

Zeichnen und Animieren in Flash

Prof. Dr. Robert Strzebkowski – TFH Berlin

Überblick - Zeichnen und Animieren

1. Klassische Animationsformen und -Techniken
2. Zeichnen mit ActionScript
3. Interaktives Zeichnen
4. Animieren mit ActionScript
5. Interaktives Animieren mit ActionScript
6. Neue grafische Effekte in Flash 8
7. 3D-Animation mit Flash

1. Klassische Animationsformen

□ Lineare Animation

- Wird vom Anfang bis Ende abgespielt
- Keine bis einfache Abspielsteuerung (Play, Pause, Stop)

□ Mögliche Anwendungsarten

- Vorspann-Animation
- Effekte-Animation (wiederholend)
- Cartoon
- Werbung
- Information, Erläuterung, Beschreibung

3

1. Klassische Animationsformen

□ Lineare Animation

- