

# Workshop: Director FOLGE 10

## Einarmiger Bandit

In dieser Folge zeigen wir, wie Sie einen einfachen Spielautomaten programmieren, der mit einem Gewinnspiel gekoppelt ist.

Alle benötigten Dateien finden Sie im Web unter [www.director7.de/workshop](http://www.director7.de/workshop)

■ Das Spiel, das wir in dieser Folge unseres Workshops realisieren, funktioniert in etwa wie ein einarmiger Bandit (Abbildung 1): Mit jeder Betätigung des Hebels generiert Director eine dreistellige Zufallszahl. Enthält die interne Liste der Gewinnnummern die gewählte Zahl, hat der Anwender die Möglichkeit, eine E-Mail zu versenden, die ihn als Gewinner ausweist. Um die Zahl der Versuche einzugrenzen, erzeugt der Director-Film täglich ein Logfile, in dem er den entsprechenden Wert speichert. Nur wenn die maximal zulässige Anzahl noch nicht erreicht ist, kann der User den Hebel betätigen. Ein links im Bühnenbild angeordneter Fortschrittsbalken visualisiert die wachsende Zahl der Versuche.

**Bühnenbild und Drehbuch.** Der Director-Film verwendet eine Bühnengröße von 450 mal 210 Pixeln. Die Maße können

Sie im Dialog „Eigenschaften“ einstellen (Tastenkürzel: Befehl-Umschalt-D, Menüpfad „Modifizieren/Film“). Das Bühnenbild besteht ausschließlich aus der Graphik des Spielautomaten, den drei zur Anzeige der Ziffern verwendeten Textdarstellern sowie der Graphik des Hebels. Den Aufbau des Drehbuchs zeigt Abbildung 2.

Der Hebel soll in den Bildern 1 bis 4 in der oberen Position stehen. Platzieren Sie den Darsteller „hebel.6“ also zum Beispiel in Kanal 4. Die Bewegung des Hebels ab Bild 5 realisieren wir mit den fünf Phasen „hebel.1“ bis „hebel.5“. Sobald der Anwender auf den Hebel klickt, werden die Darsteller nacheinander angezeigt. So entsteht der Eindruck, der Hebel bewege sich abwärts.

Wir erstellen die Animation des Hebels der Einfachheit halber über das Drehbuch. Markieren Sie zunächst die Darsteller „hebel.1“ bis „hebel.5“ in der

Besetzung. Wählen Sie dann die Zelle des Drehbuchs, in der die Sequenz beginnen soll, und verwenden Sie den Befehl „Darsteller in Kanal“ aus dem Menü „Modifizieren“. Verlängern Sie die Animation jetzt, indem Sie das letzte Bild des Sprites ins Bild 16 ziehen. Um den Hebel wieder in die ursprüngliche Position zurückfahren zu lassen, kopieren Sie den soeben erzeugten Sprite, indem Sie ihn etwa mit gedrückter Wahl-taste in das erste leere Bild nach dem Ende des Sprites ziehen. Zur Umkehrung der so duplizierten Sequenz verwenden Sie den gleichnamigen Befehl aus dem Menü „Modifizieren“.

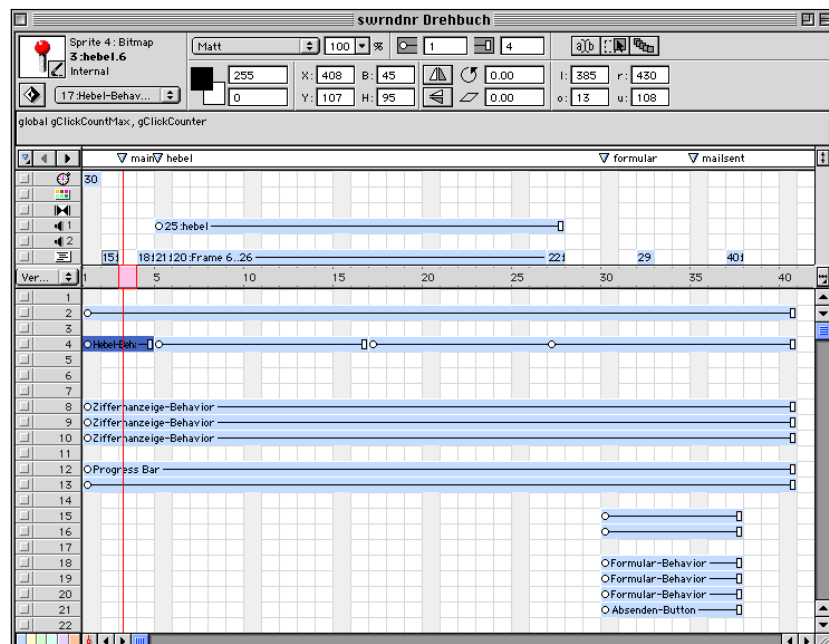
Die Textdarsteller „t.1“, „t.2“, „t.3“ werden mit dem Textwerkzeug der Toolpalette (Befehlstaste-7) oder direkt im Textfenster (Befehlstaste-6) erzeugt. Im Beispiel verwenden wir die Schrift Arial in der Größe 72 Punkt.

