

Flash-Filme in Director

Macromedia bietet neben Director auch ein anderes, vektorbasiertes Animationsprogramm, nämlich Flash an. Flash erlaubt die einfache Erstellung von Vektor-Animationen, vorzugsweise für das Internet. Es können aber auch analog zu Director-Projektoren Präsentationen erstellt werden.

DIRECTOR UNTERSTÜTZTE SEIT DIRECTOR 6.5 DEN IMPORT VON SWF-DATEIEN (fertigen Flash-Filmen) in Director. Director 7.0 unterstützte das Format Flash 3 und das Update 7.02 auch Flash 4-Dateien sowie deren erweiterte Programmierung und Variablen. Director 8 nun erlaubt auch den Zugriff auf die Eigenschaftsvariablen in einem Flash-Film.

Dieses Kapitel wird sich verständlicherweise mit einigen Techniken in Flash auseinandersetzen müssen, jedoch nur in Bezug auf die Handhabung in Director. Sollten Sie ein weitergehendes Interesse an der Entwicklung von Flash-Inhalten haben, so möchten wir Sie hiermit auf das wirklich hervorragende Buch »Flash 4« von Sascha Wolter (ISBN: 3-934358-05-5) verweisen, das sich ausschließlich mit der Flash-Programmierung beschäftigt.

Die Möglichkeiten von Flash

Das Programm Flash wurde von einer Firma namens FutureWave entwickelt und hieß zunächst FutureSplash. Nach dem Kauf von Macromedia wurden das Programm und das Format weiterentwickelt und es etablierte sich schnell zum Standard für interaktive Präsentationen im Internet. Ein kleines PlugIn von gerade mal 170KB Größe, gepaart mit einer aggressiven Distributionspolitik, haben heute aus Flash eines der weit verbreitetsten Formate im Internet gemacht. Vor kurzem hat Macromedia angekündigt, nach Windows und MacOS das Flash-PlugIn auch für Unix-Rechner zu vertreiben.

Flash bietet die einfache Erstellung von vektorbasierten, bildschirmfüllenden Animationen, gepaart mit logischer Flußsteuerung und Variablen.

Hinzu kommt das Einfügen von Audiodaten, die zeitsynchron zur Animation ausgerichtet werden können. Flash 2/3 unterstützte eine WAV-Audiokompression nach dem ADPCM-Kompressionsschema, bei Flash 4 kam noch eine MP3-Audiokompression von 8 Kbps bis hinauf zu 160 Kbps hinzu.

Das Hauptziel von Flash sind wohl weiterhin flotte Web-Präsentationen mit begrenzter Interaktivität, die extrem schnell geladen werden können und optisch dank Anti-Aliasing und ausgefeilter Animationstechniken überzeugen können.

Um sich einen Eindruck der Möglichkeiten von Flash zu verschaffen, empfehlen wir die besonders schöne Website der englischen Agentur Matinee: <http://www.matinee.co.uk>.

Weiterführende Infosites

URLs zu guten Flash-Sites finden Sie unter anderem unter <http://www.flashplanet.com/>.

Ausführliche technische Dokumentationen finden Sie unter <http://www.FlashCentral.com/Tech/>

Für Entwickler liegt eine Definition des Dateiformats unter <http://www.flash.com/open> bereit.



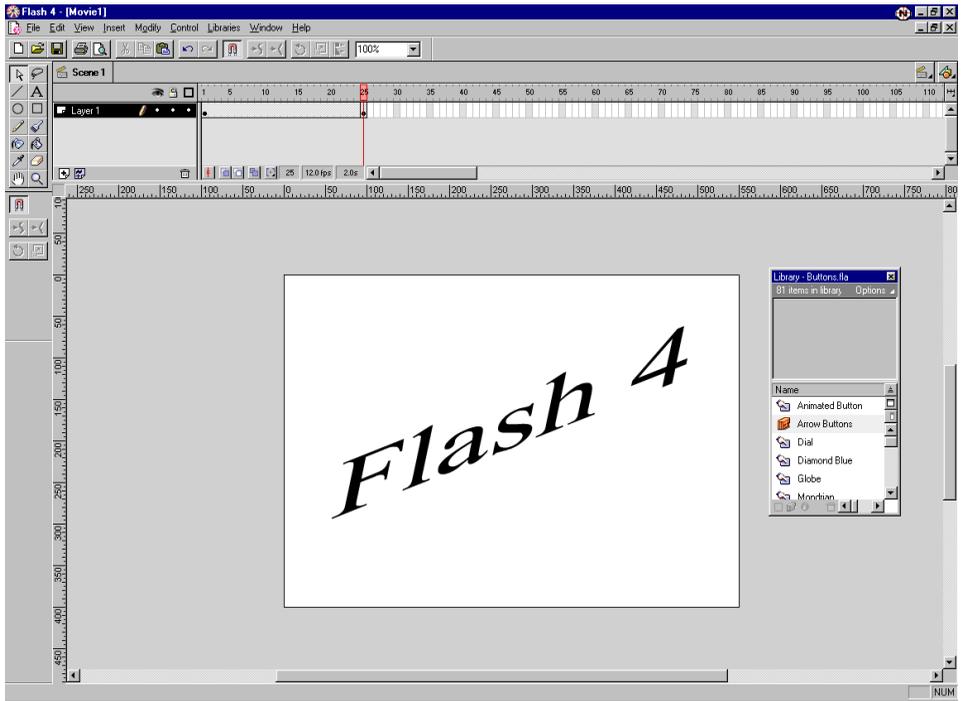
Arbeiten in Flash

Dem Programmierer bietet Flash einen ähnlichen Aufbau wie Director: Grafiken und Klänge werden in einer sogenannten Bibliothek analog zur Besetzung in Director angelegt und auf einer Bühne über eine Zeitleiste animiert.

Unterschiede zwischen Flash und Director gibt es allerdings insbesondere im Bereich der Möglichkeiten: Flash erlaubt weiterhin eigentlich »nur« eine erweiterte Flußkontrolle des Films, während Director mit Lingo eine eigene Programmiersprache stellt. Zwar bietet Flash 4 nun auch Variablen und die Prüfung komplexer Ausdrücke an, jedoch sind die Möglichkeiten bei weitem nicht so mächtig wie in Lingo.

Der Flash Generator

Zusätzlich zu dem Autorenwerkzeug Flash hat Macromedia auch einen sogenannten Flash Generator herausgebracht, mit dem man Flash-Filme



▲ **Abbildung 1**
Das Benutzerinterface von Flash 4 ähnelt dem von Director

zur Laufzeit mit Inhalten aus Datenbanken bestücken kann, so daß interaktive und ständig aktuelle Inhalte erzeugt werden können. Dank Coldfusion, ASP oder PHP können so Datenbankinhalte in einer animierten und attraktiven Art und Weise präsentiert werden.

Es soll an dieser Stelle aber auch nicht verschwiegen werden, daß es bereits auch schon Alternativen zu Generator gibt. Der vielversprechendste Kandidat ist SWIFT von Olivier Debon, zu finden unter <http://www.swift-tools.com>.



Weiterführende Infos

Mehr Infos dazu präsentiert Macromedia unter <http://www.macromedia.com/software/generator/>. Informationen für Entwickler, wie zum Beispiel die Anbindung einer Datenbank an Generator, finden sich unter <http://www.flashgen.com>.

Flash und Director

Bei der Einbettung von Flash-Filmen in Director gibt es eine Reihe von technischen Problemen, die durch eine thematische Überschneidung der beiden Programme resultieren:

- ▶ Flash-Filme sind in sich gekapselte interaktive Filme mit eigenen Tempoangaben und Interaktivität, die sich nahtlos in Director einfügen sollen. Director hat aber sein eigenes Tempo und seine eigene Skriptsprache
- ▶ Flash-Filme kommen mit ihrer eigenen Musik und Soundeffekten, die eventuell mit anderen Sounds in Director gemischt werden müssen
- ▶ Flash-Filme bestehen zum großen Teil aus Vektorgrafiken, Director ist aber eher ein pixelbasiertes Programm

Die Lösung für diese Probleme in Director werden wir uns nun genauer ansehen.

Die Einbettung von Flash-Filmen in Director

Der Schlüssel für die Einbettung von Flash-Filmen in Director ist das »Flash Asset«-Xtra. Ein weiteres Xtra namens »Flash Asset Options« stellt die Import-Möglichkeiten in Director zur Verfügung.

Flash-Filme können importiert, extern verlinkt oder auch über das Internet geladen werden, wie jeder andere Darsteller auch, inklusive Streaming, das heißt dem Laden der Datei während des Abspielens. Die Darstellungsleistung von Flash-Filmen in Director kann durch die Nutzung der Option DIRECT-TO-STAGE analog wie bei Video-Clips erhöht werden. Dafür können dann aber keine Sprites mehr über den Film gelegt werden.

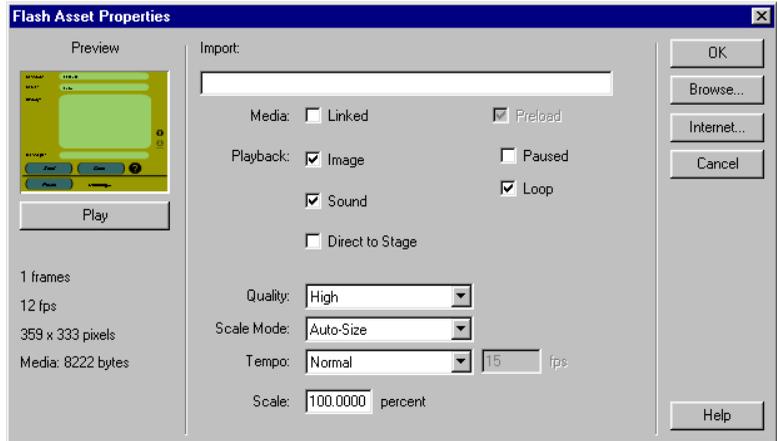
Flash-Filme können in Director normal verzerrt, rotiert und gekippt werden, sie werden dennoch weiterhin sauber abgespielt und bleiben interaktiv.

