

Workshop: Director FOLGE 13

Timing von Animationen

In der vorletzten Folge des Director-Workshops untersuchen wir Möglichkeiten des Timings von Animationen.

Im Vordergrund stehen dabei insbesondere QuickTime und die Steuerung von Animationen durch Lingo.



■ Um zu demonstrieren, wie sich das Timing von Animationen unabhängig vom aktuellen Abspieltempo des Films festlegen lässt, betrachten wir zwei Beispiele: die Steuerung eines QuickTime-Films sowie die Realisierung einer Animation in Lingo. Für beide Varianten entwerfen wir eine gemeinsame aus den Buttons „play“, „stop“, „rewind“ und einem Temporegler bestehende Abspielsteuerung (Abbildung 1). Das dazugehörige Scripting konzipieren wir so, dass alle Button-Behaviors und auch das Behavior des Temporeglers sowohl für den QuickTime-Film als auch für die in Lingo realisierte Animation einsetzbar sind, ohne dass an den Skripttexten Änderungen notwendig werden.

Import der Animationsphasen. Die Laufbewegung der Trickfigur besteht aus insgesamt zehn Phasen, die als externe Grafiken im Pict-Format vorliegen. Der Import erfolgt wie gewohnt über den

entsprechenden Befehl im Menü „Datei“ oder per Drag-and-Drop aus dem Finder. Bei der Betrachtung der Sequenz im Malfenster (zum Blättern können Sie übrigens außer den Pfeilbuttons auch die Pfeiltasten auf Ihrer Tastatur verwenden) fällt auf, dass die Darstellerfolge nicht überall einen flüssigen Bewegungsablauf bewirkt. Solche Verwacklungseffekte entstehen durch die automatische Zentrierung der Grafik beim Import, genauer gesagt durch die Position des so genannten Registrierungspunkts. Diesen platziert Director beim Import automatisch in der geometrischen Mitte des Darstellers. Alternativ zum Registrierungspunkt-Werkzeug im Malfenster können Sie in Lingo übrigens auch die Darstellereigenschaft „the regpoint of member“ setzen, was insbesondere in dem Fall sinnvoll ist, wenn mehrere Darsteller einen festen, von der Voreinstellung abweichenden Registrierungspunkt erhalten sollen.

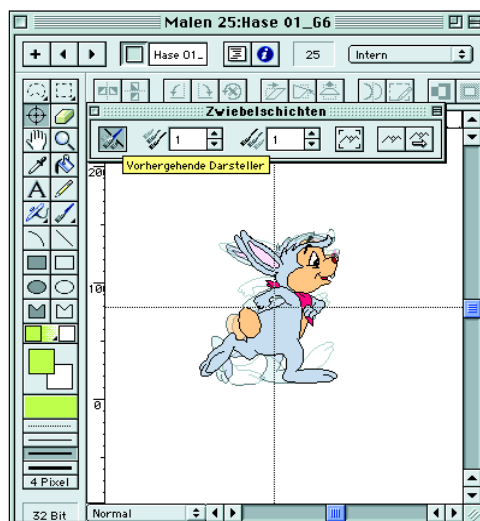
Director unterstützt die Herstellung und Nachbearbeitung von Phasenzeichnungen mit dem so genannten Zwiebschicht- respektive Onion-skinning-Werkzeug. Die Zwiebschichten-Palette erhalten Sie, indem Sie bei geöffnetem Malfenster den betreffenden Eintrag im Menü „Ansicht“ wählen. Auch wenn die einzelnen Phasen außerhalb von Director erstellt wurden und anschließend importiert werden, bietet der Zwiebschicht-Effekt Hilfeleistung (Abbildung 2).

QuickTime-Export. Mit Director erzeugte Filme lassen sich schnell und einfach in das QuickTime-Format konvertieren. Doch natürlich bleiben die interaktiven Funktionen des Director-Films bei diesem Vorgang nicht erhalten. Eine gute Vorschau auf den zeitlichen Ablauf des Films bekommen Sie, wenn Sie ihn mit der Einstellung „Steuerung/Skripts deaktivieren“ abspielen.