In Bild 2 soll der Film auf eine Betätigung des Hebels warten. Das im Skriptkanal von Bild 2 platzierte Verhaltensskript (Behavior) sorgt für das Verharren des Abspielkopfs im aktuellen Bild (Abbildung 3).

### 1: Der fertige einarmige Bandit



### 2: Aufbau des Drehbuchs



**Das Behavior des Hebels.** Das Hebel-Behavior muß vor dem Sprung zum Label „hebel“ – in dem die Animation des Hebels beginnt – prüfen, ob die maximal erlaubte Anzahl von Versuchen bereits überschritten ist. Dazu wird die Variable „gClickCounter“ erst um eins erhöht und anschließend mit der Variablen „gClickCountMax“ verglichen (Abbildung 4). Falls das Maximum erreicht ist, gibt Director eine entsprechende Meldung aus. Andernfalls aktualisiert das Behavior im Kanal 12 den Stand des Fortschrittsbalkens und verzweigt zum Label „hebel“.

Das Behavior von Bild 5 schaltet den Mauscursor für die Dauer der Animation des Hebels ab, um damit weitere Klicks zu verhindern. Die Bilder 6 bis 26 haben ein und dasselbe Behavior (Abbildung 5). Hier verwenden wir den Befehl „sendAllSprites“, um die Nachricht „animate“ an alle Sprites zu versenden. Da nur die Sprites der Textdarsteller in den Kanälen 8, 9 und 10 eine entsprechende Funktion besitzen, könnte das Behavior im Skriptkanal diese Sprites darüber hinaus direkt mit dem Befehl

„sendSprite (<spriteNummer>, <Nachricht>)“ ansprechen. Dieses Vorgehen hätte jedoch den Nachteil, daß wir mit festen Spritenummern arbeiten müßten, die dann im Falle einer Verschiebung im Drehbuch entsprechend zu ändern wären.

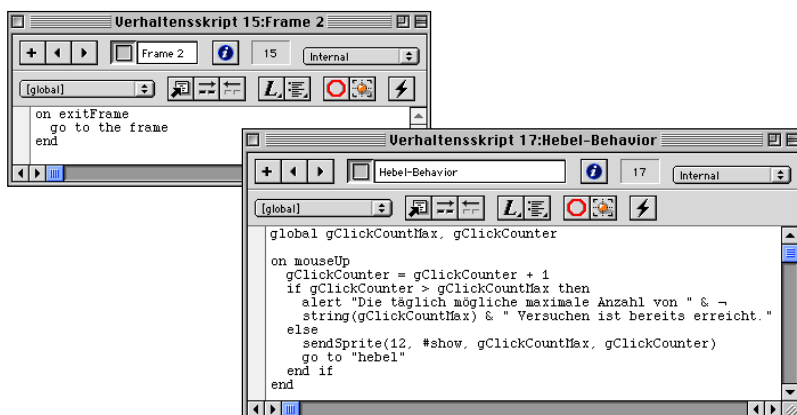
Während der Animation des Hebels soll der Sounddarsteller „hebel“ zum Einsatz kommen. Ziehen Sie den Darsteller dazu einfach aus der Besetzung ins Bild 5, und verlängern Sie gegebenenfalls das letzte Bild des Sprites bis ins Bild 27.

Im Bild 27 sorgt das Behavior im Skriptkanal (Abbildung 6) dafür, daß Director den im Soundkanal 1 plazierten Sound bis zum Ende abspielt und die Textdarsteller weiterhin animiert. Erst wenn der Soundkanal inaktiv ist, die Funktion „soundBusy()“ also den Wert Null liefert, springt Director zum Label „main“ und schaltet den Mauscursor wieder ein.