Intern wird ein Flash-Film in Director mit einer Reihe von Eigenschaften abgelegt, die alle dem Programmierer per Lingo zugänglich sind. Der spezielle Befehl

```
showprops member ("flashfilm")
```

ermöglicht eine Ausgabe dieser Eigenschaften, allerdings nur in der Director-Umgebung, nicht in Projektoren oder Shockwave-Filmen:

Abbildung 2 ►
Die Optionen des Flash
Asset Options-Xtra



showProps member 1

```

preload:          1
bufferSize:      32768
directToStage:   0
centerRegPoint:  1
regPoint:        point(275, 200)
defaultRect:     rect(0, 0, 550, 400)
sound:           1
imageEnabled:    1
pausedAtStart:  1
loop:            1
quality:         #high
playbackMode:    #normal
fixedRate:       15
scale:           100.0000
rotation:        0.0000
originMode:      #center
originPoint:     point(0, 0)
originH:         0.0000
originV:         0.0000
viewScale:       100.0000
viewPoint:       point(0, 0)
viewH:           0.0000

```

```

viewV:                0.0000
broadcastProps:       1
scaleMode:            #autoSize
static:               0
buttonsEnabled:       1
actionsEnabled:       1
eventPassMode:        #passAlways
clickMode:            #opaque
posterFrame:          1
streamMode:           #frame
defaultRectMode:     #flash
obeyScoreRotation:    1
percentStreamed:      100
bytesStreamed:        305569
streamSize:           305569
state:                4
flashRect:            rect(0, 0, 550, 400)
frameRate:            20
frameCount:           2513
pathName:             0
linked:               0

```

Allein diese Liste zeigt schon, daß Flash-Filme in Director als vollkommen abgekapselte Entitäten abgelegt werden. Die meisten Eigenschaften betreffen dabei die visuelle Darstellung des Flash-Films.

Lingo für Flash-Filme

Um die weiter oben angesprochenen Überschneidungen bezüglich der Interaktivität und der Sounds von Flash-Filmen in Director in den Griff zu bekommen, hat Macromedia mit Director 7 eine Reihe von neuen Lingo-Befehlen für die Steuerung von Flash-Filmen veröffentlicht.



Diese Befehle lassen sich grob in vier verschiedene Gruppen unterteilen:

1. Interaktivität
2. Darstellung
3. Streaming
4. Handhabung der Flash-Filme in Director

Es folgt daher nun eine kurze Übersicht dieser Befehle.

Lingo-Befehle zur Interaktivität

Für Flash und Director bedeutet Interaktivität vor allen Dingen Mausbewegungen, insbesondere Ereignisse wie Rollovers und Mausklicks. Da beide Filmformen auf solche Ereignisse reagieren können, kann man in Director durch das Setzen von Eigenschaften klären, welcher Film nun auf diese Aktionen reagieren soll. Für den entsprechenden Film werden dann die Ereignisse durchgelassen. Soll etwa der Flash-Film interaktiv sein, so wird Director die Mausereignisse an den Film weiterleiten, ansonsten wird er sie abfangen und als Ereignisse des Sprites für eigenen Zwecke benutzen. Mit den folgenden Lingo-Befehlen kann man die Interaktivität eines Flash-Films in Director beeinflussen:

The ActionsEnabled

Eine Eigenschaft eines Flash-Films, die bestimmt, ob die Interaktivität innerhalb eines Flash-Films an oder aus sein soll. Wenn Sie aus ist, erkennt der Flash-Film nur noch Mouseover-Ereignisse für Schaltflächen aber keine Mausklicks mehr. Dies entspricht dem Menüpunkt AKTIONEN AKTIVIEREN aus dem CONTROL-Menü in Flash.

Beispiel:

```
sprite (me.spritenum).actionsenabled=false
```

the ButtonsEnabled

Eine weitere Spriteeigenschaft für Flash-Filme, die dem Menüeintrag BUTTONS AKTIVIEREN in Flash entspricht. Ist dieser Wert **false**, so werden die Knöpfe in einem Flash-Film ausgeschaltet. Standardmäßig steht dieser Wert nach jedem Neustart eines Director-Projektes wieder auf **true**, um Interaktivität in Bezug auf die Knöpfe zu erlauben.

Beispiel:

```
Sprite (me.spritenum).buttonsenabled=false
```

Hold

Der Befehl **hold** hält einen abspielenden Flash-Film in seiner Ausführung an. Falls jedoch Schaltflächen aktiviert sind, kann der Film nach einem Mausklick oft dennoch weiterspielen. **Hold** pausiert somit nur den Film.

Beispiel:

```
Hold sprite 1
```

PausedAtStart

Diese Eigenschaft eines Darstellers (nicht Sprites!) ist ausschlaggebend dafür, ob der Film bei seinem ersten Auftritt auf der Bühne abgespielt wird oder im Pausemodus verharrt. Mögliche Wert sind `true` oder `false`. Im Falle von `true` spielt der Film nicht vor der Ausgabe eines Play-Befehls für das entsprechende Sprite.

Beispiel:

```
Member("Flashfilm").pausedatstart=true
```

Lingo-Befehle zur Darstellung von Flash-Filmen

Jeder Flash-Film ist eine Vektoranimation, ein Umstand, den Director als pixelbasiertes Animationssystem integrieren muß. Flash-Animationen können beliebig skaliert werden und bleiben dennoch von ihren Proportionen her gleich. Auch dies ist innerhalb von Director neu, denn bisher arbeitete man mit Pixelgrafiken, die genaue Abmessungen in bezug auf die Bühnengröße hatten. Weiterhin hat jede Flash-Animation auch noch zusätzlich ihr eigenes Tempo, eventuell im Gegensatz zum vorgegebenen Tempo in Director. Es folgt deswegen eine Aufzählung der Eigenschaftsvariablen für Flash-Filme, die die Darstellung von Flash-Filmen in Director beeinflussen:

Framerate

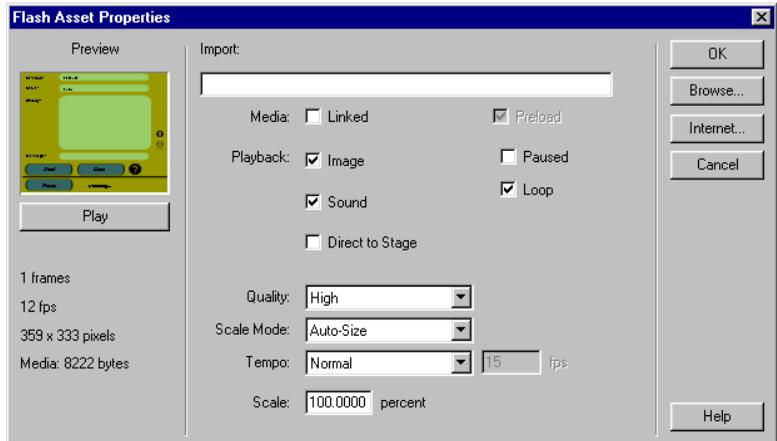
Die Eigenschaft `Framerate` liefert das Ursprungstempo des Flash-Films zurück. Sie kann nur ausgelesen werden.

Beispiel:

```
put member("flashfilm").framerate  
-- 20
```

Inwieweit nun das Tempo des Flash-Films mit dem Tempo in Director kollidiert, ist eine Frage der eingestellten Optionen im Flash-Asset Xtra,

Abbildung 3 ►
Das Feld TEMPO
bestimmt, wie schnell
ein Flash-Film in Director
abgespielt wird.



zu finden unter den EIGENSCHAFTEN DES DARSTELLERS, dann OPTIONEN. Die Option TEMPO bestimmt, welches Tempo der Flash-Film beim Abspielen annimmt:

1. Die Standardeinstellung NORMAL läßt den Flash-Film mit Originaltempo ablaufen, soweit möglich.
2. Mittels der Einstellung FESTE BILDRATE kann man dafür Sorge tragen, daß der Flash-Film ein bestimmtes Abspieltempo hat. Die exakte Bildrate wird in das Kästchen daneben eingetragen. Sollte die Bildrate zu hoch sein, so läuft der Film mit der höchstmöglichen Geschwindigkeit.
3. Die Einstellung LOCK-STEP läßt je ein Bild des Flash-Films pro Einzelbild in Director ablaufen und eignet sich somit gut zum **Synchronisieren** von Inhalten.

The ImageEnabled

Die Eigenschaft **ImageEnabled** schaltet die Darstellung der Bilder eines Flash-Films an und aus und kann sowohl für einen Darsteller als auch für ein Sprite gesetzt und ausgelesen werden. Mögliche Wert sind **true** und **false**, Standardwert ist **true**. Wird diese Eigenschaft auf **false** gesetzt, so wird das Sprite mit dem Flash-Film transparent und spielt nur noch Sound ab. Dieser Wert entspricht der Checkbox VIDEO im oben abgebildeten Flash Asset Xtra.

Beispiel:

```
Sprite(1).imageEnabled=false  
put member(1).imageenabled  
-- 1
```

OriginMode, OriginV, OriginH, OriginPoint

Alle vier oben aufgeführten Befehle beschäftigen sich mit dem Ursprungspunkt eines Flash-Films, auf englisch dem »Originpoint«. Dieser Punkt bestimmt, wo eine Grafik ihren Rotationspunkt und Skalierungsurprung hat.