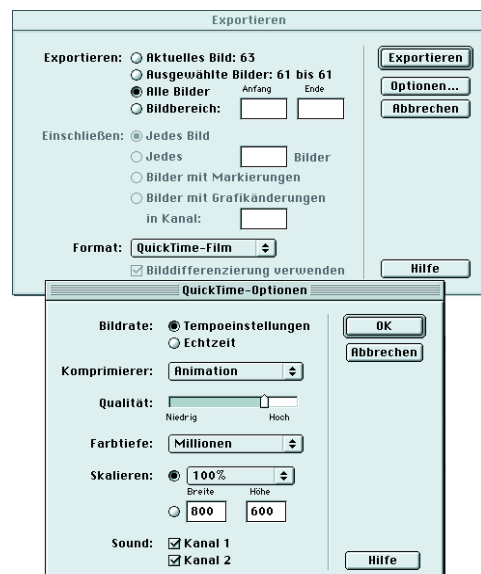
1: Fertige Abspielsteuerung



2: Zwiebschicht-Werkzeug



3: QuickTime-Exportoptionen



Stellen Sie im „Eigenschaften“-Dialog (Menüpfad: Modifizieren/Film) 160 mal 160 Pixel als Bühnengröße ein, und korrigieren Sie gegebenenfalls die Bühnenposition der Animation. Dies ist notwendig, da beim Export immer das gesamte Bühnenbild berücksichtigt wird. Die Angabe eines Bereichs ist hier nicht möglich.

Wählen Sie den Befehl „Exportieren“ im Menü „Datei“ (Befehlstaste-Umschalttaste-R), und stellen Sie im Dialog als Format „QuickTime-Film“ ein. Klicken Sie auf das Auswahlfeld „Alle Bilder“, und öffnen Sie anschließend den Dialog „Optionen“. Hier verwenden wir als Bildrate die Tempoeinstellung und den Codec (Komprimierer) „Animation“ (Abbildung 3).

QuickTime-Import. Erzeugen Sie zunächst einen neuen Film in der Bühnengröße 160 mal 260 Pixel. Um den soeben erzeugten QuickTime-Film zu importieren, verwenden wir entweder den bekannten „Importieren“-Dialog, oder wir ziehen den Film aus dem Finder in die Besetzung.

Im Dialog des Befehls „Darstellerinfo“ (Befehlstaste-I) klicken Sie auf „Optionen“. Schalten Sie die Option „direkt auf der Bühne“ aus und die Option „Schleife“ ein. Ersteres ermöglicht die Verwendung des Farbeffekts „Hintergrund transparent“, Letzteres sorgt für eine andauernde Filmwiedergabe.

Aufbau der Abspielsteuerung. Die Abspielsteuerung besteht aus den Buttons „play“, „stop“ und „rewind“ sowie dem

Schieberegler zur Kontrolle der Bildrate der Animation beziehungsweise des QuickTime-Films. Alle Darsteller werden mit dem Button- oder Rechteckwerkzeug der Werkzeugpalette (Befehlstaste-7) erzeugt. Die Behaviors der Steuerungsknöpfe unterscheiden sich nur durch die Nachricht, die an den zu steuernden Sprite (hier mit der Nummer 11) geschickt wird (Abbildung 4).

Der in diesem Beispiel verwendete einfache Schieberegler besteht aus nur zwei Darstellern: dem beweglichen Schieberegler (Slider) und der Skala. Beide Darsteller lassen sich mit dem Rechteck-Tool der Werkzeugpalette erzeugen. Die Skala hat im Beispiel die Abmessungen 113 mal 15, der Schieberegler 13 mal 13 Pixel. Die Funktionalität des Reglers wird komplett vom Behavior des Sliders realisiert (Abbildung 5). Es arbeitet nach folgendem Prinzip: In der Prozedur „beginSprite“ initialisieren wir die verwendeten Property-Variablen. Beachten Sie, dass das Skript den Sprite der Skala im Drehbuch unmittelbar hinter dem Sprite des Schiebereglers erwartet. Andernfalls müssen Sie die Wertzuweisung für die Property „pScaleSprite“ entsprechend ändern. Die linke und rechte horizontale Koordinate der Skala verwenden wir als Grenzen des Schiebereglers („pMinH“ und „pMaxH“), wobei für den rechten Anschlag noch die Breite des Schiebereglers vom rechten Skalenrand zu subtrahieren ist. Die Property „pMouseH“ speichert die später von der Prozedur „enterFrame“ verwendete horizontale Position des Mauszeigers. ➔

Um in der Prozedur „mouseDown“ eine Repeat-Schleife und die damit verbundenen Probleme bei der Reaktion auf andere Ereignisse zu vermeiden, setzen wir dort mithilfe der Property „pState“ lediglich ein Flag, welches das Drücken der Maustaste anzeigt. Damit der Schieberegler beim ersten Ziehen der Maus nicht ruckartig an die Position des Mausursors springt, speichern wir in der Property „pOffs“ den horizontalen Abstand zwischen Sprite- und Mauskoordinate.