**Die Bildung der Zufallszahlen.** Die Generierung und Anzeige der einzelnen Zufallszahlen realisieren wir mit der →

### 3: Warten auf Hebel-Aktion

### 4: Behavior des Hebels



→ bereits erwähnten Prozedur „animate“ im Behavior der Textdarsteller (Abbildung 7). Sobald das Behavior die Nachricht „animate“ erhält, ersetzt es den Text des mit dem Darsteller verbundenen Sprites „pMem“ durch eine mit der Funktion „random()“ gebildete Zufallszahl aus dem Bereich 0 bis 9. Anders als in der Programmdokumentation zu lesen, unterstützt die Funktion „random()“ auch den Aufruf mit zwei Parametern: Sobald Sie zwei Werte angeben, interpretiert Director diese als Minimal- und Maximalwert der zu bildenden Zufallszahl.

**Die Gewinnermittlung.** Das Behavior von Bild 4 (Abbildung 8) wird nach der Animation des Hebels und der parallel stattfindenden Erzeugung der Zufallszahlen ausgeführt. Es verknüpft die drei einzelnen Ziffern und speichert den numerischen Wert in der globalen Variablen „gRandomNum“. Den Test, ob die gebildete Zahl in der Liste der Gewinnnummern enthalten ist, realisieren wir mit Hilfe der im Filmskript gespeicherten Prozedur „testNum“. Diese liefert dann den Wert TRUE zurück, wenn es sich um eine Gewinnziffer handelt, und den Wert FALSE, wenn eine Niete ge-

zogen wurde. Im Erfolgsfall verzweigt das Behavior zum Label „formular“, wo der Anwender seine persönlichen Daten eingeben und per E-Mail versenden kann.

Die Prozedur setzt die Gewinnliste „numList“ jedesmal von neuem, um eine globale Deklaration zu vermeiden. Andernfalls wäre es vergleichsweise einfach möglich, die Gewinnliste auszugeben, indem man den Film in Director als Movie in a window (MiaW) öffnet und sich die globalen Variablen im Nachrichtenfenster anzeigen läßt. Eine Liste mit Zufallszahlen ist selbstverständlich auch skriptgesteuert generierbar. Die Prozedur „makeList“ des Filmskripts liefert hierfür ein Beispiel. Abschließend wird die Zahl der unternommenen Versuche in das Logfile im Ordner „Prefs“ geschrieben.

**Das Filmskript.** In der Shockwave-Umgebung sind die Möglichkeiten zur Speicherung von externen Daten aus Gründen der Sicherheit ausgesprochen beschränkt. Das in den vorangegangenen Folgen verwendete Xtra „FileIO“ ist nicht „Shockwave safe“, kann demnach vom Shockwave-Player nicht verwendet werden.

Zum Speichern und Einlesen von Textdateien stehen in Shockwave lediglich die Befehle „setPref“ und „getPref“ zur Verfügung. Mit diesen Befehlen können Sie ausschließlich auf Dateien zugreifen, die im Ordner „Prefs“ liegen. Dieser wiederum befindet sich im Verlauf des Authorings innerhalb des Programmordners von Director. Wenn der Film dann im Browser abgespielt wird, generiert der Shockwave-Player den Ordner „Prefs“ innerhalb des Ordners „Macromedia:Shockwave“ im Ordner „Systemerweiterungen“.

Innerhalb der vor dem Start der Wiedergabe automatisch ausgeführten Prozedur „prepareMovie“ (Abbildung 9) erfolgt zunächst die Bildung des aktuellen Logfile-Namens, in den Director anschließend die Anzahl der Versuche speichern soll. Die Bildung des Dateinamens übernimmt die ebenfalls im Filmskript gespeicherte Prozedur „getLogfileName“. Hier bildet Director den Dateinamen, indem die Zeichenkette „Quiz“ mit dem aktuellen Datum und der Zeichenkette „.txt“ verknüpft wird, wobei vorher die Punkte sowie die ersten beiden Ziffern der Jahreszahl gelöscht werden. Das Logfile für den 15. Oktober 1999 erhielt auf diese Weise