The **originMode** ist für jeden Flash-Darsteller beziehungsweise jedes Flash-Sprite eine Eigenschaft, der einer von drei symbolischen Werten zugeordnet werden kann:

#center	Die Mitte des Flash-Films beziehungsweise des Sprites ist der Ursprungspunkt.
#topleft	Die obere linke Ecke ist der Ursprungspunkt.
#point	Ein weiter mit originV und originH oder auch OriginPoint anzugebender Punkt ist der Ursprungspunkt.

Die Befehle **OriginV**, **OriginH** und **OriginPoint** können erst benutzt werden, wenn the **originmode** auf **#point** gesetzt wurde. The **originpoint** nimmt einen Punkt entgegen, **OriginH** und **OriginV** jeweils nur die horizontale beziehungsweise vertikale Koordinate für den Ursprungspunkt.

Beispiel:

```
sprite(1).originmode=#point  
sprite(1).originpoint=point (50,50)
```

ist äquivalent zu

```
sprite(1).originH=50  
sprite(1).originV=50
```

the playBackmode

Dies ist die entsprechende Eigenschaftsvariable zur TEMPO-Einstellung im Flash Asset Xtra. Diese Variable kann folgende symbolische Werte annehmen:

#normal	(Standard) Flash-Film wird soweit möglich mit Originaltempo abgespielt.
#lockstep	Flash-Film wird Bild für Bild zum Director-Film abgespielt.
#fixed	Flash-Film wird mit fester Bildrate abgespielt, die mit fixedrate angegeben werden kann – Standardwert ist 15.

Beispiel:

```
member(1).playbackmode=#fixed  
member(1).fixedrate=10
```

the playing

Als eine Eigenschaftsvariable eines Sprites gibt **theplaying** nur auslesbare Ergebnisse bezüglich des Abspielmodus eines Flash-Films.

Beispiel:

```
put sprite(1).playing  
-- 1
```

the posterframe

Diese Darstellervariable kann gelesen und gesetzt werden und dient dazu, ein Bild in einem Flash-Film als Thumbnail in der Besetzung auszuwählen.

Beispiel:

```
member("Flashfilm").posterframe=150
```

the quality

Mit **quality** kann mal sowohl auf der Ebene eines Darstellers als auch der eines Sprites die Abspielqualität für den jeweiligen Flash-Film festsetzen. Möglich sind vier symbolische Werte, die den Eintragungen unter **SCALE** im Flash Asset Xtra entsprechen:

#autohigh	Der Flash-Film wird zunächst in hoher Qualität mit Anti-Aliasing abgespielt. Sollte die Abspielgeschwindigkeit aufgrund technischer Begrenzungen des Computers unter den Ursprungswert fallen, so wird die niedrige Darstellung gewählt.
#autolow	Der Flash-Film wird zunächst in niedriger Qualität, d. h. ohne Anti-Aliasing abgespielt. Sollte der Prozessor in der Lage sein, den Inhalt auch in hoher Qualität abzuspielen, so schaltet Director automatisch um.
#high	(Standard) Der Flash-Film wird immer in hoher Qualität, also mit Anti-Aliasing abgespielt.
#low	Der Film wird nur in niedriger Qualität abgespielt.

Beispiel:

```
sprite(1).quality=#autohigh
```

the scalemode

Diese Eigenschaftsvariable entspricht dem Eintrag SCALE aus dem Flash Asset Xtra und bestimmt, wie ein Flash-Sprite oder Vektordarsteller innerhalb seines Sprite-Rechtecks skaliert wird.

#showall	(Standard für Director-Filme vor Version 7) Behält die Proportionen des Ursprungsdarstellers bei. Falls nötig werden horizontale oder vertikale Spalten mit der Hintergrundfarbe aufgefüllt.
#noborder	Behält die Proportionen des Films bei und stutzt bei Bedarf vertikale oder horizontale Anteile des Films in Bezug auf das Sprite-Rechteck.
#exactfit	Berücksichtigt nicht die Proportionen des Flash-Films sondern skaliert den Flash-Film genau auf die Abmessungen des Sprite-Rechtecks.
#noscale	Zeigt den Flash-Film in der ursprünglichen Größe an, egal wie groß das Sprite auf der Bühne ist. Falls das Sprite-Rechteck kleiner ist als der Film, wird dieser beschnitten dargestellt.
#autosize	(Standardwert für Director 7) Das Sprite-Rechteck wird automatisch skaliert und unter Berücksichtigung der Befehle <code>rotation</code> , <code>skew</code> , <code>flipH</code> und <code>flipV</code> positioniert. Das bedeutet, daß ein Flash-Film bei der Rotation nicht wie in früheren Versionen von Director beschnitten wird. <code>#autosize</code> funktioniert nur, wenn <code>scale</code> , <code>viewScale</code> , <code>originPoint</code> und <code>viewPoint</code> Standardwerte tragen.

Beispiel:

```
sprite(1).scaleMode=#exactfit
```

the sound

Mit Hilfe dieser Variablen kann ermittelt werden, ob der derzeitige Flash-Film Sound abspielen soll oder nicht. Mögliche Werte sind **true/false**. Bei **false** wird kein Sound aus dem Flash-Film abgespielt. Diese Eigenschaftsvariable entspricht der Checkbox SOUND im Flash Asset Xtra.

the static

Eine Eigenschaftsvariable eines Darstellers oder Sprites, die Auskunft darüber gibt, ob der Flash-Film statisch ist (d. h. nur ein Bild besitzt) oder eine Animation beinhaltet. Außerdem kann man darüber die Abspielperformance eines Flash-Films kontrollieren. Sollte der Film nur ein Bild haben (statisch), so wird das Sprite nicht bei jedem Bild neu gezeichnet, sondern nur bei Bewegung oder Skalierung des Sprites in Director. Mögliche Werte sind **true/false**, standardmäßig steht **the static** auf **false**, was bedeutet, daß ein Flash-Film immer Animationen hat.

The static sollte auch nur auf **true** gesetzt werden, wenn sich der Film mit keinem anderen Sprite überschneidet, da sonst unter Umständen die Darstellung nicht sauber erfolgt.

Die Variable kann sowohl für Darsteller als auch für Sprites gesetzt werden. Des weiteren gibt sie auch zunächst immer Auskunft über den Flash-Film, denn solange sie nicht durch eine Wertzuweisung manipuliert wurde, trägt sie die Art des Films in sich (statisch oder nicht).

Beispiel:

```
If sprite(1).static then
-- sprite(1) ist ein statischer Film
else
-- sprite (1) beinhaltet eine Animation
end if
```

Viewpoint, ViewH, ViewV

Mit Hilfe dieser Eigenschaftsvariablen kann man den Versatz eines Flash-Films innerhalb seines Sprite-Rechtecks bestimmen. Die Variable **the Viewpoint** nimmt dabei als Wert einen Punkt entgegen, der relativ zum Registrierpunkt des Flash-Films interpretiert wird. Alternativ können seine horizontalen und vertikalen Werte auch per **viewH** und **viewV** eingegeben werden.

Beispiel:

```
sprite(1).viewpoint=point (50,50)
```

the viewscale

Diese Eigenschaftsvariable eines Darstellers oder Sprites entspricht dem Eintrag SCALE im weiter oben abgebildeten Flash Asset Xtra. Ihr Wert kann als Zoom verstanden werden, d.h. er zeigt, wie groß der Flash-Film im Sprite-Rechteck dargestellt wird. Der Standardwert liegt bei 100%.

Diese Eigenschaft muß den Standardwert 100% haben, falls **the scaleMode** auf den Wert **#autosize** gesetzt wurde, weil sonst das Sprite eventuell nicht richtig dargestellt wird.

Beispiel:

```
sprite(1).viewscale=200
```

Lingo-Befehle zum Streaming von Flash-Filmen

Flash-Filme können wie Shockwave-Filme während des Abspielens auch in Director bzw. Projektoren oder Shockwave-Filmen gestreamt werden. Folgende Lingo-Befehle erleichtern den Umgang mit dem Streaming von Flash-Filmen.

the bufferSize

Die ist eine Eigenschaftsvariable für Flash-Darsteller. Sie beschreibt, wie viele Bytes eines verlinkten Flash-Films auf einmal in den Speicher eingelesen werden können. Der Standardwert beträgt 32768 Bytes (= 32 KB). Dieser Eigenschaftswert tritt nur dann in Kraft, wenn das Vorausladen (preload) für den betroffenen Darsteller ausgestellt ist.

Beispiel:

```
member("FlashFilm").bufferSize=65535
```

the bytesStreamed

Diese nur lesbare Eigenschaftsvariable eines Flash-Darstellers gibt einen Wert zurück, der die Anzahl bereits eingelesener Bytes der Datei beschreibt. Dies funktioniert nur, wenn der Film läuft.

Beispiel:

```
on exitframe
put member("FlashFilm").bytesstreamed
end
```

clearError

Mit diesem Befehl kann ein Flash-Darsteller nach dem Auftreten eines Fehlers beim Einlesen wieder zurückgesetzt werden. Wenn ein Fehler beim Einlesen eines externen Darstellers auftritt, so ändert sich der Wert der Variablen `state` zu `-1`. Mit Hilfe der `GetError()`-Funktion kann man dann ermitteln, was der Fehler war und nach Behebung der Fehlerursache (z. B. der Angabe einer neuen URL) ein `clearError` ausgeben, um den Wert der Eigenschaftsvariable `the state` wieder zurückzusetzen.

