(Sanke 2004)

Einführung in die Programmierung mit Hypercarddialekten

Überblick: Gemeinsamkeiten von Hypercarddialekten (X-Talk-Sprachen)

1. Hypercard (Mac, Claris)
2. HyperPad (DOS, IQ-Technologies)
3. Plus (Mac + Windows, ObjectPlus)
4. Toolbook (Windows, Asymetrix)
5. Supercard (Mac, Allegiant)
6. Oracle Media Objects (Windows + Mac, Oracle)
7. Metacard (Windows + Mac + Unix + Linux, Metacard)
8. Revolution (Windows + Mac + Unix + Linux, Runtime Revolution)

Von den hier aufgeführten Autorensystemen sind inzwischen HyperPad, Plus und Oracle Media Objects vom Markt verschwunden. Hypercard ist seit Jahren nicht weiterentwickelt worden und beschränkt sich wie Supercard auf die Mac-Umgebung. Toolbook ist nur unter Windows benutzbar und hat eine Preisebene von mehreren 1000 Euro erreicht, so dass dieses Programm für schulische Einrichtungen kaum mehr in Frage kommt.
Metacard läuft unter den meisten verfügbaren Betriebssystemen, wobei die erzeugten Dateien leicht von einer Plattform auf die andere portiert werden kännen und liegt in einer kostenlosen "Starter Kit"-Version vor. Die einzige Einschränkung dieser Einstiegsversion ist eine Begrenzung der Programmzeilen im Skript der Objekte, die jedoch sogar überwunden werden kann.

Revolution ist eine Weiterentwicklung von Metacard, die eine Reihe neuer Funktionen enthält, jedoch im gegenwärtigen Entwicklungszustand noch mit vielen „bugs“ behaftet ist, die das Arbeiten mit Revolution empfindlich beeinträchtigen. Ebenso ist bei Revolution keine kostenlose "Starter Kit"-Version mehr vorhanden, sondern nur eine auf 30 Tage beschränkte Trial Version.
Revolution hat 2003 die „Metacard engine“, d.h. das Kernstück von Metacard, aufgekauft. Die Kaufvereinbarung sieht jedoch vor, dass Metacard-Nutzer die Metacard engine unter den bisherigen Bedingungen weiter verwenden können.
Neue Interessenten an Metacard können von Runtime Revolution (www.runrev.com) beim Erwerb einer Lizenz auf Wunsch gleichzeitig eine kostenlose Metacard-Lizenz erhalten.

Die Weiterentwicklung der Metacard IDE wird von einer Gruppe von erfahrenen Metacard-Nutzern betrieben, die unter der Email-Adresse metacard@lists.runrev.com angesprochen werden kann.

Metacard ist wesentlich leichter zu erlernen als beispielsweise Macromedia "Director" und bietet trotzdem eine Funktionalität, die der der C-Sprachen nahe kommt.
Metacard ist daher ein für schulische Einrichtungen geeignetes Werkzeug zur Entwicklung von Multimediaanwendungen. Metacard ist kein Programm zur Entwicklung von Webseiten, kann aber auf Webseiten zugreifen oder sie aufrufen (oder kann Emails verschicken) und erlaubt es, Metacardprogramme online aufzurufen und sie dann on- oder offline zu verwenden; andererseits kann Metacard sogar anstelle von cgi-Skripten oder Perl zur Programmierung von Internet-Servern verwendet werden.
Die gesamte Programmierumgebung von Metacard umfasst zip-komprimiert nur 1,4 MB; sie benötigt keinerlei DLLs und kann deshalb ohne spezielle Installation von jedem Verzeichnis und jedem Datenträger gestartet werden.-

Die gemeinsamen Merkmale der X-Talk-Sprachen sind:

Integrierte Entwicklungsumgebung (IDE)

Direkte visuelle Gestaltung der Oberfläche (visual design)

"Objekte" als Programmelemente:

a) Dateien/Files: stacks, books, pads, projects

b) Seiten: pages, cards - bestehend aus den Teilelementen

1. backgrounds (Hintergrund)
2. page, card (Seite)
3. fields (Felder für Text, Graphik oder "Mal"-Objekte)
4. buttons (Schaltflächen, Tasten)
5. images (Bilder)
6. player, stage (für Video- und Sound-Dateien)
7. graphics (intern erstellte Graphiken)

"Objektorientierte" Programmierung,

d.h. keine lineare bzw. sequentielle Programmierung in einem File, sondern Verteilung des Programmcodes auf die Scripte der einzelnen Objekte (s. III.)

Natürlichsprachige Programmiersprache,

die den Aufwand an Kommentierung drastisch vermindert.

Vordefinierte Programmelemente zur Sprachmanipulation,

auf die unmittelbar zugegriffen werden kann:

a) characters (chars)
b) words
c) items
d) lines

Herstellung von Hypertext

mit Verknüpfungen über "go to" oder "linktext" und "clicktext"

Leichter Einstieg in die Programmierung

mit elementarem Befehlssatz von 10 - 20 Befehlen

Globale und lokale Variablen ohne weitere Typdefinition möglich

und ohne vorherige Deklaration (+ Spezial-Variable "it")

Voller Befehlssatz mit Kontrollstrukturen

wie bei anderen Programmiersprachen (C, Delphi, Pascal, Basicdialekte etc.)

Ergänzung der Sprachen
durch eigene Prozeduren und externe Befehle möglich

I. Objekte

Die 8 Hypercarddialekte (**Hypercard** selbst, **Supercard**, **Toolbook**, **WinPlus, HyperPad, Oracle Media Objects** und **Metacard/Revolution**) haben als gleichartige Elemente (-> **Objekte**):

1. **Dateien** ("stacks" in Hypercard und Metacard und "books" in Toolbook), die aus mehreren **Seiten** bzw. aus mindestens einer Seite bestehen.

2. **Seiten** mit einem **Hintergrund** (background), auf denen **Felder** und **Buttons** sowie **Graphikobjekte** angebracht werden.
   In Hypercard und Metacard werden die Seiten als "cards", in Toolbook als "pages" bezeichnet.

3. **Hintergrund** (background): Auf dem Hintergrund einer oder zumeist mehrerer Seiten mit einem gemeinsamen Hintergrund werden Objekte, d.h. Felder und Buttons angebracht, die auf allen zugehörigen Seiten benötigt werden.