## 5/6: Automat starten / Sound abspielen

## 7: Generierung der einzelnen Ziffern

## 8: Speichern der Zufallszahl

```

Verhaltensskript 20:Frame 6..26
[global]
on exitFrame
  sendAllSprites(#animate)
end
  
```

```

Verhaltensskript 19:Zifferanzeige-Behavior
property pMem
on beginSprite me
  pMem = (sprite me.spritenum).member
end
on animate me
  pMem.text = string(random(0, 9))
end
  
```

```

Verhaltensskript 18:Frame 4
[global] gRandomNum, gClickCounter, gLogFileName
on beginSprite me
  gRandomNum = value((member "t.1").text & -
    (member "t.2").text & (member "t.3").text)
  if testnum(gRandomNum) then
    -- Gewinn
    puppetsound "kling"
    go to "formular"
  else
    puppetsound "no"
  end if
  logStr = string(gClickCounter)
  setPref (gLogFileName, logStr)
end
on exitFrame
  go to the frame
end
  
```

```

Verhaltensskript 22:Frame 27
exitFrame
on exitFrame
  if soundBusy(1) = 0 then
    go to "main"
    cursor -1
  else
    sendAllSprites(#animate)
    go to the frame
  end if
end
  
```

zum Beispiel den Dateinamen „Quiz 151099.txt“. Im nächsten Schritt wird mit Hilfe von „getpref()“ getestet, ob die Datei bereits im Ordner „Prefs“ existiert. Ist dies nicht der Fall, setzt Director die in der Variablen „gClickCounter“ enthaltene Anzahl der Versuche auf den Wert Null und sichert diesen mittels „setpref“ in einer neuen Datei.

Anschließend wird eine Null in die drei Textdarsteller „t.1“ bis „t.3“ geschrieben und ein globales Objekt des Parentskripts „mailer“ erzeugt, das für den Versand der Gewinnbenachrichtigung verantwortlich ist und weiter unten noch genauer vorgestellt wird.

Der Fortschrittsbalken besteht aus zwei mit den Rechteck-Werkzeugen erzeugten QuickDraw-Shapes: einem ungefüllten Rahmen mit fester Abmessung (35 mal 130 Pixel) im Kanal 13 und einem gleich großen gefüllten Rechteck im Kanal 12, dessen Länge später von der aktuellen Anzahl der Versuche bestimmt wird.

Die Ausführung der Prozedur „startMovie“ erfolgt automatisch nach dem Start der Filmwiedergabe. Hier stellen wir nur einen definierten Anfangszustand der Fortschrittsanzeige her, indem wir dem entsprechenden Sprite

die Nachricht „show“ senden. Als Parameter erwartet die gleichnamige Funktion des Behaviors die maximal mögliche und die aktuelle Versuchsanzahl (Abbildung 10).

Das Behavior ändert die Größe des Rechtecks, indem es die Sprite-Eigenschaft „the rect of sprite“ modifiziert. Diese enthält die Koordinaten des den Sprite umschließenden Rechtecks, in der Form „rect(linke, obere, rechte, untere Koordinate)“. Die obere und untere Koordinate lassen sich also in der Punktyntax als „rect[2]“ beziehungsweise „rect[4]“ adressieren. In der Prozedur „beginsprite“ speichern wir die ursprüngliche Höhe des Sprites und dessen Rechteckkoordinaten in den Properties „myHeight“ und „myRect“.

Die Prozedur „show“ ermittelt mit Hilfe des Dreisatzes die obere Koordinate, die der aktuellen Anzahl von Versuchen „cnt“ entspricht. Weil sich die einzelnen Werte in Punkt- oder Rechteckkoordinaten von Sprite-Eigenschaften nicht direkt ändern lassen, muß zunächst eine Übertragung des neuen Werts in die Property „myRect“ erfolgen. Diese verwendet Director danach zum Setzen der Sprite-Eigenschaft „the rect of sprite“.