Beispiel:

```
if member("Flashfilm").state=-1 then
member("Flashfilm").URL="http://www.neuerServer.de"
clearError(member "Flashfilm")
end if
```

frameReady()

`FrameReady` ist eine Funktion, die anzeigt, ob alle Darsteller eines bestimmten Bildes im Drehbuch schon geladen sind. Dies lässt sich mit der Angabe eines Flash-Sprites auch für Bilder innerhalb eines Flash-Films ermitteln.

Beispiel:

```
If frameready(sprite(2)) then
go the frame +1
end if
```

getError()

Diese Funktion arbeitet ebenso mit Flash- wie auch mit Shockwave Audio-Darstellern und liefert eine Fehlerbeschreibung bei Netzwerkproblemen während des Downloads. Die Funktion liefert für Flash-Darsteller folgende Werte zurück:

<code>false</code>	Kein Fehler ist aufgetreten
<code>#memory</code>	Es gibt nicht genug Speicher, um den Darsteller zu laden
<code>#filenotfound</code>	Die Datei konnte nicht gefunden werden
<code>#network</code>	Ein Netzwerkfehler verhinderte das Laden des Darstellers
<code>#fileformat</code>	Die Datei wurde gefunden, hat aber ein anderes Format oder es trat ein Fehler beim Einlesen der Datei auf
<code>#other</code>	Ein nicht näher beschreibbarer Fehler ist aufgetreten

Wenn während des Streamings eines Flash-Darstellers in den Speicher ein Fehler auftritt, wechselt die Eigenschaftsvariable **"the state"** ihren Wert nach `-1`. Dann kann die `getNetError()` Funktion aufgerufen werden, um die Art des Fehlers festzustellen.

the Linked

Eine Eigenschaftsvariable, die bestimmt, ob ein Flash-Film extern gelinkt (ergibt den Wert `true`) oder in der Besetzung eingebettet ist (Wert `false`). Der Wert dieser Eigenschaft kann ausgelesen und gesetzt werden:

Beispiel:

```
put Member("flashfilm").linked
```

the percentStreamed

Eine Eigenschaftsvariable, die nur ausgelesen werden kann und die eine Prozentangabe bezüglich des Ladezustandes des Flash-Films liefert.

Beispiel:

```
on exitframe me
member("info").text=member("flashfilm").percentStreamed&"%"
end
```

the state

The `state` gibt den Wert des Zustands des Darstellers in bezug auf das Streaming zurück. Der Wert kann nur ausgelesen werden.

Für Flash-Darsteller sind folgende Rückgabewerte möglich:

0	Darsteller ist nicht im Speicher
1	Der Header (Kopfteil) der Datei wird geladen
2	Der Header wurde eingelesen
3	Die Mediadaten des Darstellers werden eingelesen
4	Die Mediadaten wurden eingelesen
-1	Ein Fehler ist aufgetreten (siehe auch die <code>getError()</code> -Funktion)

Stream()

Eine neue Lingo-Funktion, mit der man manuell einen Anteil von einem Flash-Film in den Speicher einlesen kann. Die genaue Syntax lautet

```
stream (member strNAME, intANZAHL)
```

Dabei ist `strName` der Name des Darstellers und `intAnzahl` die Byte-menge, die eingelesen werden soll. Wird diese nicht angegeben, so liest Director die per `bufferSize` angegebene Bytemenge ein.

Die Funktion gibt auch die Anzahl bereits gestreamter Bytes eines Flash-Films zurück.

Beispiel:

```
stream (member("FlashFilm"), 65535)
```

the streamMode

Mit diese Eigenschaftsvariablen kann das Streamingverhalten von Flash-Filmen in Director beeinflusst werden. Folgende Werte sind dafür möglich:

<code>#frame</code>	Standard Bei jedem Bilddurchlauf in Director, bei dem sich der Flash-Film auf der Bühne befindet, wird ein Anteil des Films (angegeben mit <code>the bufferSize</code>) in den Speicher geladen
<code>#idle</code>	Liest Teile des Flash-Films bei jeder Generierung eines »idle«-Ereignisses in Director ein, mindestens jedoch einmal pro Bilddurchlauf.
<code>#manual</code>	Liest nur dann Teile des Flash-Films ein, wenn der Stream-Befehl für den entsprechenden Darsteller gegeben wird.

Beispiel:

```
member("flashfilm").streammode=#idle
```

the streamSize

Diese Eigenschaftsvariable beinhaltet die Gesamtgröße des einzulesenden Flash-Films in Bytes.

Beispiel:

```
put member(1).streamsize  
-- 305569
```

the URL

Dies beschreibt die URL oder die Pfadangabe zum Flash-Darsteller und kann ausgelesen und gesetzt werden. Für Flash-Darsteller ist dieser Wert synonym mit **the pathname**.

Beispiel:

```
member("Flashfilm").url="http://www.macromedia.com/uber.swf"
```

Lingo-Befehle zur Interaktion von Flash-Filmen in Director

Flash-Filme sind eigene, in sich gekapselte interaktive Container, die in Director auf verschiedene Art und Weise mit der Bühne interagieren können. Auch dafür gibt es eine ganze Reihe von neuen Lingo-Befehlen.

the broadcastProps

Änderungen an Vektor- oder Flash-Darstellern müssen nicht zu einer sofortigen Änderung **aller** davon abgeleiteten Sprites auf der Bühne führen. Der Schlüssel dazu ist diese Eigenschaftsvariable, die die Werte **true** oder **false** annehmen kann. Ist ihr Wert **true**, so werden Änderungen an den Darstellern sofort auf die Sprites übertragen (Standardwert). Bei **false** werden Änderungen an den Darstellern nur bei der Erzeugung neuer Sprites berücksichtigt, die bereits auf der Bühne stehenden Sprites werden davon nicht betroffen.

Beispiel:

```
member("Flashfilm").broadcastProps=false
```

the centerRegPoint

Diese Eigenschaftsvariable kontrolliert, ob der Registrierungspunkt für einen Flash-Film automatisch beim Skalieren des Films angepaßt werden soll (**true**, Standardwert) oder der aktuelle Punktwert trotz eventueller Transformationen am Film beibehalten werden soll.

Beispiel:

```
member("Flashfilm").centerregpoint=false
```

the clickMode

Eine wichtige Eigenschaftsvariable für Darsteller und Sprites, die Director anweist, wie mit Mausereignissen umzugehen ist. Betroffen davon sind `mousetenter`, `mousewithin` und `mouseleave`.

Mögliche Werte sind:

#boundingbox	Mausklicks werden überall im und RollOver-Ereignisse an den Rändern des Sprite-Rechtecks registriert.
#opaque	Standard Mausklick- und Rollover-Ereignisse werden nur wahrgenommen, wenn der Mauszeiger dabei auf einer nicht transparenten Fläche im Flash-Film war und der Einkopiermodus auf HINTERGRUND TRANSPARENT steht. Sollte ein anderer Einkopiermodus gewählt worden sein, so verhält sich dieser Wert wie die obige #boundingbox .
#object	Entdeckt Mausclicks, wenn der Mauszeiger auf einer gefüllten Fläche im Flash-Film steht (Nicht-Hintergrundfarbe der Films) und erzeugt Rollover-Ereignisse für gefüllte Flächen, gleichgültig welcher Einkopiermodus für das Sprite gewählt wurde.

Beispiel:

```
if sprite(1).clickmode=#opaque then  
sprite(1).clickmode=#object  
end if
```

Dieses kurze Lingo-Fragment läßt einen Flash-Film nur Mausereignisse in den gefüllten Flächen des Flash-Films erzeugen.

the DefaultRectMode

Diese Eigenschaftsvariable eines Flash-Darstellers legt fest, wie groß die vom Darsteller abgeleiteten Sprites auf der Bühne werden sollen. Möglich Werte sind `#flash` (Standard) und `#fixed`.

Bei `#flash` wird die Größe der Sprites (das sogenannte `defaultrect`) an die Größe des Flash-Films angepaßt. Wenn also ein Flash-Film in 320 x 200 Pixel angelegt wurde, so ist jedes davon abgeleitete Sprite zunächst 320 x 200 Pixel groß.

Beim Wert `#fixed` hingegen wird der Wert der Eigenschaftsvariablen `defaultRect` bei der Erstellung neuer Sprites berücksichtigt.

Beispiel:

```
member("Flashfilm").defaultrectmode=#fixed  
member("Flashfilm").flashrect=rect (0,0,160,120)
```

the defaultRect

Eine Darstellereigenschaftsvariable, die die Standardgröße neuer, vom Flash-Darsteller abgeleiteter Sprites beschreibt, wenn dieser `defaultrectmode=#fixed` darauf eingestellt wurde.

Die neue angegebene Größe betrifft auch alle Sprites auf der Bühne, die bisher nicht in irgendeiner Art und Weise transformiert worden sind. Der Wert für die Variable wird in der Form eines Rechtecks (`rect`) übergeben:

```
member("Flashfilm").defaultrect=rect (0,0,120,120)
```

Ab sofort haben alle neuen, vom Darsteller »Flashfilm« abgeleiteten Sprites die Standardgröße 120 mal 120.

the Playing

Eine Eigenschaftsvariable eines Sprites, die per `true/false` mitteilt, ob der Flash-Film spielt oder nicht. Diese Variable kann nur ausgelesen werden.

Beispiel:

```
If Sprite(1).playing then put "Flash-Film läuft..."
```

the directToStage

Die Sprite- und Darstellereigenschaft entspricht der Checkbox DIRECT TO STAGE im weiter oben abgebildeten Flash Asset Xtra. Im Falle von **true** wird der Darsteller oder das Sprite unter Umgehung des Bildschirmpuffers von Director direkt in den Bildschirmspeicher geschrieben. Dies verbessert die Abspielleistung, erlaubt aber kein Plazieren mehr von anderen Sprites über das betreffende Objekt. Auch die Einkopiermodi werden nicht mehr berücksichtigt, der Film wird immer »kopiert«. Die Eigenschaftsvariable kann gesetzt und ausgelesen werden.