4. **Felder** (fields), die der Aufnahme von Text dienen. Innerhalb der Felder und zum Textausstausch zwischen den Feldern kann wie mit einer einfachen Textverarbeitung gearbeitet werden, d.h. es stehen neben einer automatischem "wordwrap"-Funktion "cut", "copy" und "paste" zur Verfügung. Ebenso können in allen Dialekten (mit Ausnahme von HyperPad) in einem Feld unterschiedliche Schrifttypen, Größen, Farben etc. zur Strukturierung und Hervorhebung verwendet werden.

5. **Buttons** („Tastern“ oder "Schaltflächen" - z.B. in der deutschen Version von Toolbook -), die als primäre Schaltelemente den Hauptteil des Programmcodes aufnehmen.

6. **Images**, d.h. importierte Bilder oder Zeichnungen.

7. **Graphics**, intern mit Zeichenwerkzeugen erstellte Graphiken

8. Toolbook, Metaard und Revolution haben ein zusätzliches **Player**- oder **Stage**-Objekt für Video- und Sound-Dateien (Quicktime, MPEG, AVI, Flash)

9. Metacard und Revolution haben separate **Scrollbars**, die u.a. zur Fortschrittsanzeige verwendet werden können.

II. Adressen

Alle diese Objekte können generell auf verschiedene Weise angesprochen werden:

a) mit ihrem **Namen**, der ihnen vom Programmentwickler zugeordnet wird:

card "Einführung", field "Lexikon", button "Start" etc., wobei die Namen in Anführungszeichen zu setzen sind.

b) mit einer automatisch zugeteilten **ID-Nummer**, die für das einzelne Objekt unverändert bleibt, auch wenn z.b. ein Objekt mit einer niedrigeren ID-Nummer gelöscht wird (z.B. button ID 5 von insgesamt 10 Buttons):

card ID 1234, field ID 1006, button ID 1004, group ID 1103.

c) mit einer **laufenden Nummer** für das einzelne Objekt, die sich automatisch ändert, wenn Objekte, z.B. Seiten (cards, pages), zwischen anderen Objekten eingefügt oder gelöscht werden:

card 1, button 10, field 4.

Die Objektadressen a - c werden kombiniert, wenn z.B. von einer Seite auf ein Objekt einer anderen Seite zugegriffen werden soll:

button "Aufgabe" of card ID 1016 of stack "Grammatik"
field 5 of card "Lexikon"
field ID1005 of card "Auswertung" etc.

III. Programmiersprache

Jedes der oben unter I. beschriebenen **Objekte** kann mit einem **Skript** zur Steuerung des Programmablaufs versehen werden, d.h. der Programmcode wird mit den einzelnen Objekten verknüpft bzw. auf sie verteilt. Die einzelnen Objekte rufen sich ggf. durch ihre Skripte gegenseitig auf oder warten auf Nutzereingaben z.B. durch Mausklick oder Texteingaben.

Die Programmiersprachen der Hypercarddialekte entsprechen weitgehend einem in Syntax und Vokabular vereinfachten Englisch, so daß man streckenweise ohne Kommentierung des Programmcodes auskommen kann. Beispiele:

put first word of card field "Lexikon" into line 2 of card field "Aufgabe" of card "Übung"

Bei **Toolbook** muß bei allen Textmanipulationen (notwendigerweise, aber im Entwurf der Programmiersprache logisch nicht erforderlich) das Wort "text" hinzugefügt werden; die obige **Hypercard**zeile lautet danach in Toolbook:

put the first word of the text of field "Lexikon" into textline 2 of the text of field "Aufgabe" of page "Übung".

(Der Artikel "the" ist optional und dient nur der besseren Lesbarkeit).

Weitere Beispiele in der Sprache von **Metacard**:

```
put field "Eingabe" into Eingabe   (d.h. plaziere den Text des Feldes "Eingabe" - der
                                     eingetippt wurde - in die Variable Eingabe)
if Eingabe is Loesung then         (falls der Inhalt der Variable Eingabe gleich der
                                     Variable Loesung ist, dann ...)
show field "richtig"               (zeige das Feld "richtig")
wait 2 seconds                      (warte 2 Sekunden)
hide field "Richtig"                (verstecke das Feld "richtig")
```

put the number of words of field "Lexikon" into Anzahl
(zählt die Anzahl der Wörter im Feld und plaziert sie in die Variable Anzahl)

go to next card (schaltet zur nächsten Seite).

Soweit einige Beispiele, die innerhalb eines Skripts zeilenweise kombiniert werden können.

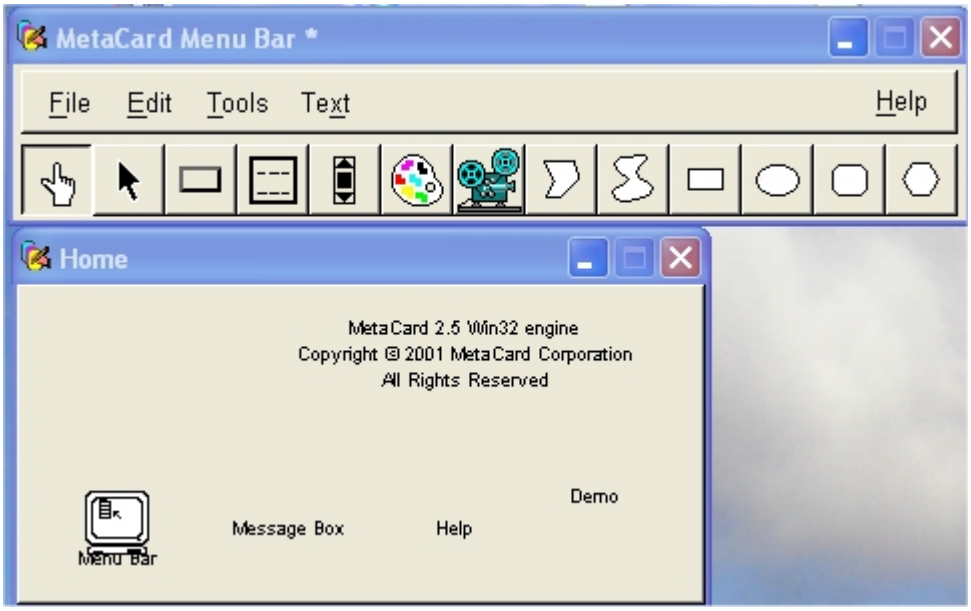
Die Skripte der Objekte werden durch Anwenderaktionen, d.h. durch Mausklicks oder Eingaben über die Tastatur oder indirekt von den Skripten anderer Objekte aus ausgelöst.