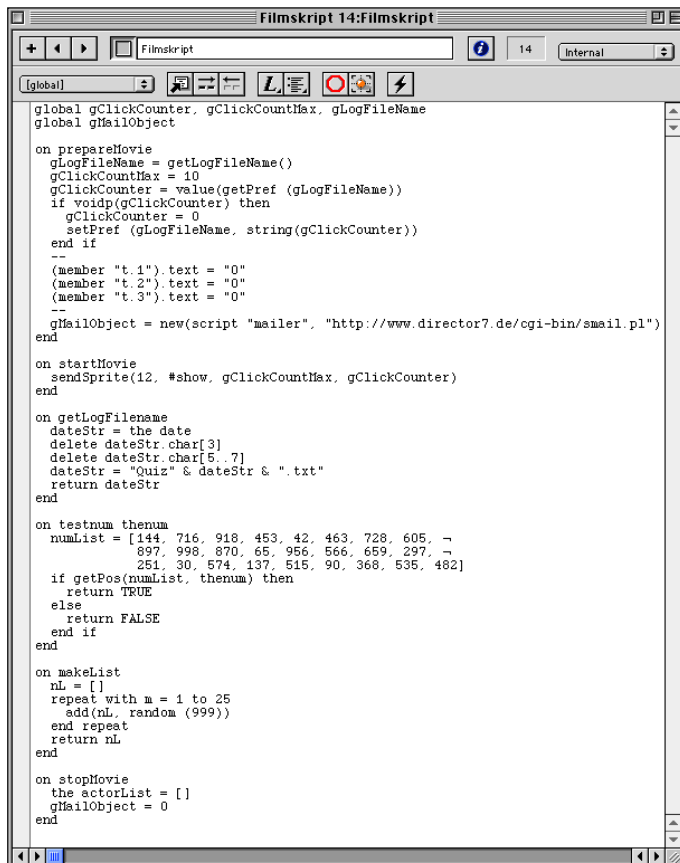
**Das Parent-Skript „mailer“.** Ein Shockwave-Film allein ist nicht in der Lage, E-Mails zu versenden. Eine recht einfache Möglichkeit bietet der Befehl „gotonetpage“. Die folgende Zeile öffnet den Mail-Client des Systems und trägt die als Parameter übergebenen Daten in die entsprechenden Felder der Dialogbox „neue Email“ ein.

```
gotonetpage
("mailto:user@server.com?
subject=SWMail&body=Test")
```

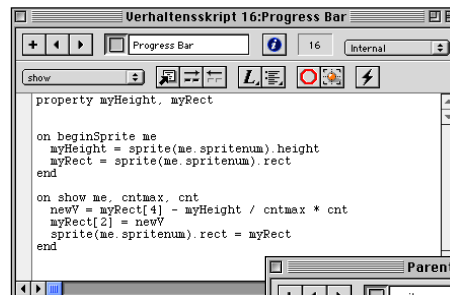
Dieses Verfahren hat den offensichtlichen Nachteil, daß der Anwender zum einen über einen konfigurierten Mail-Client verfügen muß und zum anderen die Daten vor dem Versand editieren und kopieren kann. Insbesondere bei Aktionen wie einem Gewinnspiel ist das nicht unbedingt als wünschenswert anzusehen.

Eine bessere Alternative versendet die Daten mit Hilfe des Befehls „postNetText“ an ein auf dem Server installiertes CGI-Skript, das dann den eigentlichen E-Mail-Versand übernimmt. Die Mailfunktion wird ausgelagert in ein eigenständiges Parent-Skript, das unter anderem eine einfache Wiederver- ➔

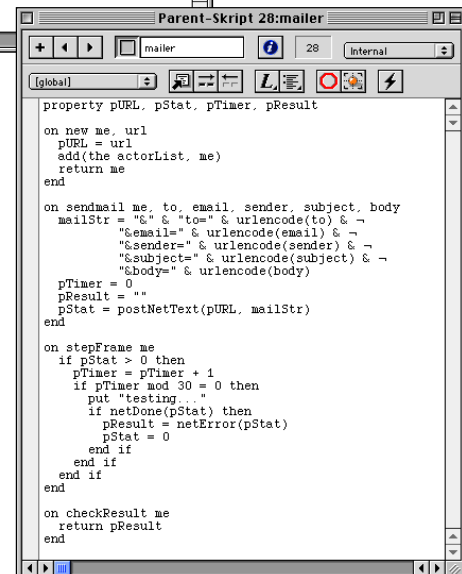
## 9: Das Filmskript



## 10 (oben): Fortschrittsbalken



## 11 (unten): Parent-Skript „mailer“



→ wendung der Programmierung in anderen Projekten erleichtert.