Beispiel:

```
member("Flashfilm").directtostage=false  
sprite(1).directtostage=true
```

Obiges Beispiel erzeugt einen Darsteller ohne **DirectToStage**, aber das von ihm abgeleitete Sprite im Kanal 1 wird dennoch direkt in die Bühne gemalt, was die Darstellung des Films verbessert.

the eventPassMode

Eine Eigenschaftsvariable für Flash-Darsteller und Sprites gleichermaßen. Sie kann verschiedene symbolische Werte annehmen und beschreibt die Weiterleitung von Mausclicks innerhalb des Flash-Films oder Director. Da Flash-Filme selbst Interaktivität beinhalten und somit direkt auf Mausereignisse reagieren können (z. B. einen Klick), kann man als Entwickler mit dieser Variablen Einfluß darauf nehmen, ob, und wenn ja, wie diese Ereignisse an Director weitergeleitet werden, um eventuell an das Sprite angefügte Skripte zusätzlich auszuführen. Der Wert dieser Variablen kann ausgelesen und gesetzt werden.

#passAlways	Standard Mausereignisse werden vom Flash-Film an Director weitergeleitet.
#passButton	Mausereignisse werden nur weitergeleitet, wenn im Flash-Film eine Schaltfläche (Button) geklickt wurde.
#passNot-Button	Mausereignisse werden nur weitergeleitet, wenn im Flash-Film nicht eine Schaltfläche (=jedes beliebige andere Objekt) geklickt wurde.
#passNever	Es werden keine Mausereignisse vom Flash-Film an Director weitergeleitet.

Beispiel:

```
member("Flashfilm").eventPassMode=#passNotButton
```

Mit dem obigen Beispiel werden nur die Mausereignisse an Director weitergeleitet, die nicht im Zusammenhang mit einer Schaltfläche (Button) in dem Flash-Film stehen. Jene werden vom Flash-Film selbst bearbeitet.

findLabel()

In Flash-Filmen kann man analog zum Drehbuch in Director sogenannte Marker angeben. Mit der Funktion `findLabel()` kann man sich die Bildnummer im Flash-Film zu einem bestimmten Label zurückgeben lassen. Eine Null als Ergebnis bedeutet, daß der Flash-Film keinen entsprechenden Labelnamen trägt oder daß der entsprechende Teil des Films beim Streaming noch nicht in den Speicher eingelesen wurde.

Beispiel:

```
put findLabel(sprite(1), "Start")  
-- 10
```

the fixedRate

Dies beschreibt die Abspielgeschwindigkeit eines Flash-Darstellers oder Sprites, wenn der Wert von `the playbackmode` auf `#fixed` steht. Sollte dies nicht der Fall sein, so wird dieser Wert ignoriert.

Beispiel:

```
sprite(me.spritenum).member.playbackmode=#fixed  
sprite(me.spritenum).fixedrate=10
```

the FlashRect

Als eine nur lesbare Darsteller-Eigenschaftsvariable beschreibt sie die Ursprungsgröße des Flash-Films oder der Vektorgrafik. Ihr Ergebnis ist ein Rechteck. Bei gelinkten Flash-Filmen werden nur richtige Ergebnisse zurückgegeben, wenn der Film bereits in den Speicher gelesen wurde.

Beispiel:

```
put member(1).flashrect  
-- rect(0, 0, 550, 400)
```

flashToStage()

Eine Funktion, die eine Koordinatenangabe in einem Flash-Sprite in den entsprechenden Wert für die Bühne übersetzt. Das Ergebnis ist eine Punktangabe für die Bühne.

Die X- und Y-Werte eines Punkts in einem Flash-Film beziehen sich immer auf die Ursprungsgröße, mit der ein Flash-Film erstellt wurde. Die linke obere Ecke ist dabei wie in Director `point(0,0)`. Da Flash-Filme skaliert, rotiert und gestaucht werden können, stimmt ihr Raster oft nicht mehr mit dem der Director-Bühne überein. Diese Funktion hilft dann dabei, den entsprechenden Punkt vom Flash-Film auf die Bühne zu übertragen. Siehe auch die Funktion `StageToFlash()`.

Beispiel:

```
put flashtostage (sprite(1), point(50,50))  
-- point(110, 81)
```

Der ursprüngliche Punkt `point(50,50)` im Flash-Film entspricht im Moment dem Punkt `point(110,81)` auf der Director-Bühne.

The frame

Eine Eigenschaftsvariable für Sprites, die das aktuelle Bild des Flash-Films als Wert innehat. Standardwert ist 1, der Wert kann gesetzt und ausgelesen werden. Im Zusammenhang mit dem Wert von `FrameCount` kann damit festgestellt werden, ob ein Flash-Film zu Ende gespielt ist.

Beispiel:

```
sprite(1).frame=40
```

Obiges Beispiel setzt das Bild 40 im Sprite 1.

the frameCount

Diese Eigenschaftsvariable für Flash-Darsteller enthält die Gesamtzahl an Bildern in einem Flash-Film und kann nur ausgelesen werden. Im Zusammenhang mit der obigen `frame`-Variablen kann man damit feststellen, ob ein Flash-Film zu Ende gespielt ist.

Beispiel:

```
on exitframe me
if sprite(spritenum).frame=sprite(spritenum).framecount then
    put "Flash-Film ist zuende."
end if
end
```

the frameRate

Nur lesbare Eigenschaftsvariable eines Flash-Darstellers, die die ursprüngliche Abspielgeschwindigkeit angibt.

Beispiel:

```
put member("Flash").framerate
-- 20
```

frameReady()

Eine normale Lingo-Funktion, die mit einer modifizierten Syntax auch für Flash-Sprites genutzt werden kann:

```
frameready (sprite (FlashSprite), Bildnummer)
```

Dies ermittelt, ob die Bildnummer des Flash-Films schon geladen ist oder nicht. Die Funktion ergibt dann **true**, ansonsten **false**.

Beispiel:

```
on exitframe me
if NOT frameready(sprite(1), 100) then
go to the frame
end if
end
```

getFrameLabel()

Eine Funktion, die den Markernamen (**frameLabel**) für das Bild im angegebenen Sprite ermittelt. Falls das Bild nicht existiert oder der Anteil noch nicht im Speicher ist, ergibt dieser Wert einen leeren String ("").

Beispiel:

```
put getFramelabel(sprite(1), 150)
-- "Start"
```

goToFrame

Der Befehl, der einen Flash-Film ab einem bestimmten Bild abspielen läßt. Die Angaben für das Bild können entweder ein String für seinen Markernamen oder die Bildnummer sein. Dieser Befehl hat denselben Effekt wie die Definition des Werts der Sprite-Eigenschaft **frame**.

Beispiel:

```
GoToFrame (sprite(1), "Start")
```

HitTest()

Eine Funktion, die überprüfen kann, welcher Anteil eines Flash-Films sich über einem bestimmten Punkt der Bühne befindet. Mögliche Rückgabewert sind:

#background	Der angegebene Punkt liegt in einer Zone mit der Hintergrundfarbe des Flash-Films
#normal	Über dem Punkt liegt ein Objekt des Flash-Films
#button	Der angegebene Punkt liegt in der Zone einer aktiven Schaltfläche (Button) im Flash-Film
Ab dem Director 7.02-Update zusätzlich noch möglich:	
#editText	Über dem angegebenen Punkt auf der Bühne liegt ein editierbares Textfeld im Flash-Film

Folgendes Beispiel prüft, über welchem Anteil eines Flash-Sprites der Mauszeiger gerade liegt:

```
on exitframe
put hittest(sprite (1), the mousetloc)
end
```

the mouseoverbutton

Eine nur lesbare Eigenschaft für Flash-Sprites, die anzeigt, ob der Mauszeiger sich gerade über einer aktiven Schaltfläche (**button**) befindet. In diesem Fall ergibt diese Eigenschaft **true**, ansonsten **false**.

Beispiel:

```
on exitframe me
if sprite(me.spriteenum).mouseoverbutton then puppetsound "Klick"
end
```

the obeyScoreRotation

Die Eigenschaft für Flash-Darsteller, die bestimmt, ob die abgeleiteten Sprites mit Hilfe des seit Director 7 neuen Rotationswerkzeugs rotiert werden dürfen (**true**, Standardwert für neue Filme) oder nicht (**false**).

Diese Eigenschaft wird automatisch auf **false** gesetzt für Darsteller in Director-Filmen vor der Version 7 (faktisch nur für Filme mit Director 6.5). Diese Darsteller konnten nur mittels einer Eigenschaftsvariablen rotiert werden.

Wenn auf **true** gesetzt, werden diese alten Anweisungen ignoriert. Diese Eigenschaft kann ausgelesen und gesetzt werden.

Beispiel:

```
member("Flash").obeyScoreRotation=false
```

Dank der obigen Zeile können ab sofort keine vom Darsteller »Flash« abgeleitete Sprites mehr rotiert werden.

rewind sprite NUMMER

Ein Befehl, der den Film eines Flash-Sprites wieder an den Anfang (Bildnummer 1) zurückspult, unabhängig davon, ob der Film gerade gestoppt ist oder läuft.

Beispiel:

```
rewind Sprite(1)
```

the rotation

Neue Darsteller- und Spriteeigenschaft, die den Rotationswert für das Objekt beinhaltet. Durch die Zuweisung eines neuen Wertes kann der Flash-Film innerhalb des Sprite-Rechtecks rotiert werden, ohne das Sprite als solches zu verändern. Dies funktioniert für Flash-Sprites nur, wenn **obeyScoreRotation** auf **true** steht (Standardwert, siehe oben).