Die grundlegende Aktion ist der **Mausklick auf einen Button**, der in den Skriptsprachen von **Hypercard** und **Metacard** im Skript des Buttons mit "on mouseup" abgefangen wird. Ein Button, der zur nächsten Seite weiterschaltet, muß folgendes Skript enthalten:

"**on mouseup**
go to next card
end mouseup"

"on mouseup" und "end mouseup" sind in jedem Hypercard- und Metacardbutton bereits als Skripttext dieser Grundfunktion enthalten.

IV. Übungen in Metacard 2.5



Zunächst muß Metacard mit der Datei "mc.exe" (unter Windows) bzw. "Metacard" (auf der Mac-Oberfläche) gestartet werden, falls es nicht über das Icon einer Programmgruppe auf der Windows-Oberfläche zugänglich ist. Es erscheint der sogenannte "Homestack", auf dem das Icon "Menu Bar" angeklickt werden muß, so daß die Werkzeugleiste „Metacard Menu Bar“ erscheint.

Nach Anklicken des Menüpunkts "File" den Unterpunkt "New Stack" wählen, d.h. Anklicken.
Es erscheint der "Stack Properties"-Dialog, in den für "Stack Title" und "Stack Name" Bezeichnungen eingegeben werden müssen und die Größe des Stacks auf dem Bildschirm in den Feldern "Width" (Breite) und "Height" (Höhe) zu bestimmt werden kann ist; zum sinnvollen Arbeiten wählen Sie etwa eine Größe von 400 zu 350. Die Größe des Stacks lässt sich jedoch einfacher mit der Maus verändern: Die Maus an den Rand des Stacks bewegen bis der Doppelpfeil erscheint, dann mit gedrückter Maustaste den Stack auf die gewünschte Größe ziehen.

Danach schließen Sie den "Stack Properties"-Dialog mit einem Klick auf das Kreuz in der oberen Leiste des Dialogfeldes (d.h. das unter Windows übliche Symbol für das Schließen von Fenstern).

Grundlegende Operationen für Metacard 2.5:

Erstellen eines Textfeldes

In der **Werkzeugpalette** das **vierte Symbol von links aus**  (d.h. das "field tool") anklicken - es erscheint ein Kreuz, wenn man die Maus auf die Fläche des Stacks bewegt - und mit Drücken der linken Maustaste ein **Feld** beliebiger Größe aufziehen.

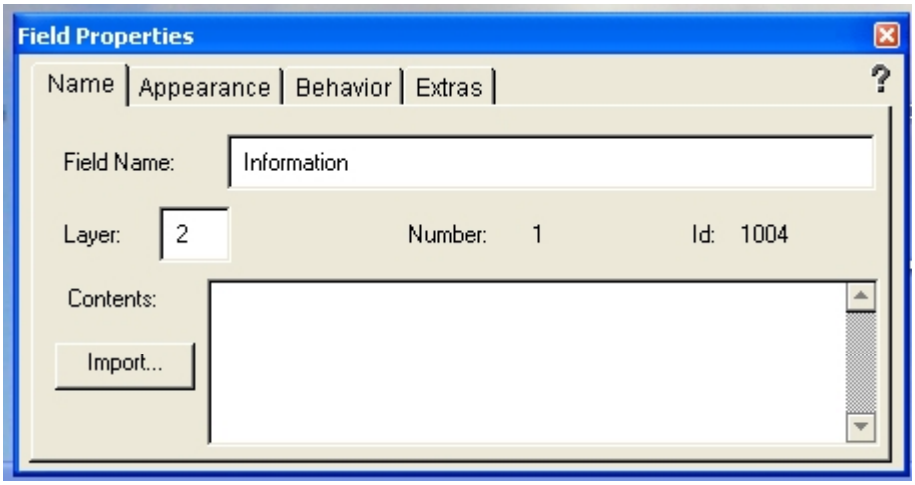
Wenn jetzt das Feld angeklickt wird, wird es als **Objekt aktiviert:** als **Zeichen der Aktivierung** erscheinen **8 kleine Quadrate am Feldrand**, mit denen die Größe verändert werden kann (mit der linken

gedrückten Maustaste die Quadrate ziehen). Klickt man - wenn das Feld aktiviert ist - **in** das Feld und hält die linke Maustaste gedrückt, kann man es an eine beliebige Position auf dem Bildschirm ziehen.

Jede Änderung der Eigenschaften eines Feldes - wie auch aller anderen Objekte, wie Buttons und Graphiken - setzt die vorherige Aktivierung im "select"-Modus voraus. Dazu muß in der Werkzeugleiste, der "Metacard Menu Bar" der Pfeil  (2. Symbol von links) angeklickt werden; danach das zu aktivierende Objekt anklicken, so daß die 8 kleinen Quadrate am Rand des Objektes erscheinen. Diesem "select"-Modus entspricht der "Autorenmodus" in Toolbook, in dem Änderungen der Objekte vorgenommen und die Skripte der Objekte "programmiert" werden können.

Mit Klick auf das **Handsymbol** in der Werkzeugleiste wird in den "browse"-Modus umgeschaltet, in dem das Programm nach Änderungen oder Fertigstellung dann abläuft und ausprobiert werden kann.
Für erneute Änderungen wiederum zunächst auf den **Pfeil** "select tools" klicken, danach mit Klick auf das zu verändernde Objekt, z.B. ein Feld, dieses **aktivieren**.

Die Objekteigenschaften - **object properties** - können jetzt auf **zweierlei** Art bestimmt werden:

1. Im Menüpunkt **Edit** der Werkzeugleiste und den Unterpunkt **Object Properties** anklicken. Im Falle eines aktivierten Feldes erscheint jetzt das "**Field Properties"** -Dialogfenster mit den Teilbereichen **name**, **appearance**, **behavior** und **extras**, in denen verschiedene Eigenschaften des Feldes eingestellt werden können.
2. 

2. Anklicken des aktivierten Objektes mit der **rechten Maustaste.** Es erscheint ein kleines "pull-down-Menü" mit den beiden ersten Unterpunkten "script" und "properties". Das Anklicke von "properties" öffnet jetzt - ähnlich wie oben, aber auf kürzerem Wege - das "field properties"-Dialogfenster.

Geben Sie dem Feld einen **Namen**, unter dem es später im Buttonskript o.ä. angesprochen werden kann. Schließen Sie das Field-Property-Dialogfenster mit Mausklick auf das Kreuzsymbol rechts oben am Fensterrand.