Die Gemeinsamkeiten und Unterschiede zwischen Behaviors und Parent-Skripts haben wir in der vorigen Folge bereits diskutiert. Parent-Skripts unterscheiden sich von Filmskripts dahingehend, daß die enthaltenen Prozeduren nicht direkt aufgerufen werden. Vielmehr bildet Director mit der Funktion „new“ vom Parent-Skript erst eine Instanz im Speicher. Diese Instanz besitzt alle Prozeduren des zugrundeliegenden Skripts und führt diese aus, sobald sie eine gleichnamige Nachricht erhält (Abbildung 11).

In unserem Beispiel erfolgt der Aufruf von „new“ im Filmskript innerhalb der Prozedur „prepareMovie“. Jede „new“-Prozedur muß mindestens den Befehl „return me“ enthalten. Er liefert die Objektinstanz an die aufrufende Prozedur zurück. Beim Aufruf von „new“ lassen sich noch weitere, durch Kommata getrennte Parameter übergeben. Der Parameter „url“ enthält hierbei die Adresse des CGI-Skripts, an das die jeweiligen Maildaten gesendet werden sollen. Sein Speicherort ist die Property „pURL“.

Der Befehl „add(the actorList, me)“ fügt die Objektinstanz der sogenannten „ActorList“ hinzu. Dabei handelt es

sich um eine spezielle, automatisch von Director verwaltete Liste. Sobald sich der Abspielkopf bewegt oder der Befehl „updateStage“ verwendet wird, erhalten sämtliche in der ActorList enthaltenen Objekte die Nachricht „stepFrame“. Sofern ein Objekt eine gleichnamige Prozedur besitzt, kommt diese somit zur Ausführung.

Prinzipiell läßt sich „stepFrame“ mit „enterFrame“ in anderen Skripts vergleichen. In unserem Beispiel verwenden wir „stepframe“, um den Abschluß und den Erfolg der Netzoperation zu prüfen, die vom Befehl „postNetText“ innerhalb der Prozedur „sendmail“ initiiert wurde. Beachten Sie dabei, daß Director die ActorList beim Beenden der Wiedergabe oder beim Aufruf eines anderen Films nicht automatisch löscht. Dies kann laut Programmdokumentation „zu unvorhersehbarem Verhalten im neuen Film führen“. Da dies ebenso für den wiederholten Start ein und desselben Films gilt, löschen wir die ActorList bei Beendigung der Wiedergabe des Films in der Prozedur „stopMovie“ des Filmskripts.

Beim Aufruf werden der Prozedur „sendmail“ sämtliche zum Mailversand benötigten Daten übergeben. Daraufhin codiert sie die einzelnen Bestandteile für den Versand und verknüpft sie

zu einer gemeinsamen Zeichenkette. Den Versand der Zeichenkette „mail-Str“ an das CGI-Skript bewirkt der Befehl „postNetText“. Dieser liefert – wie die meisten anderen Netzoperationen auch – eine eindeutige Net-ID zurück, die sich zum Test des Abschlusses der Operation verwenden läßt. Diese Net-ID speichern wir innerhalb der Property „pStat“.

Sobald „pStat“ einen anderen Wert als Null hat, erhöhen wir in der Prozedur „stepframe“ die Property „pTimer“ um eins. Um das System nicht mit fortwährend aufeinanderfolgenden Abfragen des Netzstatus zu überlasten, wird die Prüfung nur dann zugelassen, wenn „pTimer“ einen durch 30 teilbaren Wert hat. Da im Tempokanal eine Bildrate von 30 Bildern pro Sekunde eingestellt ist, erfolgt die Prüfung somit einmal pro Sekunde.

In unserem Beispiel erfolgt der Aufruf von „sendmail“ ausschließlich vom Behavior des Buttons „Absenden“ aus. Allerdings spricht nichts dagegen, die Bildung des Objekts und den Versand einer Nachricht durch Aufrufe im Nachrichtenfenster zu testen. Beachten Sie dabei jedoch, daß die Wiedergabe des Films laufen muß, damit die Prozedur „stepFrame“ – und damit die Prüfung des Abschlusses der Netzoperation –

## 12/13: Test des Mail-Skripts

## 14 (unten): Das Mail-Formular

## 15: Textdarsteller-Eigenschaften

```

Nachrichten
-- Willkommen bei Director --
gMailObject = new(script "mailer", "http://www.director7.de/cgi-bin/Smail.pl")
put gMailObject
-- <offspring "mailer" 3 67b9146)
sendmail(gMailObject, "gewinnspiel@bfim.com", "gerd@gillmaier.de", "GG", "Gewinnspiel", "nur ein Test :-)")
put checkResult(gMailObject)
-- "OK"