Die Rotation erfolgt im Falle eines Flash-Films um seinen Ursprungspunkt («**originpoint**«).

Der Wert wird in Grad angegeben (360 = eine volle Drehung) und kann auch als Fließkommazahl übergeben werden. Das Drehbuch kann Rotationen von +21,474,836.47° bis -21,474,836.48° durchführen und speichern. Dies ergibt 59652 volle Rotationen.

Standardwert ist 0, diese Eigenschaft kann ausgelesen und gesetzt werden.

Beispiel:

```
sprite(me.spritenum).rotation=180
```

the scale

Eine Sprite- und Darstellereigenschaft, die unter anderem auch das Skalieren von Flash-Filmen in Prozent kontrolliert. Der Rückgabewert ist eine Fließkommazahl, standardmäßig trägt sie den Wert 100.0000 (= 100%). Bei der Änderung des Wertes für diese Eigenschaft wird der Flash-Film ausgehend von seinem Ursprungspunkt (**originpoint**) skaliert, wie per **originMode** angegeben.

Beispiel:

```
sprite(me.spritenum).scale=200
```

showProps member DARSTELLER

Zeigt eine Liste von Eigenschaften (nur in Director) für einen Flash-Darsteller im Nachrichtenfenster und funktioniert nur in Director selbst.

◀ 246

Siehe auch die Liste am Anfang dieses Kapitels.

Beispiel:

```
showProps member("Flashfilm")
```

the sound/MixMedia

Seit Director 7.02 gibt es diese globale Eigenschaftsvariable, mit der man festlegen kann, ob Flash-Filme ihren Sound mit dem in Director mischen sollen (**true**, Standardwert), oder ob sie getrennt abgespielt werden sollen (**false**). Diese Eigenschaft ist standardmäßig **true** für Director 7-Filme



und `false` für frühere Versionen (faktisch nur für Director 6.5-Filme, bzw. Director-Filme mit dem früheren Flash Asset Xtra).

stageToFlash()

Mit dieser Funktion kann ein Punkt auf der Director-Bühne in einen Punkt des Flash-Films übersetzt werden. Der Rückgabewert ist ein Punkt (z. B. `point(25,20)`). Für Flash- wie auch für Director-Filme ist der Punkt `point(0,0)` immer die linke obere Ecke. Dies ist die umgekehrte Funktion zu `FlashToStage()` - siehe auch weiter oben.

Beispiel:

```
put stagetoFlash(sprite(1), point (300,300))  
-- point(181, 192)
```

Der Punkt (300,300) auf der Director-Bühne entspricht somit dem Punkt 181,192 im Flash-Film. Im Zusammenhang mit der `hittest()`-Funktion lassen sich damit einfache Labyrinth-Spiele realisieren. Sehen Sie dazu auch weiter unten das Beispiel.

stop sprite NUMMER

Ein Befehl für Sprites, der das Abspielen des Flash-Films anhält.

Beispiel:

```
stop sprite (1)
```

Neue Befehle aus dem Director 7.02 Update

Mit dem Director 7.02 Update kamen auch Änderungen in Bezug auf das Flash Xtra. Es wurden zwei neue Befehle zur Programmierung der Variableninhalten in Flash 4 implementiert sowie der bereits bestehende Befehl `hittest` erweitert.

getVariable(sprite X , VariablenName)

Dieser neue Lingo-Befehl ergibt den aktuellen Wert der angegebenen Variablen im Flash 4 Sprite, allerdings ist das Ergebnis immer ein String.

setVariable(sprite X , VariablenName , neuerWert)

Weist der Variablen im Flash 4-Sprite den angegebenen neuen Wert zu. Dieser neue Wert muß ebenfalls immer ein String sein.

hitTest(sprite X , PunktaufBühne)

Der Befehl `hitTest` prüfte bisher nur, was für ein Flash-Objekt sich über dem angegebenen Punkt der Bühne befand. Mögliche Ergebnisse waren bisher

`#background`

`#normal`

`#button`



Nun ist noch das Symbol

`#editText`

hinzugekommen, das beschreibt, daß sich über dem angegebenen Punkt auf der Bühne im Flash-Film ein editierbares Textfeld befindet.

Neue Flash-Befehle in Director 8

Außer zwei neuen Befehlen, mit denen man als Programmierer Zugriff auf Eigenschaften eines Flash-Films erhält, hat Director 8 hat in punkto Flash nicht sehr viel Neues zu bieten. Diese Befehle sind `GetFlashPropert()` und `setFlashProperty()`.

GetFlashProperty()

Eine Funktion, die den Zugriff auf interne Eigenschaftsvariablen eines Flash-Films gestattet. Sie entspricht dem Aktion Script-Befehl `getProperty()` in der Flash-Programmierung, mit dem sich Eigenschaften eines Filmclips auslesen lassen. Da Flash-Filme in Filme geladen werden können, kann man mit diesem Befehl auch Eigenschaftswerte von Filmen in Flash-Filmen auslesen. Diese müssen dann in der richtigen Hierarchie für den Parameter »Ziel« eingegeben werden:

```
Sprite(1).getFlashProperty("Ziel", #posX)
```

Mit dem oben angegebenen Beispiel können Sie etwa die X-Position des eingebetteten Filmclips oder der Ebene »Ziel« innerhalb der Flash-Präsentation erfahren. Der zurückgegebene Wert bezieht sich dabei aber auf die Werte für die Flash-Präsentation und hat nichts mit den Werten des aktuellen Sprites in Director zu tun.

Der zweite Parameter neben dem Ziel ist ein Symbol für die Eigenschaft, die abgerufen werden soll. Auf diese Weise lassen sich folgende Filmclipeigenschaften abfragen:

```
#posX, #posY, #scaleX, #scaleY, #visible, #rotate, #alpha, #name,  
#width, #height, #target, #url, #dropTarget, #totalFrames, #current-  
Frame und #lastframeLoaded
```

Diese Eigenschaftswerte sind Flash-spezifisch und können deswegen an dieser Stelle nicht weiter erläutert werden. Sie entsprechen in ihrer Funktion den Eigenschaften eines Sprites in Director und beschreiben unter anderem die Position und Rotation eines »Sprites« im Flash-Film. Eine ausführlichere Beschreibung dieser Eigenschaften finden Sie in Ihrer Flash-Dokumentation.

Wenn Sie eine globale Eigenschaft des Flash-Sprites abrufen möchten, übergeben Sie einen leeren String als Wert für »Ziel«. Auf diese Weise lassen sich `#focusRect` und `#spriteSoundBufferTime` für den gesamten Flash-Film abrufen.

SetFlashProperty()

Der neue Befehl `SetFlashProperty()` ermöglicht Ihnen den Aufruf der Funktion `setProperty()` in einem Flash-Film. Damit erhalten Sie aus Director heraus die Möglichkeit des Zugriffs auf verschaltete Ebenen oder eingebettete Filme in einem Flash-Film.

```
Sprite(1).setFlashProperty("Ziel", #rotate, 180)
```

Beim Parameter »Ziel« handelt es sich um den Namen der Ebene oder des Unterfilms in Flash, dessen Eigenschaft im Hauptfilm verändert werden soll. So wird etwa mit dem obigen Beispiel die Ebene oder der eingebettete Film »Ziel« um 180 Grad im Hauptfilm gedreht. Sie erhalten also somit die Möglichkeit, auf die Abspielparameter des Flash-Films zuzugreifen und sie zur Laufzeit dynamisch umzuprogrammieren.

Der zweite Parameter, im obigen Beispiel das Symbol `#rotate`, beschreibt, welche Eigenschaft der zuvor angegebenen Ebene oder des Unterfilms im Hauptfilm geändert werden soll. Mögliche Werte sind:

`#posX`, `#posY`, `#scaleX`, `#scaleY`, `#visible`, `#rotate`, `#alpha` und `#name`.

Um die globalen Flash-Film-Eigenschaften `#focusRect` und `#spriteSoundBufferTime` des Hauptfilms umzuprogrammieren, übergeben sie als ersten Parameter einen leeren String ("") an die Anweisung:

```
Sprite(1).setFlashProperty("", #Soundbuffertime, 10)
```

Flash und Director Hand in Hand

Flash und Director besitzen mehrere Möglichkeiten, miteinander zu kommunizieren. Neben den obigen Lingo-Befehlen, die von Director aus fast jeden Aspekt eines Flash-Films in bezug auf seine Integration kontrollieren können, gibt es zusätzlich noch die direkte Kommunikation. So kann etwa der Flash-Film Lingo-Befehle an Director senden sowie ab Flash 4 und dem Update Director7.02 auch den Inhalt seiner Variablen mit Director abgleichen. Dazu im folgenden ein paar Beispiele.

