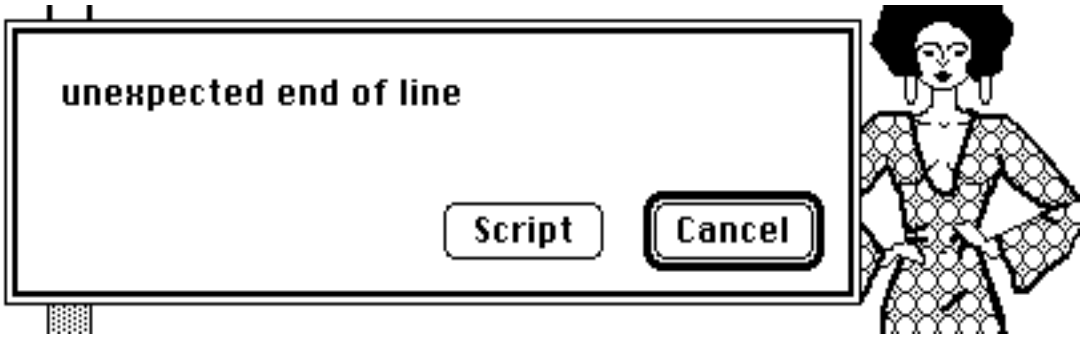
```
    repeat while the sound is not "done"
    end repeat
put "Gleich geht's weiter ..." into line 3 of card field "Feld1"
  -- Nochmal der Musikteil, diesmal wieder "Boing" und neues Tempo:
play "boing" tempo 260 "g# f f# a# b d#5t c#5e c c5# bb4e."
play "boing" tempo 260 "be d5t d#e g#4"
play "boing" tempo 260 "bb4t be f gt g#e c#"
play "boing" tempo 260 "et fe f# at a#e c# et f#e a3# c4t c#e f#3w"
    repeat while the sound is not "done"
    end repeat
put "... in dieser Demo ..." into line 4 of card field "Feld1"
  -- Und nochmal die Musik; wieder "Harpsichord" und Tempo diesmal bei 380:
play "harpsichord" tempo 380 "g# f f# a# b d#5t c#5e c c5# bb4e."
play "harpsichord" tempo 380 "be d5t d#e g#4"
play "harpsichord" tempo 380 "bb4t be f gt g#e c#"
play "harpsichord" "et fe f# at a#e c# et f#e a3# c4t c#e f#3w"
    repeat while the sound is not "done"
    end repeat
put "... So, jetzt ist aber Schluß." into line 5 of card field "Feld1"
  --
put "E N D E" into line 7 of card field "Feld1"
  -- Als vorletzte Aktion "räumen" wir das Feld wieder weg, damit wir
  -- die Ausgangsposition wieder herstellen; aber zuerst brauchen wir
  -- eine kleine Warteschleife, damit der Anwender die letzten beiden
  -- Textzeilen in Ruhe lesen kann:
Wait 150
HIDE CARD FIELD "Feld1"
  -- Natürlich könnte man dem Feld auch einen ok-Button zuordnen, den
  -- der Anwender klickt, wenn er will, daß das Feld verschwindet...
  -- Jetzt muß nur noch die MENUBAR wieder eingesetzt werden
SHOW MENUBAR
end mouseUp
```

Einige Bemerkungen zu dem Listing:

1. Sinn und Zweck von **Kommentaren**: Es ist eine der größten **Gefahren, daß Programmierer zu wenig kommentieren/dokumentieren!** Diese Bemerkungen im *Script* sind aber sehr hilfreich, erstens für einen anderen Menschen, der Ihr Script liest, und zweitens für Sie selbst.
   Wenn Sie nämlich Monate (oder Jahre) später in alte Programme hineinsehen, dann wissen Sie oft selbst nicht mehr, ob es von Ihnen selbst oder "geklaut" war (ein Copyright-Hinweis wäre hilfreich) und oft wissen Sie später nicht mehr, *warum* Sie (damals) so und nicht anders vorgegangen sind.
   So könnte man z.B. als Bemerkung in obiges Programm (falls Sie es wirklich eintippen) noch eintragen, daß es nur in Verbindung mit dem Feld (und dem Script dieser Karte!) funktioniert. Sollten Sie dann später einmal den Button an eine an-

dere Stelle kopieren wollen, dann wissen Sie auch dann sofort (wieder), was zu diesem Button noch "dazugehört", weil Sie die Kommentare lesen können. Deshalb lohnen sie sich.