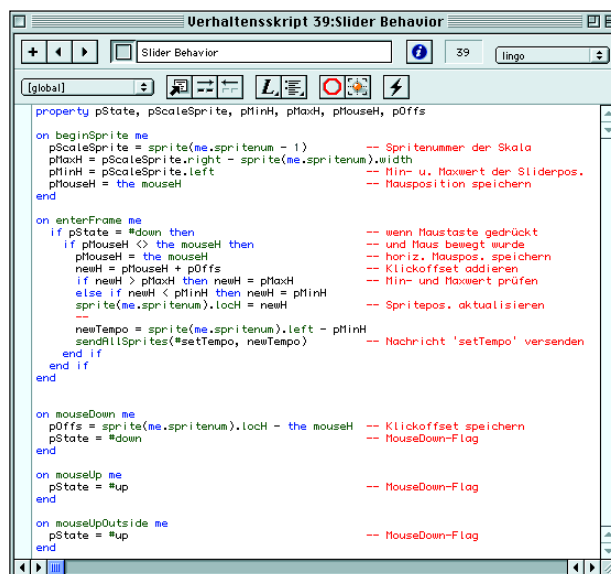
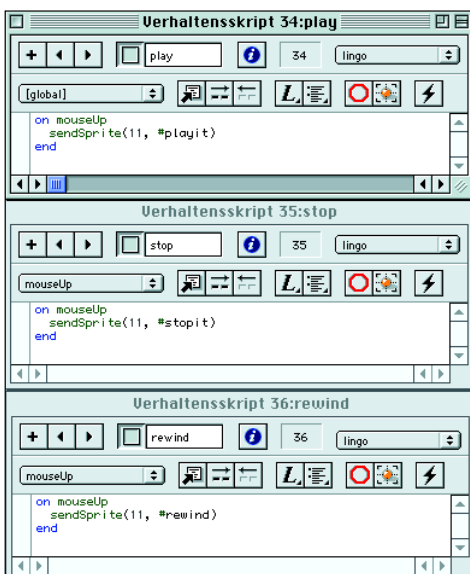
Animationen im QuickTime-Format lassen sich erwartungsgemäß mit weniger Programmieraufwand steuern

Sobald man die Maustaste über dem Sprite (Prozedur „mouseUp“) oder außerhalb der Sprites (Prozedur „mouseUpOutside“) loslässt, wird die Property „pState“ wieder auf „#up“ gesetzt.

Der Wert der Property „pState“ entscheidet in der Prozedur „enterFrame“ darüber, ob die Mausposition in Bühnenkoordinaten für den Schieberegler und eine Tempoangabe für unseren Animationssprite umgerechnet wird oder nicht. Um die Berechnungen nicht unnötig oft vorzunehmen, merkt sich die Prozedur „enterFrame“ die horizontale Position der Maus und schaltet sich nur dann ein, wenn der Anwender die Maus tatsächlich nach links oder rechts verschoben hat. ➔

4: Behavior der Steuerungsknöpfe

5: Schieberegler-Behavior



➔ In diesem Fall speichern wir die neue Position der Maus in der Property „pMouseH“ und verwenden die Variable „newH“, um die aus der Mausposition resultierende Bühnenkoordinate des Schiebereglers zu ermitteln. Abschließend muss aus der Bühnenkoordinate „newH“ des Reglers noch das resultierende Abspieltempo berechnet werden: Die Subtraktion der linken Skalenbegrenzung „pMinH“ von der aktuellen Reglerposition liefert eine Zahl im Bereich zwischen 0 (linker Anschlag) und 100 (rechter Anschlag).