Direkte Kommunikation zwischen Flash und Director

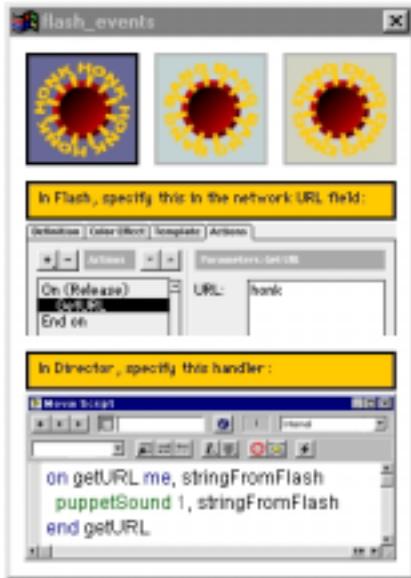
Insgesamt gibt es für einen Flash-Film drei Möglichkeiten, sich in Director mitzuteilen. Alle drei funktionieren über den Skriptbefehl `GetURL` in Flash (Abb. 4)

Informationsangabe über den GetURL-Parameter

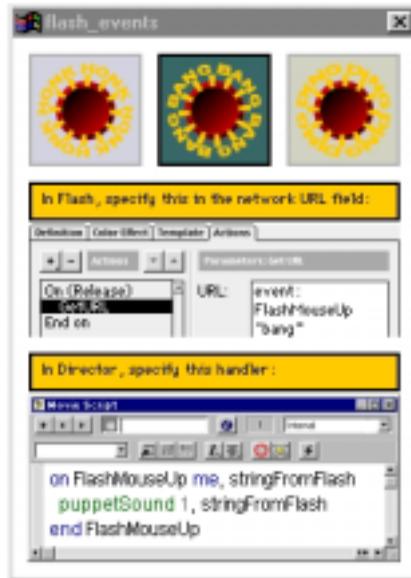
Die erste Möglichkeit zur Mitteilung besteht darin, daß der Flash-Film als URL für die Aktion `GetURL` eine Zeichenkette überreicht bekommt, die als Parameter in Director beim Aufruf übernommen wird.

In Flash:

```
on (release)
getURL "Eine Zeichenkette"
end on
```



▲ **Abbildung 4**
 Der Director-Beispielfilm flash_events veranschaulicht sehr schön die Möglichkeiten der Mitteilung aus Flash-Filmen



▲ **Abbildung 5**
 Flash kann auch Ereignisse in Director erzeugen.

Die Aktion `getURL` wird in Director als Systemereignis abgefangen und kann ausgewertet werden.

In Director kann deshalb folgender Handler geschrieben werden:

```
on getURL Zeichenkette
  put Zeichenkette
end
```

Als Parameter können natürlich auch ab Flash 4 Variablen bzw. somit ihre Werte übergeben werden.

Direkte Ereigniserzeugung durch GetURL

Eine etwas weiterführende Methode der Kommunikation des Flash-Films besteht darin, den Flash-Film selbst Ereignisse in Director erzeugen zu lassen (Abb. 5).

Dazu ist eine Erweiterung der URL-Angabe bei der `GetURL`-Aktion in Flash notwendig. Dem eigentlichen Kommando wird dazu die Kennung `event:` vorangestellt.

In Flash:

```
On (release)
  GetURL event:
  Click #leicht
End on
```

Dies erzeugt in Director unmittelbar das Ereignis »Click« mit dem Parameter #leicht, das durch einen entsprechenden Handler abgefangen werden kann.

In Director:

```
On Click parameter
Put parameter
End
```

Direkte Lingo-Aufrufe aus Flash

Die letzte Möglichkeit der Kommunikation besteht darin, daß der Flash-Film als Parameter für die GetURL-Aktion eine ganze Lingo-Zeile mitgeteilt bekommt. Diese wird in Director dann direkt ausgeführt (Abb. 6).

In der Praxis sieht das dann so aus, daß dem Parameter für die Aktion GetURL in Flash die Kennzeichnung **lingo:** vorangestellt wird, gefolgt von der Lingo-Zeile.

In Flash:

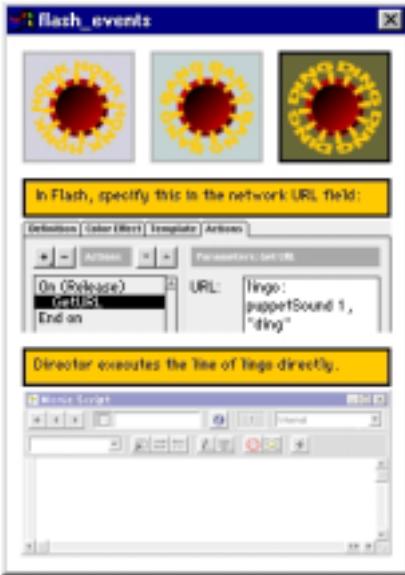
```
On (release)
Lingo:
Put "dies kommt direkt aus dem Flashfilm."
End on
```

Dies bedarf von Seiten Directors keiner weiteren Vorbereitungen, die Lingo-Zeile wird direkt ausgeführt.

Programmieren von Variablen in Flash (4-)Filmen aus Director

Seit dem Update Director 7.02 ist es möglich, aus Director-Filmen mit Hilfe der beiden neuen Lingo-Befehle

GetVariable und SetVariable



◀ **Abbildung 6**
Flash kann direkt Lingo-Anweisungen an Director geben

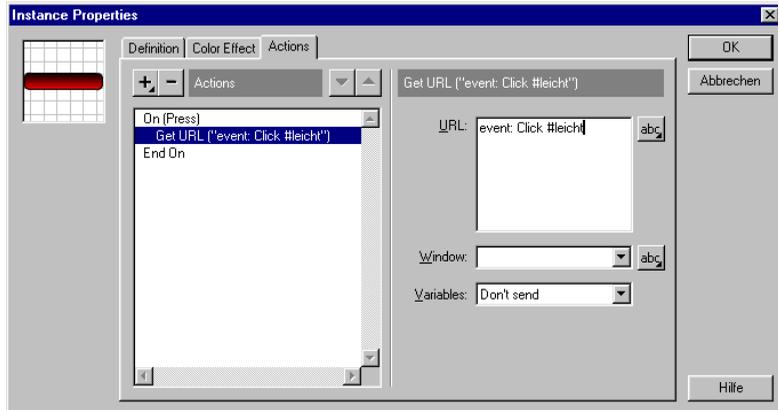
Variableninhalte in Flash 4-Filmen als Zeichenketten auszulesen und zu setzen. Dies ist sehr interessant, da sich damit zur Laufzeit Information in eine Flash-Präsentation hineinprogrammieren läßt. Die Flash-Datei muß dazu natürlich auch Variablen definiert haben, die die Werte annehmen können. Dies erfordert ein wenig Vorarbeit in Flash, weswegen hier an dieser Stelle leider nicht weiter darauf eingegangen werden kann. Eine gute Beschreibung zur Erzeugung von sogenannten Flash-Templates liefert das schon eingangs erwähnte Buch »Flash 4« von Sascha Wolter, ebenfalls im Verlag Galileo Press erschienen.

Ein Labyrinth-Spiel in Flash

Um die Nutzung der Kommunikation mittels der GetURL-Aktion sowie einiger neuer Lingo-Befehle zu demonstrieren, soll nun ein kleines Spiel mit Director und Flash entworfen werden. Es geht dabei natürlich weniger um die Ästhetik als mehr um die Demonstration der Möglichkeiten.

Zunächst einmal wird es ein Startmenü geben, auf dem der Spieler eine Schwierigkeitsstufe auswählen kann. Je nach Spielgrad steht dem

Abbildung 7 ►
Die Aktionsanweisung
für die erste Schaltfläche
für die leichte Spielstufe



Spieler mehr oder weniger Zeit zur Verfügung, um aus dem Labyrinth herauszukommen.

Das Menü wird den Schwierigkeitsgrad als Symbol per `getURL`-Aktion an Director mitteilen, der daraufhin eine interne Variable mit dem übergebenen Argument aus dem Flash-Film füllen wird. Dabei wird das Ereignis »Click« aus dem Flash-Film erzeugt werden, und als Parameter wird ein Symbol mit der Schwierigkeitsstufe übergeben werden.

Das Labyrinth selbst wird eine Vektorform sein, die mit den Funktionen `stageToFlash()` und `hitTest()` bei der Steuerung der Spielfigur abgefragt werden wird.

http://web.kyoto-inet.or.jp/people/eisaku/e_maze/

► Der Flash-Film
Der Flash-Film läßt sich in zwei Teile untergliedern: einmal das Startmenü und dann das Labyrinth selbst. Auf die Animationsfähigkeiten von Flash wird in diesem Beispiel weniger Wert gelegt.

Das Startmenü

Das Startmenü für das Labyrinthspiel beinhaltet drei Schaltflächen, die jeweils mit einer Aktion ausgestattet sind (Abb. 7).

Beim Klicken dieser Schaltfläche wird in Director das Ereignis »Click« ausgelöst, für das es folgendes Verhaltensskript am Flash-Sprite gibt:

In Director:

```
property spritenum
```

```
on Click me, par
  global gSpielstufe
  if NOT symbolp(par) then
    -- kein Symbol übergeben
    alert "Es wurde aus dem Flash-Film kein Symbol an Director
          übergeben!"
  halt
  -- Ausführung wird angehalten
else
  -- ein Symbol (#irgendwas) wurde übergeben.
  -- welches?
  case par of:
    -- Anpassung der Spielstufe je nach Symbol
    #leicht:
      gSpielstufe=120
    #mittel:
      gSpielstufe=90
    #schwer:
      gSpielstufe=60
  end case
end if
go "maze"
-- springe zum Marker "maze" im Director-Film
-- dies zeigt das Labyrinthbild an
end
```

Die globale Variable `gSpielstufe` enthält danach die Anzahl Sekunden für die Absolvierung des Parcours.