```

```

Gewinnspiel
Sender: postmaster+48328@post.webmail.de
Received: from aslmov.dns-host.com (aslmov.dns-host.com [209.225.102.21])
by spangaa.comuserve.com (8.9.3/8.9.3/SUN-1.6) with ESMTP id HAA02711
for <gillmaier@csi.com>; Wed, 15 Sep 1999 07:01:53 -0400 (EDT)
Received: from post.webmail.de ([192.67.198.38])
by aslmov.dns-host.com (8.9.1/8.9.1) with ESMTP id HAA15670
for <gewinnspiel@bfim.com>; Wed, 15 Sep 1999 07:01:51 -0400 (EDT)
Received: from chroot.webmail.de ([1web2-local [172.23.44.35])
by post.webmail.de (8.9.3/8.9.7) with ESMTP id NAR24913
for <gewinnspiel@bfim.com>; Wed, 15 Sep 1999 13:01:49 +0200 (MET DST)
Received: (from Unknown UID 48328@localhost)
by chroot.webmail.de (8.9.3/8.9.7) id LRA11313;
Wed, 15 Sep 1999 11:01:47 GMT
Date: Wed, 15 Sep 1999 11:01:47 GMT
Message-Id: <199909151101.LRA11313@chroot.webmail.de>
To: gewinnspiel@bfim.com
From: gerd@gillmaier.de (GG)
Subject: Gewinnspiel

nur ein Test :-)

```

Textdarsteller-Eigenschaften

Rahmen: **Anpassen**

Pre-Render: **Keine**

Bitmap speichern

Optionen:  Bearbeitbar

Mit Tabulatortaste zum nächsten Element

Direkt auf Bühne

Hypertext-Stil verwenden

Anti-Alias:  Gesamter Text

Größer als  Punkt

Kein

Unterscheidung:  Gesamter Text

Größer als  Punkt

Kein

OK  
Abbrechen  
Hilfe

Bühne

Sie haben gewonnen! Bitte geben Sie Ihren Namen, Ihre Telefonnummer und Ihre E-Mail-Adresse an und klicken einmal auf den Button "Absenden".

Ihr Name:

Ihre Telefonnr.:

Ihre E-Mail:

zur Ausführung kommen kann (Abbildungen 12 und 13).

**Das Mail-Formular.** Zum Label „Formular“ (Abbildung 14) gelangt der Anwender lediglich, wenn er eine der Gewinnziffern gezogen hat. Geben Sie den im Formular verwendeten Darstellern die Namen „name“, „telefon“ und „email“. Wählen Sie dann in den „Optionen“ des Befehls „Darstellerinfo“ die Einträge „Bearbeitbar“ und „mit Tabulatortaste zum nächsten Element“ (Abbildung 15). Das darauf folgende Behavior (Abbildung 16) verhindert mehrzeilige Eingaben mit der Return- beziehungsweise der Eingabetaste.

Das Behavior des Buttons „Absenden“ (Abbildung 17) bildet aus den Eingaben des Anwenders sowie aus der gezogenen Zufallszahl und dem aktuellen Datum samt Uhrzeit den zu sendenden Text der E-Mail. Anschließend sendet es dem globalen Objekt des Parent-Skripts „mailer“ die Nachricht „sendmail“, wobei unter anderem der Text der E-Mail im Parameter „mailBody“ übergeben wird (Abbildung 18).

Den dabei erscheinenden Dialog erzeugt Director prinzipiell immer dann, wenn der Film im Laufe der Wiedergabe im Browser auf externe Daten zugreifen will. Nach dem Start des Mailversands

verzweigen wir zum Label „mailed“, wo wir auf den Abschluß der Netzoperation warten.

Das im Skriptkanal von Bild 37 platzierte Behavior (Abbildung 19) ist für die Ausgabe der Erfolgs- beziehungsweise Fehlermeldung nach dem Versand der Nachricht verantwortlich. Um im Falle einer fehlerhaften Netzanbindung die Bildung einer Endlosschleife des Programms in diesem Bild zu verhindern, wird innerhalb der Prozedur „beginSprite“ der Stand des internen Zählers „the timer“ gespeichert und die maximale Wartezeit auf 30 Sekunden begrenzt.