Die in Lingo realisierte Animation ist nicht von externen Daten abhängig und ermöglicht bessere Reaktionen auf Maus-Events

Das so errechnete Tempo übergeben wir als Parameter der Prozedur „setTempo“, die wir mithilfe des Befehls „sendAllSprites“ der Einfachheit halber an alle Sprites versenden. Die Alternative zum Versand der Nachricht an alle Sprites wäre der Befehl „sendSprite“. Er würde die Angabe der Sprite-Nummer erwarten, welche die Nachricht erhalten soll. Der Nachteil bei der Arbeit mit festen Sprite-Nummern besteht darin, dass man das Behavior nach einer eventuellen Verschiebung des adressierten Sprites in einen anderen Kanal editieren muss.

Da sich das Behavior des Schiebereglers auf das Versenden der Nachricht „setTempo“ beschränkt, können wir es sowohl für die Steuerung der in Lingo realisierten Animation als auch zur Änderung des Abspieltempos des QuickTime-Films einsetzen. Um auf die Nachricht „setTempo“ reagieren zu kön-

nen, müssen die beiden Behaviors lediglich über eine gleichnamige Funktion verfügen.

Steuerung des QuickTime-Films. Lingo bietet weit reichende Möglichkeiten zur Kontrolle eines QuickTime-Films. Um das Abspielverhalten zu kontrollieren, verwenden wir folgende Sprite-Propertys:

„The movieRate of sprite“ bestimmt das Tempo. Ein Wert von eins entspricht der originalen Geschwindigkeit. Darunter liegende Werte verlangsamen die Wiedergabe, größere beschleunigen sie.

„The movieTime of sprite“ legt die aktuelle Zeit des Films, also den Ausschnitt fest. Am Filmanfang hat diese Eigenschaft den Wert null, am Ende ist sie gleich der Filmdauer, die sich mit der Eigenschaft „the duration of member“ ermitteln lässt.

Damit das Behavior des QuickTime-Sprites auf die von den Buttons der Abspielsteuerung versandten Nachrichten reagieren kann, integrieren wir hier sämtliche Prozeduren zur Steuerung der Filmwiedergabe.

Die Prozedur „playit“ startet die Wiedergabe des Films, indem sie die Darsteller-Property „the movieRate of sprite“ auf den Wert eins setzt (Abbildung 6).

Das Pendant bildet die Prozedur „stopit“: Sie setzt die Property auf den Wert null und stoppt so die Filmwiedergabe. Die Funktion des Zurückspulens an den Anfang realisieren wir mit der Prozedur „rewind“. Diese setzt die Darsteller-Property „the movieTime of sprite“ auf den Wert null.

Um innerhalb der Prozedur „setTempo“ die Bildrate des QuickTime-Films zu steuern, genügt es, die vom Schieberegler gelieferte Zahl zwischen 0 und 100 in eine Gleitkommazahl im Bereich von 0 bis 1.0 umzurechnen und sie der Darsteller-Property „the movieRate of sprite“ zuzuweisen.

In Lingo realisierte Animation. In Folge 2 des Workshops hatten wir so genannte Filmschleifen (Darsteller vom Typ #filmloop) verwendet, um komfortabel mehrere Animationen kombinieren zu können. Aus Lingo heraus sind die Möglichkeiten zur Steuerung von Filmschleifen allerdings sehr beschränkt. War es zum Beispiel noch in Director 6.5 möglich, in einer Filmschleife mithilfe des Befehls „updateStage“ auf den jeweils folgenden Darsteller zu schalten, so funktioniert dieses Verfahren in der Version 7 nicht mehr.

Die in Lingo realisierte „Filmschleife“ funktioniert nach folgendem Prinzip: Im Drehbuch platzieren wir nur den Darsteller der ersten Animationsphase, die anderen werden während der Filmwiedergabe vom Behavior des Sprites aktiviert. Das Behavior verwaltet dazu unter anderem die Propertys „pMemList“, „pTempo“ und „pState“ (Abbildung 7). Während in „pMemList“ die Liste aller anzuzeigenden Members gespeichert ist, enthält „pState“ den aktuellen Status; mögliche Werte für Letzteren sind hier „#play“ und „#stop“. Der Status wird von der Prozedur „enterFrame“ getestet. Er bestimmt, ob die Animation läuft oder nicht.