Die Markierung »Maze« im Drehbuch zeigt nun dasselbe Flash-Sprite, diesmal aber mit einem anderen Verhaltensskript:

```
property spritenum
```

```
on beginsprite me
  gotoFrame (sprite(spritenum), "maze")
  sprite(spritenum).static=true
end
```

Damit wird zunächst das Bild mit dem Marker »Maze« im Flash-Sprite eingestellt (`gotoFrame...`) und anschließend aus Performance-Gründen die statische Anzeige des Flash-Film angeschaltet, da sich nichts im Flash-Film bewegen soll. Lediglich die Umrisse werden abgefragt, um festzustellen, ob die Spielfigur in die entsprechende Richtung laufen darf. Das dazugehörige Verhaltensskript der Figur:

```
property spritenum

on exitframe me
  pos=sprite(spritenum).loc
  -- Zwischenspeichern der eigenen Position

  case the keycode of:
    -- Abfrage der Tastatur
    123:
      -- links
      pos.locH=pos.locH-2
    124:
      -- rechts
      pos.locH=pos.locH+2
    125:
      -- unten
      pos.locV=pos.locV+2
    126:
      -- oben
      pos.locV=pos.locV-2
  end case

  if hittest(sprite(2), pos)=#background then
    -- falls die Koordinate pos in einen
    -- Hintergrundbereich des Flash-Sprites fällt...
    sprite(spritenum).loc=pos
    -- wird die Position geändert (erlaubter Schritt)
    -- sonst nicht.
  end if
end
```

Damit wäre auch schon das Schlimmste erledigt. Es kommt noch ein Skript hinzu für den Timer, der die globale Variable `gSpielstufe` einmal

pro Sekunde herunterrechnet und im Falle des Zeitablaufs zum Marker »GameOver« springt, an dem der Spieler die Möglichkeit hat, noch einmal zu spielen:

```
property pZeit

on beginsprite me
  pZeit=the milliseconds
end

on exitframe me
  global gSpielstufe
  if the milliseconds>pZeit+1000 then
    gSpielstufe=gSpielstufe-1
    sprite(me.spriteum).member.text="Zeit:"&&gSpielstufe
    pZeit=the milliseconds
  end if
  if gSpielstufe<1 then
    go "GameOver"
  end if
end
```

Am Ausgang des Labyrinths wird dann noch ein kleines Sprite plaziert, das ständig prüft, ob es nicht vom Spielersprite berührt wird. Ist dies der Fall, bedeutet es, daß der Spieler den Weg aus dem Labyrinth herausgefunden hat. In diesem Falle erscheint eine entsprechende Meldung und es wird zum Hauptmenü zurückgesprungen, um eine neue Partie zu starten, und fertig ist das kleine Spiel:

```
property spritenum

on exitframe me
  if sprite(spritenum).intersects(3) then
    alert "Sie haben gewonnen!"&&return&"Herzlichen Glückwunsch!"
    go "Start"
  end if
end
```

Weiterführende Möglichkeiten in Flash

Das Zusammenspiel zwischen Director und Flash bietet mehrere interessante Ansatzpunkte. Zunächst wäre einmal die freie Skalierbarkeit von Flash-Animationen zu nennen, die immer geglättete Vollbildanimationen erlaubt. Dies ist ein Punkt, den Director bisher nicht alleine bieten kann. Im Zusammenspiel von Flash-Filmen mit der Lingo-Programmierung eröffnen sich damit neue Horizonte.

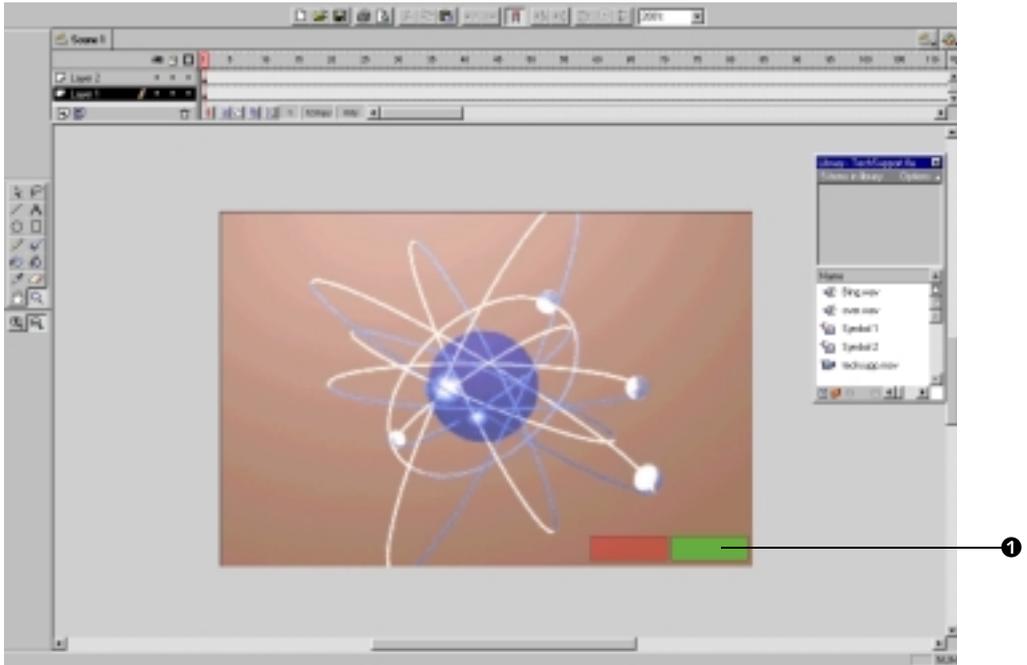
Ein weiterer Punkt ist die Möglichkeit des Streamings: Flash-Filme können auch in Director und Shockwave-Filmen gestreamt werden, was die potentiellen Anwendungsmöglichkeiten erweitert: eine komfortable, in Director programmierte Arbeitsoberfläche kann zum Beispiel je nach Auswahl des Benutzers animierte Flash-Animationen aus dem Internet laden. In diesem Zusammenspiel würde sich auch der Flash-Generator gut behaupten, um diese Animationen zur Laufzeit mit aktuellen Datenbankinhalten zu erstellen.

Ein weiterer sehr interessanter Punkt betrifft die beiden neuen 4er Versionen von Flash und Quicktime: in Flash 4 ist es möglich geworden, einen Flash-Track über ein Quicktime-Video zu legen. Der Flash-Track kann dann dank der Skriptingfähigkeiten von Flash das Quicktime-MOV steuern. Apple hat dazu auch den Flash-Player von Macromedia lizenziert und in seinen Quicktime-Player eingebaut. Da Director 7.02 Quicktime 4 unterstützt, kann man somit ganz einfach optisch ansprechende und multimediale Video-Anwendungen programmieren (Abb. 8).

Alle Möglichkeiten von Flash können dazu benutzt werden, inklusive Animationen, Sounds, Skripting und MP3-Kompression. Beim Exportieren erzeugt Flash 4 dann auf Wunsch einen Quicktime 4-Movie, der im Quicktime-Player die zusätzliche Flash-Spur samt ihrer Programmierung anzeigt (Abb. 9).

Auch diese Technik ermöglicht eine Reihe neuer Anwendungen. Denkbar sind zum Beispiel gestreamte Videokataloge mit der direkten Möglichkeit zur Anzeige weiterer Auskünfte auf Knopfdruck. Da Flash 4 auch neuerdings Web-Formulare unterstützt, könnten so auch zum Beispiel interaktive Fragebögen im Internet realisiert werden.

Neben dieser Verquickung von Quicktime und Flash wird die Umprogrammierung von Flash-Inhalten zur Laufzeit auch in Zukunft weiterhin immer wichtiger werden. Für das Internet hat Macromedia deswegen den Generator 2 herausgebracht und in Director 8 die beiden neuen Funktionen `getFlashProperty()` und `setFlashProperty()` hinzugefügt. Der Generator erlaubt das Ersetzen von Komponenten im Flash-Film zur



▲ **Abbildung 8**
 Ein Quicktime-Movie kann in Flash um eine interaktive Flash-Spur erweitert werden (hier die beiden Knöpfe rechts unten: ❶)

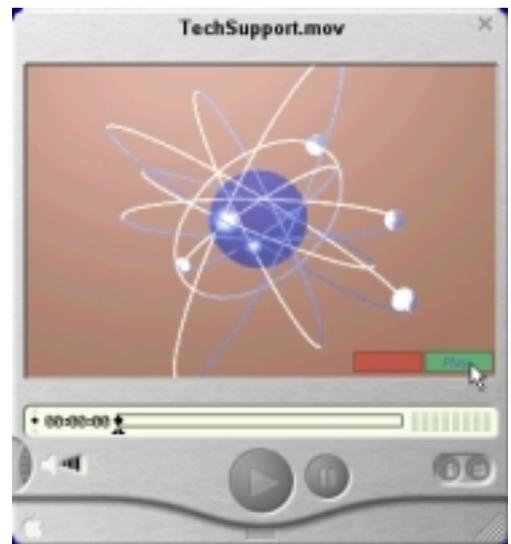


Abbildung 9 ►
 Der Quicktime-Movie mit der interaktiven Flash-Spur im Quicktime-Player

Laufzeit, während `get-` und `setFlashProperty()` in Director 8 die Umprogrammierung der internen Komponenten in bezug auf ihre Erscheinung und Position im Hauptfilm erlaubt. Dies eröffnet gerade im Zusammenhang mit Shockwave neue, extrem flexible Möglichkeiten der Online-Programmierung, bei der eine Shockwave-Präsentation etwa einen aktuellen Flash-Film von einem Server mit Generator anfordern kann und ihn in Abhängigkeit der Benutzerinteraktion im Browser von der Erscheinung her umprogrammiert. Da mit Director 8 auch Shockwave-Präsentationen nun skaliert werden können und die darin enthaltenen Flash-Präsentationen ohnehin vektorbasiert sind und somit immer scharf und ohne Verluste dargestellt werden, kann man als Director- und Flash-Programmierer die besten Möglichkeiten der beiden Werkzeuge für seine eigenen Projekte nutzen: die Skalierbarkeit von Flashpräsentationen und die Interaktivität und Programmierbarkeit von Director.