2. Wenn wir sicher gehen wollen, daß unser **TextFeld** zu Beginn unserer kleinen Vorführung **leer** ist, dann müssen wir es zunächst löschen. Das Löschen der Zeilen wird bewirkt durch das Setzen (PUT) von einer "Botschaft" pro Zeile, wenn die Botschaft aus *Leerzeichen* besteht. Zuerst meinte ich, man müsse mindestens so viele Leerzeichen setzen, wie Text "überdeckt" werden soll, aber das ist nicht notwendig; ein Leerzeichen genügt. Denn bei PUT plus minimum 1 Zeichen wird die ganze Zeile "ersetzt"; was in praxis natürlich bedeutet: gelöscht.
3. Das Programm **wirkt** nur beim ersten Hinsehen **lang**, denn erstens enthält es eine **Menge Kommentare** die Ihrem Verständnis dienen, **aber nicht eingegeben werden müssen**, und zweitens gibt es eine **Reihe von identischen Befehlszeilen**, bei denen sich natürlich KOPIEREN und EINSETZEN anbietet. Bitte beachten Sie, *daß beim Schreiben der Scripten keine MENU-Befehle via Maus möglich sind.* Also nur über die Tasten KOPIEREN (Befehl plus **c**), SCHNEIDEN (Befehl + **x**) und EINSETZEN (Befehl + **v**) wenn Sie **scripten**. (Übrigens werden GanzseitenBildschirm-Besitzer sich freuen, denn das Script-Fenster geht über die gesamte Höhe des Bildschirms, was bei langen Scripts natürlich super ist!)
4. Achtung: **Normalerweise** gilt die Regel: *Nur ein Befehl pro Zeile*, gefolgt von dem Signal, das dem Computer zeigt, daß ein Befehl beendet ist: **RETURN**. Aber manche Befehlszeilen können sehr lang werden; dann verschwinden sie rechts aus dem Blickfeld, weil das Script-Window leider nicht seitlich scrollen kann! Deshalb gilt: Lange Programmzeilen mittels **Soft-RETURN** (Wahl + RETURN) unterteilen. Lange Zeichenketten (neudeutsch "Strings", wie z.B. die Noten in den PLAY-Befehlen) dürfen allerdings leider nicht auf diese Weise optisch getrennt werden. Dies führt zu *Fehlermeldungen*. Daher werden lange Strings am besten in mehrere Befehle unterteilt, wie in unserem Beispiel auch!
6. Apropos **Fehlermeldung**: Ich habe mir angewöhnt, ein Programm, das ich laufend revidieren möchte, bewußt mit einem Programmierfehler zu beenden (z.B. zwei xx vor dem letzten: *end MouseUp*). Dann erscheint die Fehlermeldung (Abb.6), ich klicke auf *Script* und bin schon wieder mitten im Programm. Übrigens können Sie jederzeit sehr schnell ans Script eines Objekts gelangen, wenn Sie mit dem diesbezüglichen Tool (z.B. Button-Tool) *mit Shift-Taste* auf das Objekt *doppelklicken*! Diese Fehlermeldungs-Methode ist bei Probeläufen jedoch schneller, weil Sie für den Test-Lauf des Buttons ja zwangsläufig als Tool das *Händchen* (zum Klicken des Buttons) aktiviert hatten.