Einfügen von Text in Felder

2 Möglichkeiten sind gegeben:

1. Das Handsymbol in der Werkzeugleiste zum Umschalten in den "browse"-Modus anklicken, danach mit der Maus in das Feld klicken, so daß die blinkende Schreibmarke erscheint. Jetzt kann wie bei einer normalen Textverarbeitung Text eingegeben und redigiert werden. Über das Menüpunkt "Edit" in der Werkzeugleiste stehen die Funktionen cut, copy und paste zur Verfügung. (Für "paste" in Feldern bitte

die Tastenkombination "CTRL-V", auf deutscher Tastatur "Strg-V" verwenden.)

2. Im "field-properties"-Dialogfenster findet sich auf der Unterkarte "name" der Button "Import", über den Textdateien in ein Feld importiert werden können.

Textattribute wie Font ,Schriftgröße, Stil und Farbe können über die Menüpunkte der Werkzeugleiste (Metacard Menu Bar) geändert werden oder können als Grundeinstellung für ein gesamtes Feld in der Unterkarte "appearance" des "field-properties"-Dialogfensters festgelegt werden.

Erstellen eines Buttons (in neuerer Übersetzung auch Schaltfläche genannt)

In der **Werkzeugpalette** das **dritte Symbol von links**  anklicken - es erscheint ein Kreuz, wenn man die Maus auf die Arbeitsfläche bewegt -und mit Drücken der linken Maustaste einen **Button** beliebiger Größe aufziehen.

Wenn jetzt der Button angeklickt wird, wird er als **Objekt aktiviert** und es erscheinen - ebenso wie beim Feld - 8 kleine Quadrate am Buttonrand, mit denen die Größe verändert werden kann (mit der linken Maustaste die Quadrate ziehen)..

Wie oben für Felder beschrieben entweder über den Menüpunkt **Edit** in der Werkzeugleiste **Object Properties** aktivieren (anklicken) oder den aktivierten Button direkt mit der rechten Maustaste anklicken und das "properties"-Dialogfenster aufrufen, in dem jetzt die verschiedenen Eigenschaften des Buttons eingestellt werdenkönnen. Geben Sie dem Button auf der Unterkarte **name** einen Namen, d.h. eine Beschriftung, die auf dem Button sichtbar ist. Auf der Unterkarte **behavior** der **Button Properties** gelangenn Sie auch in den **Skript-Editor**. Schließen Sie die Button-Property-Dialogbox mit einem Klick auf das Kreuz oben recht im Button-Dialogfenster .

==========

Weitere Übungen in Metacard

A. das Aufblenden und Abblenden von Informationsfenstern auf einer Seite

B. die Verknüpfung von Seiten

.

A: Auf- und Abblenden von Feldern

Ü**bung A1:**

Erstellen Sie ein Feld auf Ihrer 1. Metacardseite, das Sie im **Field Properties** -Dialogfenster mit einem **Namen** versehen.

Geben Sie einen Text in das Feld ein.

Plazieren Sie darunter 2 Buttons, denen Sie die **Beschriftung** "hide" und "show" o.ä. geben (über das "**Button Properties"-Dialogfenster**). Mit diesen Buttons soll das Textfeld ein- und ausgeblendet werden können.

Die Buttons benötigen dazu folgende Skripte (der Skript-Editor ist ebenfalls über **Button Properties** zugänglich):

Button 1:

```
on mouseup
        show field "Name"
end mouseup
```

"mouseup" wird als ein Wort geschrieben!

Mit "**Name**" ist die Bezeichnung gemeint, die Sie selbst dem Feld gegeben haben; sie muß in **Anführungszeichen** stehen.

Schließen Sie den **Skripteditor** durch Anklicken des Buttons "OK" am unteren Rand des Skripteditors.

Das Skript für den "hide"-Button lautet fast identisch:

```
on mouseup
        hide field "Name"
end mouseup
```

Klicken Sie auf das Handsymbol in der Werkzeugleiste, um in den"**Browse"-Modus** umzuschalten, und probieren Sie Ihr Programm durch abwechselndes Klicken auf die beiden Buttons aus.

B: Verknüpfung von Seiten

Übung B1

Fügen Sie Ihrem **Stack** (d.h. ihrer geöffneten Metacard-Datei) eine oder mehrere Seiten hinzu -> Unterpunkt **Create Card** im Menü **Edit**. Geben Sie diesen Seiten (über den **Properties Dialog**) einen Namen und erstellen sie ein Textfeld auf jeder Seite.

Plazieren Sie auf jeder dieser Seiten einen Button oder wenn Sie mehrere Seiten hinzugefügt haben, auch ggf. weitere Buttons, mit denen zu den anderen Seiten weitergeschaltet werden kann.

Die Buttons erhalten jeweils folgendes Skript:

```
mouseup
        go to card "Name"
end mouseup
```

"Name" meint den Namen der jeweiligen Seite.
Geben Sie in **Button Properties** dem Button als **Label** den Namen/die Bezeichnung der Seite, zu dem er schalten soll.

Klicken Sie auf den Pfeil in der MenuBar (zum Auswählen des **Browsemodus**) und probieren Ihre Hyperlinks zwischen den Seiten aus.

Übung B2 : Buttons auf dem Background platzieren

Beliebige Objekte können als "**group**", d.h. als zusammengehörig bestimmt werden, wobei ggf. ein gemeinsames Skript für alle Objekte einer Gruppe hinzugefügt werden kann.

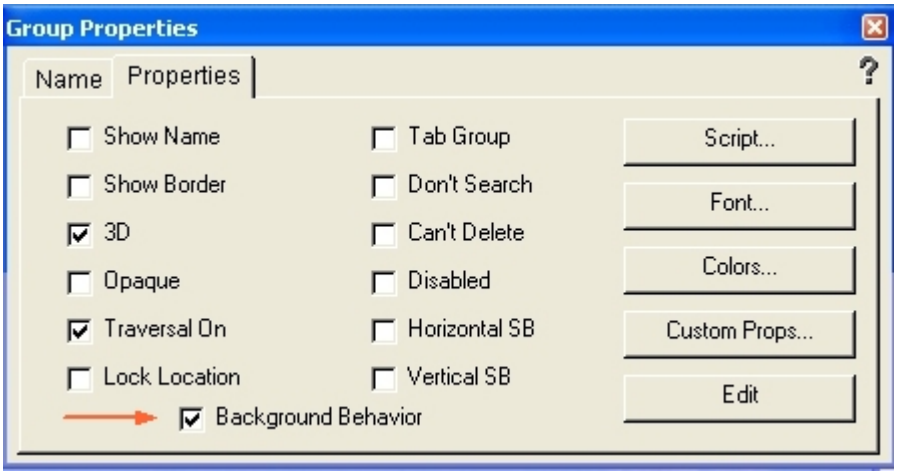
Das normale Verfahren zur Bildung einer Gruppe ist wie folgt:

Klicken Sie - bei gedrückter Shift-Taste - alle Objekte, die zur "**group**" gehören sollen, nacheinander an.