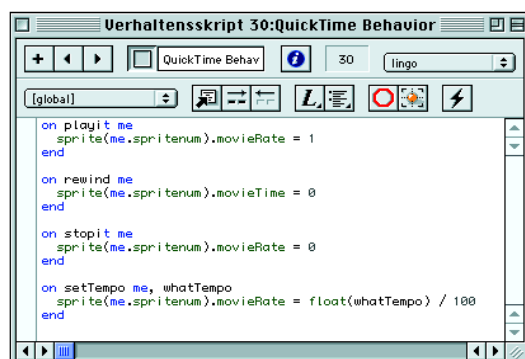
Wenn sich die Animation im Modus „#play“ befindet, sendet die Prozedur „enterFrame“ die Nachricht „animate“. Die gleichnamige Prozedur ist ebenfalls im Behavior des Sprites gespeichert und für die zyklische Anzeige der in der Liste „pMemList“ enthaltenen Darsteller zuständig. Doch nicht bei jedem Bildwechsel soll auch ein Wechsel des aktuellen Darstellers erfolgen. Vielmehr bestimmt der in der Property „pTempo“ enthaltene Wert, nach wie vielen „Ticks“ (1 Tick = 1/60 Sekunde) der Wechsel vonstatten gehen soll.

Das Behavior merkt sich den Zeitpunkt des letzten Darstellerwechsels in der Property „pTimer“. Diese wird von der Prozedur „beginSprite“ auf den Wert des Director-internen Zählers „the ticks“ gesetzt, der die seit dem Programmstart vergangene Zeit enthält.

Ist der aktuelle Wert von „the ticks“ größer als die Summe aus gespeichertem Zeitpunkt „pTimer“ und dem Tempo „pTempo“, schaltet Director auf den folgenden Darsteller in der Liste „pMemList“ um. Dazu verwenden wir die Property „pCounter“. Sie wird von der Prozedur „beginSprite“ mit dem Wert null initialisiert und anschließend von der Prozedur „animate“ in jedem Durchlauf um eins erhöht. Wenn so das Ende der Liste erreicht ist, hängt das weitere Verfahren vom Wert der Property „pLoop“ ab. Entweder beginnt der Durchlauf erneut beim ersten Listenelement, oder die Animation stoppt.

Property-Variablen setzen. Die Werte der Property-Variablen lassen sich auf zwei verschiedenen Wegen setzen. Zum einen kann dies direkt im Skripttext in einer automatisch ausgeführten Prozedur wie etwa „beginSprite“ geschehen. Diese Variante ist mit geringem Aufwand, aber auch mit wenig Komfort verbunden, da jede Ände- ➔

6: Behavior des QuickTime-Sprites



→ rung eines Werts eine Skript-Bearbeitung nötig macht. Andererseits lassen sich Property's auch dialoggesteuert setzen. Zahlreiche Beispiele für diese individuellen Konfigurationsdialoge finden Sie in den im Lieferumfang von Director 7 enthaltenen Behaviors der Bibliothekspalette (Menü „Fenster“).

Verantwortlich für den Inhalt des Konfigurationsdialogs ist die Prozedur mit dem reservierten Namen „getPropertyDescriptionList“. Ihre etwas kryptisch anmutende Syntax erschließt sich am besten durch die Analyse verschiedener Anwendungsbeispiele. Prinzipiell gibt die Prozedur lediglich per „return“ eine verschachtelte Property-Liste zurück. Jede zu setzende Property-Variable besitzt eine Eigenschaft (Property) in der äußeren Liste. Der zugehörige Wert (Value) jeder Eigenschaft ist wiederum eine Property-Liste, die das Aussehen des entsprechenden Eintrags im Konfigurationsdialog bestimmt. Eine solche innere Property-Liste enthält die Einträge „#comment“ (den im Dialog erscheinenden Text), „#format“, (das Datenformat der Property), „#default“ (den voreingestellten

Wert) sowie optional noch „#range“ (das ist eine Bereichsangabe bei Mehrfachauswahlen).

Zum Setzen der Property's unserer Animation benötigen wir einen Dialog mit folgenden Eingabefeldern:

- den ersten sowie den letzten anzuzehenden Darsteller (wobei wir voraussetzen, dass sich die Darsteller in aufeinander folgenden Positionen der Besetzung befinden),
- die Zeit, die zwischen zwei Darstellerwechseln gewartet werden soll,
- die Auswahl, ob die Animation in einer Endlosschleife läuft oder nicht (Abbildung 8).