Stellt die Prozedur „enterFrame“ eine Überschreitung dieser Zeitspanne fest, verzweigt sie nach der Ausgabe einer Fehlermeldung zum Label „main“. Anschließend sendet die Prozedur die Nachricht „checkResult“ an das globale Objekt des Parent-Skripts „mailer“ worauf dieses entweder mit der Zeichenkette „OK“ oder mit einer Fehlernummer antwortet.

Das im CGI-Verzeichnis des Servers zu installierende Perl-Skript, auf das wir hier aus Platzgründen nicht weiter eingehen, finden Sie unter [www.director7.de/workshop](http://www.director7.de/workshop). Achten Sie beim Upload auf die Übertragung als Text, und überprüfen Sie die Zugriffsrechte

der Datei. Alle gängigen FTP-Tools bieten einen passenden Befehl, in Fetch 3.0.3 verwenden Sie beispielsweise „Set Permissions“ aus dem „Remote“-Menü (Abbildung 20). Beachten Sie beim Einsatz des Skripts auf anderen Servern, daß auch die im Filmskript enthaltene URL sowie die für „sendmail“ verwendete Mail-Adresse entsprechend zu ändern sind. **Gerd Gillmaier** ■

## LINKS

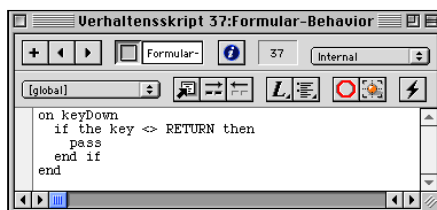
■ Weitere Infos zum Thema Shockwave-CGI-Kommunikation gibt es hier:

- [www.updatestage.com/previous/990301.html#item2](http://www.updatestage.com/previous/990301.html#item2)
- [www.updatestage.com/previous/971101.html#item2](http://www.updatestage.com/previous/971101.html#item2)
- [www.mcli.dist.maricopa.edu/director/tips.html](http://www.mcli.dist.maricopa.edu/director/tips.html)
- [www.snafu.de/~dp/util](http://www.snafu.de/~dp/util)
- [worldwidemart.com/scripts](http://worldwidemart.com/scripts)

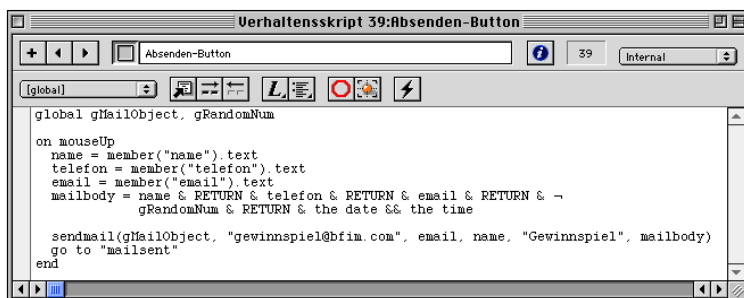
## VORSCHAU

■ Was wir in der nächsten Folge behandeln, liegt in Ihrer Hand: Schicken Sie Ihre Wünsche per Mail an [workshop@director7.de](mailto:workshop@director7.de)

16: Verhindern mehrzeiliger Eingaben



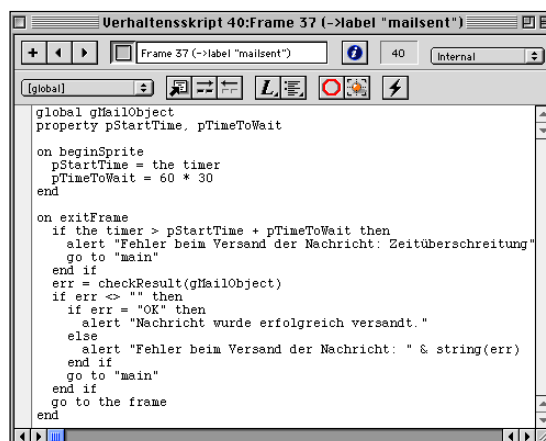
17: Behavior des „Absenden“-Buttons



18: Beispiel für einen E-Mail-Text



19: Behavior für Rückmeldung



20: Setzen der CGI-Zugriffsrechte