Sie werden dadurch ausgewählt, was durch die bei jedem Objekt erscheinenden 9 Randpunkte sichtbar wird.

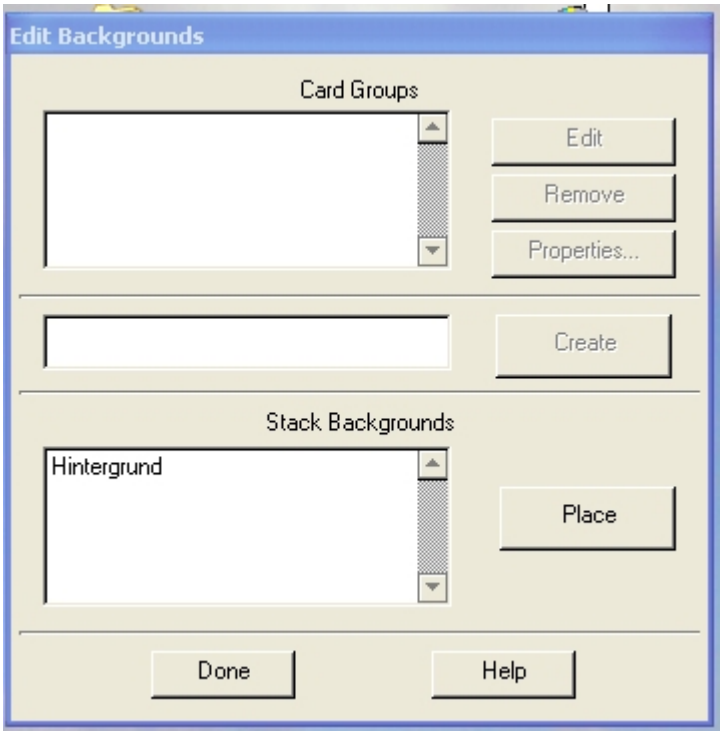
Wählen Sie dann im Edit-Menü der Metacard-Menüleiste den Befehl "**Group**".

Wenn Sie die neudefinierte "**group**" als "**background**" verwenden wollen, öffnen Sie mit Klick der rechten Maustaste auf eine Stelle innerhalb der neuen Gruppe den "**properties dialog**" und aktivieren Sie unter "**properties**" den unten am Rand befindlichen Button "**Background Behavior**" (in das Feld neben der Bezeichnung klicken). Schließen Sie den properties Dialog.



Wenn Sie jetzt eine neue Seite erzeugen (mit "**create card**" im Edit-Menü), wird die neue Gruppe automatisch auf diese Seite übertragen.

Wollen Sie die neue Hintergrundgruppe auf Seiten übertragen, die schon vor der Definition der neuen Gruppe bestanden, so gehen Sie zu der betreffenden Seite und rufen aus dem "**Edit-Menü**" den Befehl "**Background**" auf. Die neue Gruppe erscheint im Feld "**Stack Backgrounds**" des **„Edit Backgrounds“**-Dialogs.



Klicken Sie den Namen bzw. die Bezeichnung der Gruppe an und klicken Sie anschließend auf "**Place**".

Damit erscheint die neue Gruppe auf der gewählten Seite. Schließen Sie danach den "**Edit Backgrounds**"-Dialog mit Klick auf den Button "**Done**".

Auf ähnliche Weise können Sie Hintergrundgruppen von einer Seite entfernen.

Probieren Sie das aus z.B. mit der Erstellungt von 2 Navigationsbuttons, mit denen Sie dann durch einen Stack blättern können:

2 Buttons mit der Aufschrift "nächste Seite" und "vorherige Seite" o.ä. erstellen

Button 1 erhält das Skript:

```
on mouseup
        go to next card
end mouseup
```

Button 2 erhält das Skript:

```
on mouseup
        go to previous card
end mouseup
```

Führen Sie danach die oben beschriebenen Schritte zur Erzeugung einer Hintergrundgruppe aus.-

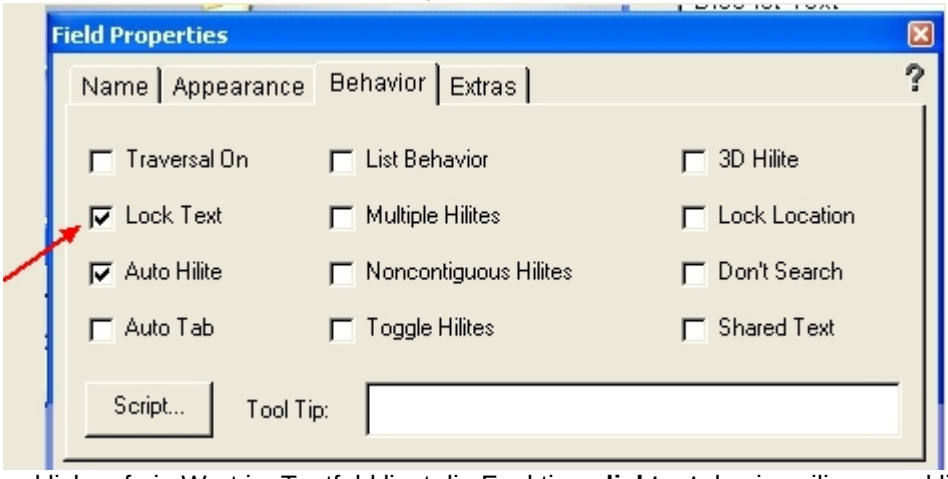
Übung B3

Eine weitere Verknüpfung von Seiten läßt sich ebenfalls über das Anklicken von sogenannten **Hotwords** in Textfeldern auslösen.
Das **Textfeld** erhält dazu folgendes Skript

```
on mouseup
        go to card the clicktext
end mouseup
```

wobei hier das **the** nicht vergessen werden darf.