Dem abgebildeten Dialog entspricht die Prozedur „getPropertyDescriptionList“ im Behavior (Abbildung 9).

Die Prozeduren „playit“, „stopit“, „rewind“ und „setTempo“ im Behavior rufen die entsprechenden Buttons oder den Temporegler auf. Während die Prozeduren „playit“ und „stopit“ die Wiedergabe der Animation durch einfaches Setzen der Property „pState“ kontrollieren, muss die Prozedur „rewind“ zunächst den internen Index „pCounter“ auf den Wert eins setzen und dann

den aktuell mit dem Sprite verbundenen Darsteller durch den ersten Darsteller der Liste „pMemList“ ersetzen. Die Steuerung des Abspieltempos übernimmt die Prozedur „setTempo“, indem sie die beim Prozeduraufruf übergebene Pausenzeit (im Bereich von 1 bis 100) umrechnet.

Der Vergleich der beiden Techniken zeigt, dass sich Animationen im QuickTime-Format erwartungsgemäß mit geringerem Programmieraufwand steuern lassen. Nachteile dieser Technik sind zum einen die deutlich schwierigere Reaktion auf Maus-Events in Teilbereichen des Films und die Abhängigkeit des Films von externen Daten.

Die in Lingo realisierte Animation besitzt diese Nachteile nicht, erfordert dafür jedoch einen höheren Scripting-Aufwand. **Gerd Gillmaier ■**

VORSCHAU

■ In der letzten Folge unseres Workshops zeigen wir, wie Sie digitale Alben und Kataloge in Director 7 gestalten.

7: Behavior der Lingo-Animation

8 + 9: Prozedur „getPropertyDescriptionList“ und der resultierende Dialog

```

Verhaltensskript 26:lingo controlled loop
[global]
property pMemList, pState, pTempo, pLoop, pMemStart, pMemStop
property pMem, pTimer, pCounter

on beginSprite me
  pMemList = []
  pTimer = the ticks
  pCounter = 0
  repeat with m = pMemStart.number to pMemStop.number
    add (pMemList, member m)
  end repeat
  put pMemList
end

on enterFrame me
  if pState = #play then
    animate me
  end if
end

on animate me
  if the ticks >= pTimer + pTempo then
    pTimer = the ticks
    pCounter = pCounter + 1
    if pCounter > count(pMemList) then
      pCounter = 1
      if pLoop = #FALSE then
        stop me
        pCounter = count(pMemList)
        exit
      end if
    end if
    pMem = pMemList[pCounter]
    sprite(me.spriteNum).member = pMem
  end if
end

on playit me
  pState = #play
end

on rewind me
  pCounter = 1
  pMem = pMemList[pCounter]
  sprite(me.spriteNum).member = pMem
  updateStage
  pState = #stop
end

on stopit me
  pState = #stop
end

on setTempo me, whatTempo
  pTempo = 100 - whatTempo
end

```

```

Verhaltensskript 26:lingo controlled loop
getPropertyDescr...
pdL = []
pMem = sprite(the currentSpriteNum).member
addprop(pdL, #pMemStart, ~
  [#comment: "erster Darsteller", #format: #member, #default: pMem])
addprop(pdL, #pMemStop, ~
  [#comment: "letzter Darsteller", #format: #member, #default: pMem])
addprop(pdL, #pTempo, ~
  [#comment: "Pause zwischen Bildern", #format: #integer, ~
  #default: 100, #range: [#min: 1, #max: 100]])
addprop(pdL, #pLoop, ~
  [#comment: "Schleife", #format: #boolean, #default: 1])
addprop(pdL, #pState, ~
  [#comment: "Anfangszustand", #format: #symbol, ~
  #default: #play, #range: [#play, #stop]])
return pdL
end getPropertyDescriptionList

```

Parameter für "lingo controlled loop"

erster Darsteller	Hase 01_G1	OK
letzter Darsteller	Hase 01_G10	Abbrechen
Pause zwischen Bildern	100	
Schleife	<input checked="" type="checkbox"/>	
Anfangszustand	play	