7. Beachten Sie bitte auch, daß die Musik erst langsam, bei jedem Durchgang jedoch schneller gespielt wird. Ohne Tempoangabe gilt (der Default-Wert von) **Tempo 200.** Übrigens wird der PLAY-Befehl im Apple-Handbuch zu HyperCard überhaupt nicht erwähnt; ich kann ihn aus Platzgründen hier (heute) nicht detailliert beschreiben; aber in den erwähnten Büchern (von Kitza, Goodman) ist er jeweils drin.
8. Nach der **Musik** erscheint eine **REPEAT-Klausel**, die bewirkt, daß die folgende Textzeile (in Feld1) erst erscheinen wird, nachdem die Musik verklungen ist! Ohne diese Klausel erscheinen nämlich zuerst alle Textzeilen während die Musik "hinterherhinkt". Sie sehen also, daß Ihre Befehls-Reihenfolge zwar die **logische Abfolge** bestimmt, **nicht** aber unbedingt die **zeitliche**. Wenn das Spielen der Musik länger dauert, wird eben der nächste Befehl (nächste Zeile erscheinen lassen) bereits befolgt. Wann immer Sie dies vermeiden wollen, hilft Ihnen diese **REPEAT-Klausel**: Nun muß das Programm warten, bis der Teil vor der Klausel abgearbeitet ist, ehe es die folgende Befehle ausführen darf!
9. Das Prograämmchen enthält einige **Standard-Konstruktionen**, die Sie immer wieder finden werden, (z.B. REPEAT ..., end REPEAT). Wer schon einmal programmiert hat, erkennt das Prinzip sofort wieder; **andernfalls nehmen Sie es anfangs bitte lediglich zur Kenntnis!! Wenn Sie Programme anderer immer wieder analysieren (und Teile davon kopieren), entwickeln Sie für die Syntax (d.h. die "Grammatik") solcher Konstruktionen bald ein Gefühl.** Das ist beim Erlernen einer **Fremdsprache** genauso. Wer eine Form mehrmals in praxi verwendet hat, wird bald **intuitiv ahnen**, wann sie wieder "paßt". **Diese Art zu lernen ist weit angenehmer, als der krampfhafte Versuch, Regeln zu pauken.** Möchten Sie dieses Prizip auf Fremdsprachen übertragen? Mein Buch hierzu gibt's in jedem Buchladen: ("Die Birkenbihl-Methode, Fremdsprachen zu lernen - gehirn-gerecht, ohne Vokabelpauken!; es geht übrigens demnächst in die 3. Auflage und wurde von mir (Anfang des Jahres 1986!!) mit WriteNow selbst "gesetzt", inkl. zahlreicher Abbildungen. Also Pseudo-DTP – aber mit einfachsten Mitteln! (**P.S. 2006:** Inzwischen gibt es auch eine gleichnamige **DVD** mit Vortrag, Diskussion und 70 Minuten Bonus-Material und das Büchlein selbst ist inzwischen auch als **Taschenbuch** erhältlich und dürfte mindestens die 15. oder 20. Auflge erreicht haben. Ich weiß bei den vielen Büchern nie auswendig, welche Auflage gerade „dran ist")
10. Die **Musik**, die das Prograämmchen erzeugt, klingt ohne **Lautsprecher** zwar "dünn" und leise, aber **das Prinzip kann man bereits mit dem "nackten Mac" hören**.
11. Sie wissen ja, daß Sie über die **MESSAGE-Box** jederzeit Befehle an das HyperCard-System geben können, z.B. SHOW CARD FIELD 1 (oder Rechenoperationen, vgl. Apple Handbuch zu HyperCard). **Diese MESSAGE-Box wird beim Schreiben von Scripten sehr wichtig.**

    Angenommen, Sie haben den Befehl HIDE (verstecke) irgend etwas (z.B. einen Button, ein Feld) in das Script eines Buttons eingebaut und testen Ihr Programm. Nun ist das Objekt aber weg (hidden!). Mit einem Befehl in der MESSAGE-Box holen Sie es nun wieder hervor und planen dann die intelligente Lösung dieses

Problems. So ging es mir, nachdem ich die MENUBAR am Anfang des Programms großartigerweise "versteckt" hatte und sie am Ende auch brav weggeblieben war.

Sie Sehen: Durch solche und ähnliche "**Pannen**" (die zum Programmieren gehören wie die Luft zum Atmen!) wird einem **schnell und schmerzlos** klar, was man zunächst **vergessen** hatte. Diese Art zu arbeiten (Script + Testlauf + Scriptfortsetzung) macht das Erlernen von HyperTalk ja so unglaublich angenehm! Wer einst BASIC oder eine andere Sprache gelernt hat, bei der man sofort testen konnte ohne erst das ganze Programm zu Kompilieren (wie Pascal), weiß das zu schätzen!

Ob Sie dieses Button-Programm **nur gelesen haben** oder es tatsächlich ausprobieren werden; ich wünsche Ihnen dabei **viel Entdeckerfreude!**

P.S. Das #-Zeichen in den Musik-Noten wird durch Wahl + Shift + 3 erzeugt.