Im Property Dialog des Feldes muß dazu die Funktion **lock text** angekreuzt werden, damit dieses Skript ausgeführt werden kann. Gleichzeitig wird das Feld „gesperrt“, d.h. es können jetzt keine Veränderungen des Textes vorgenommen werden.
Wenn Sie den Text des Feldes verändern wollen, müssen Sie **lock text** vorher wieder deaktivieren.



Auf einen Mausklick auf ein Wort im Textfeld liest die Funktion **clicktext** das jeweilige angeklickte Wort aus. Falls sich eine Seite gleichen Namens in Ihrem Stack befindet, schaltet Metacard dann zu dieser

Seite weiter.

C. Weitergehende Übungen in Metacard: Felder und Graphiken als Auslöser von Skripten

Da Felder und Graphiken als Objekte ebenfalls mit Skripten verknüpft werden können, lassen sich Befehle wie bei Buttons auch durch Anklicken oder Berühren von Feldern und Buttons auslösen.

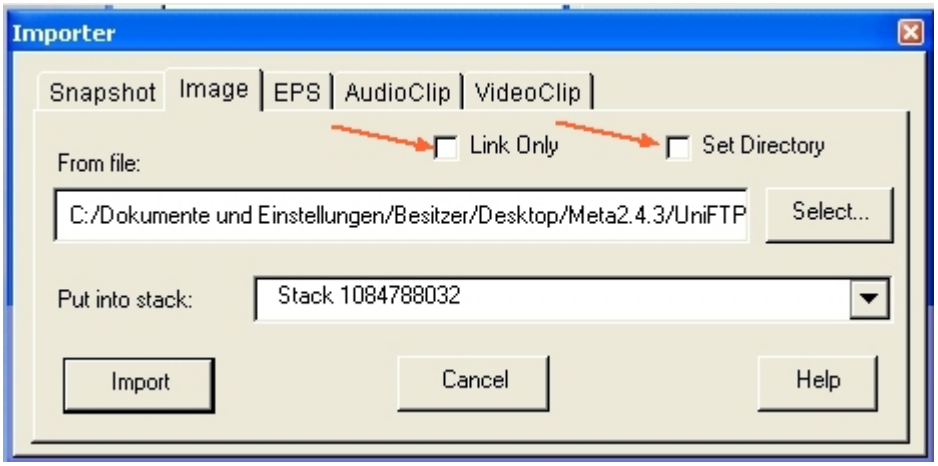
Ü**bung C1: Graphiken (Bilder)**

Bilder werden über den Unterpunkt **Import** im Menü **File** importiert.

1. Im Dialog **Importer** oben auf **Image** klicken

2. Den Button **Select** anklicken und eine Bilddatei aussuchen

3. Nach Auswahl der Bilddatei auf den Button **Import** (unten links) klicken
In der Grundeinstellung des Import-Dialogs wird nur die Verbindung zum Bild über den "Pfad" hergestellt . Soll das Bild Bestandteil des Stacks werden, müssen die „radio buttons“ "Link Only" und „Set Directory“ deaktiviert werden, d.h. wenn sich eine „checkmark“ in dem kleinen Rechteck befindet, muß sie durch Anklicken entfernt werden.



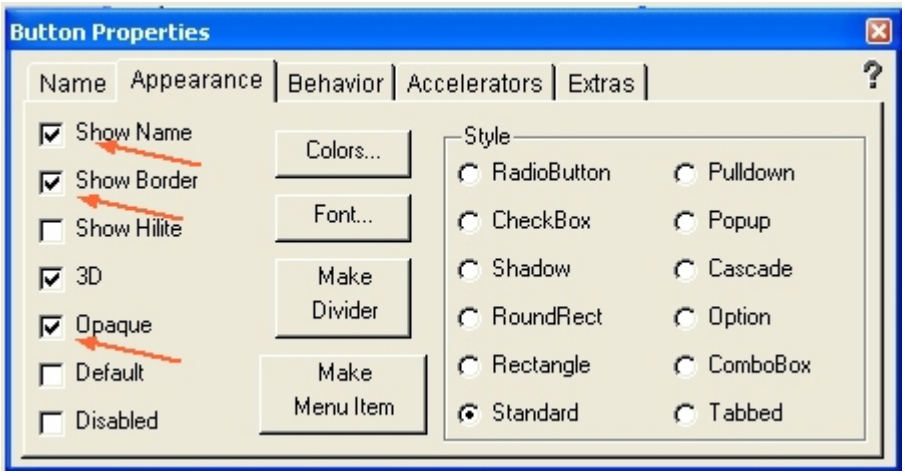
4. Das Bild erscheint in der Mitte der Seite; es kann durch Anklicken aktiviert und mit gedrückter Maustaste an eine beliebige Stelle auf der Seite plaziert werden.

Nach Anklicken des Bildes können Sie im **Image Properties Dialog** unter **Extras** auf den Button **Script** klicken und wie in den Beispielen für die Buttons ein Skript zum Weiterschalten auf eine andere Seite o.ä eingeben. Bei Anklicken der Bilder wirken sie dann wie Buttons.

```
on mouseup
        go to next card
end mouseup
```

Bei größeren Bildern könnte man **transparente Buttons ohne Rand oder mit Rand** über Stellen der Graphik legen, um dann über entsprechende Skripts (eingeleitet mit **mouseup**, **mouseenter** und **mouseleave**) zusätzliche Informationsfelder aufzublenden oder auch zu anderen Seiten zu gehen. **Unsichtbare Buttons ohne Rand** werden in **Button Properties** unter **Appearance** erzeugt:

Die Funktion **opaque** (undurchsichtig) und **Showborder** je nach Wunsch deaktivieren ; daran denken, daß auch **Showname** ausgeschaltet sein muß, falls Sie dem Button einen **Namen** oder ein **Label** gegeben haben (dieses würde sonst auf dem Bild sichtbar sein).



Ü**bung C2: Felder mit Skripten**

Wichtig:

Skripte in Feldern, d.h. nicht der sichtbare Textinhalt von Feldern, sondern der unter **Field Properties** und **Script** hinzugefügte **Programmcode**, werden nur wirksam, wenn die Funktion **lock text** (unter **Behavior**) aktiviert wurde.

Felder können nach **lock text** mit einem Skript dieselben Funktionen wie Buttons oder Hotwords ausführen.

Wenn dagegen der sichtbare Text eines Felder redigiert werden soll, muß lock text vorher wieder ausgeschaltet werden.

Ein nützliches Skript für ein Feld ist folgendes:

```
on mouseup
        hide me
end mouseup
```

Damit verschwindet das Feld, wenn es angeklickt wird. Eine solcher Programmcode ermöglicht es z.B., Felder gezielt oder einzeln verschwinden zu lassen, die vorher über das Anklicken von Buttons, Hotwords oder Graphiken aufgeblendet waren.

==========

Nach diesen Übungen besitzen Sie das elementare, aber ausreichende Werkzeug, um mit **Metacard** durchaus anspruchsvolle Hypertexte herzustellen.

Möglich sind z. B.

Vielfach verknüpfte Informationsbasen mit Übersichten, Glossaren und Einzelverknüpfungen und auch Verzweigungen auf Parallelsequenzen von Seiten mit Zusatzinformationen, die auf den einzelnen Feldern eingeblendet werden können.

Text- und Bildadventures (wenn sie zusätzlich Graphiken einbinden) als verzweigte und dadurch interaktive Geschichten.

Tutorials mit Informationsseiten und Multiple-Choice-Fragen.

(Sanke 2004)

Einführung in die Programmierung mit Hypercarddialekten II

(Die Durcharbeitung der Übungen in Teil I wird vorausgesetzt)

VI. Erweiterte Operationen und Übungen in Metacard 2.5:

Übung A2: Navigation zu einer häufig angesprochenen Hilfsseite (z.B. ein Glossar) - und jeweils zurück zur Ausgangsseite

Für häufig angesprochene Hilfsseiten, die von allen anderen Seiten - etwa über einen Hintergrundbutton - zugänglich sein sollen, ergibt sich das Problem, wie man von der Hilfsseite zur Ausgangsseite zurückgehen kann, ohne auf der Hilfsseite für jede Ausgangsseite einen separaten Button mit **go to card "xy"** anzulegen; stellen Sie sich beispielsweise ein Programm mit 100 und mehr Seiten vor.

In allen Hypercarddialekten gibt es für dieses Problem ein besonderes Befehls-Paar, nämlich die Befehle **push** und **pop**.

Mit **push** wird die Adresse der Ausgangsseite bei den meisten Dialekten programmintern gespeichert, mit **pop** wird sie von der Hilfsseite aus aktiviert.

Das würde in **Metacard** so aussehen:

Für den Hintergrundbutton

```
on mouseup
    push this card -- speichert die Adresse der Ausgangsseite
    go to card "Glossar"
end mouseup
```

Für den "Zurück"-Button (auf der Glossarseite befindlich) zur Navigation von der Glossar-Seite aus zurück zur jeweiligen Ausgangsseite

```
on mouseup
        pop card
end mouseup
```

Auf der Glossarseite sollte der Hintergrundbutton durch ein Feld (ohne Rand) auf dem Vordergrund abgedeckt werden, damit er nicht versehentlich angeklickt wird.

Eine andere Möglichkeit der unmittelbaren Rückkehr zu einer Ausgangsseite ist der Befehl

```
go to recent card
```

oder noch einfacher

```
go recent
```

Erstellen Sie eine Metacard-Datei mit mehreren Seiten und einer Glossarseite und probieren Sie die Befehle **push**, **pop** und **go recent** aus. Denken Sie daran, auf der Glossarseite den Hintergrund-Button mit einem Feld auf dem Vordergrund abzudecken!

Übung A3: Verzweigungen nach Textaufgaben durch Buttons und "aktivierte" Felder

Eines der üblichen Verfahren zur Steuerung in Lernprogrammen ist die Verzweigung durch vorgegebene Antworten, eine der Anwendungen des sogenannten "Multiple-Choice-Verfahrens".

Hier würden auf der Metacardseite nach einem Feld mit Informationen und einer Aufgabe mehrere Buttons folgen, die in ihrer Aufschrift (ihrem "Label") die Antwort, die Alternativantworten und die falschen Antworten enthalten.

Die Skripte dieser Buttons könnten mit **go to card "xy"** zu anderen Seiten führen, auf denen weitere Aufgaben oder auch Hilfen zur richtigen Lösung enthalten sind. Das setzt natürlich eine sorgfältige didaktische Strukturierung der Lerninhalte voraus, die vorher geleistet worden sein sollte, d.h. bevor die Lerninhalte mit **Metacard** in eine Lernsequenz verwandelt werden.

Die Aufschriften (die Labels) der Buttons können sich über die gesamte Bildschirmbreite - und mehr - erstrecken, allerdings wird grundsätzlich nur eine Zeile dargestellt.

Wenn die Auswahlantworten mehr als eine Zeile umfassen sollen, gibt es zwei Möglichkeiten

1. Platzieren Sie vor oder nach den Auswahlantworten , die in Feldern enthalten sind, Buttons mit der Aufschrift **A**, **B**, **C** etc. o.ä.

oder

2. Verwenden Sie die Felder mit den mehrzeiligen Auswahlantworten wie Buttons, indem Sie

   a) die Felder mit einem Skript versehen

```
on mouseup   -- obwohl es sich nicht um einen Button handelt - es ist hier
                          -- die Maustaste gemeint
    go to card "xy"
  end mouseup
```

   und

   b) unter **Field Properties** des jeweiligen Feldes **lock text** anklicken.-

Damit, d.h. nach Durchführung von a) und b), können Sie Felder wie Buttons verwenden.

Sie könnten ebenso **Graphikobjekte** mit einem Skript , wie **go to card "xy",** als Auswahlantworten verwenden oder Graphiken mit Textauswahlantworten kombinieren.

Übung A4: Überprüfung von Eingaben über die Tastatur

Eine andere Form der Beantwortung ist die Eingabe der erwarteten Lösung über die Tastatur; dies erfordert die genauere aktive Kenntnis der Antwort, da nicht bloß unter vorgegebenen sichtbaren Auswahlantworten auszuwählen ist.

Das folgende Skript gehört zu einem Feld, für das Texteingaben, z.B. die Eingabe eines Wortes als Lösung für eine gestellte Aufgabe, vorgesehen sind.
Das Skript läßt alle Tasteneingaben zu und wartet auf das Drücken der Enter-Taste, d.h. „returninfield“ kann im Gegensatz zu anderen Befehlen im Skript eines Feldes nur dann benutzt werden, wenn „**lock text**“ im **Properties** Dialog des Feld nicht aktiviert worden ist.

Skript 1

```
on returninfield

  put me into Eingabe      -- platziere den bis dahin ins Feld eingegebenen Text in
                           --  die Variable "Eingabe";
end returninfield
```

==========

Variablen, d.h. sogenannte "lokale Variablen" innerhalb eines "Handlers" (der mit **on** eingeleitet wird) brauchen in Hypercarddialekten nicht vorher "deklariert" werden, sondern werden dadurch erzeugt, daß man mit dem **put**-Befehl einen Text oder eine Zahl in ein beliebig gewähltes Wort plaziert. Dieses Wort darf nicht mit einem zur Skriptsprache gehörenden Wort identisch sein und es darf z.B. keine deutschen Umlaute einhalten; Den Inhalt des **Feldes "**Lösung**"** müßte man in eine **Variable** Loesung plazieren:

```
put field "Lösung" into Loesung
```

==========

Der Inhalt der Variablen "Eingabe" muß jetzt mit der erwarteten Lösung verglichen werden:

Dazu muß die Lösung irgendwo abgelegt worden sein, z.B. in einem Feld mit dem Namen "Lösung"; dieses Feld sollte mit dem Befehl **hide field "Lösung"** natürlich verborgen sein.

Der Inhalt des Feldes "Lösung" wird nun mit der Eingabe des Lernenden verglichen

Skript 2

```
if Eingabe is field "Lösung" then -- wenn der Inhalt der Variablen "Eingabe"
                                  -- (.s.o.) mit dem Text des Feldes Lösung
                                  -- übereinstimmt, dann
  show field "richtig"        -- zeige das Feld "richtig" (in dem O.K. oder "richtig" steht)
  wait 2 seconds              -- warte 2 Sekunden
  hide field "richtig"        -- verberge das Feld "richtig" wieder
else                          -- wenn die Eingabe nicht richtig war, dann...
  show field "falsch"
  wait 2 seconds
  hide field "falsch"
end if
```

Natürlich lassen sich als positives Feedback auf eine richtige Antwort auch andere Reaktionen vorsehen, z.B. mit **go to card xy** auf eine andere Seite zu gehen, einen "Beep" auszulösen (z.B. **beep 3**; "beep" muß immer mit einer Zahlenangabe stehen), oder im negativen Falle - bei falscher Antwort - eine weitere Eingabe in das Eingabefeld zuzulassen

z.B.:

Skript 2a

```
beep 3
put empty into field "Eingabe" -- löscht den vorher eingegebenen Text
focus on field "Eingabe"  -- setzt den blinkenden Schreibcursor wieder in das Feld
```

oder ebenfalls auf eine andere Seite gehen, die für eine falsche Lösung vorgesehen ist und möglicherweise Hilfen enthält.

Skript 2 müßte jetzt in Skript 1 integriert werden; das ergibt folgendes Bild:

Skript 3

```
on returninfield
  put me into Eingabe

 if Eingabe is field "Lösung" then
   show field "richtig"
   wait 2 seconds
   hide field "richtig"
 else
   show field "falsch"
   wait 2 seconds
   hide field "falsch"
 end if

end returninfield
```

Falls mehrere richtige Antworten möglich sein sollten, müßten sie zunächst sämtlich in das Feld "Lösung" geschrieben werden, zweckmäßigerweise durch Kommata getrennt.
Textteile, d.h. "strings", die durch Kommata abgetrennt sind, können als **items** angesprochen werden. **Items** können Leerstellen enthalten, aber natürlich selbst keine Kommata.
Die verschiedenen richtigen Antwort**Items** im Feld Lösung werden dann nacheinander abgefragt, und zwar in Metacard mit der **repeat**-Prozedur (entspricht z.B. der for-to-Prozedur in Turbo-Pascal und HyperPad):

```
repeat with i = 1 to 3    -- falls 3 richtige Antworten enthalten sind; "i" ist die sich verändernde
                          -- Variable, sie wird von 1 beginnend bei jedem Durchlauf um 1 erhöht

  if Eingabe is item i of  field "Lösung" then
   (hier folgen die Befehle)
  else
 (Befehle)
 end if
end repeat
```

Die lange if-Abfrage könnte kürzer gefaßt werden, wenn man vorher den Inhalt den Feldes Lösung in eine Variable plaziert hätte; Prozeduren innerhalb von Variablen laufen - im Gegensatz zu solchen an Texten in Feldern - wesentlich schneller ab, was natürlich erst ab einer größeren Menge zu durchsuchender Items zu spüren ist!

```
put field "Lösung" into Loesung -- kein "ö" in der Variablen!
repeat with i = 1 to 3
  if Eingabe is item i of Loesung then.....
etc.
```

Da hier 3 richtige Lösungen möglich sind, kann nicht gleich nach der ersten Nicht-Übereinstimmung ein Feedback mit "falsch" gegeben, es muß erst die Überprüfung sämtlicher richtigen Lösungen vorgenommen werden.
Das kann u.a. über eine weitere Variable - z.B. **Zaehler** - erreicht werden, die zu Beginn des Skripts auf Null gesetzt wird und dann bei einer Übereinstimmung erhöht wird.
Abgefragt wird dann, ob die Variable Zaehler den Wert Null oder einen anderen Wert enthält.

Das um diese Funktionen ergänzte bzw. veränderte Skript sieht dann so aus (die Änderungen gegenüber Skript 3 sind unterstrichen):

Skript 4

```
on returninfield
put 0 into Zaehler                      -- die Variable Zaehler wird auf Null gesetzt
  put field "Lösung" into Loesung
  put me into Eingabe
  repeat with I = 1 to 3
    if Eingabe is item i of Loesung then    -- Eingabe wird mit den 3 Lösungen verglichen
      add 1 to Zaehler                      -- die Variable Zaehler wird um 1 erhöht
    end if
  end repeat
  if Zaehler is not 0 then              -- kann auch geschrieben werden "if Zaehler <> 0"
    show field "richtig"
    wait 2 seconds
    hide field "richtig"
  else
    show field "falsch"
    wait 2 seconds
    hide field "falsch"
  end if

End returninfield
```

==========

Bei einer größeren Anzahl von Verschachtelungen von Kontrollstrukturen empfiehlt es sich, das Skript auf mehrere verborgene Buttons aufzuteilen und dann entsprechend mit if-then und mit

send "mouseup" to button "xy"

zu einem anderen Skript zu verzweigen. Dieser Befehl entspricht einem Mausklick auf einen sichtbaren Button.

==========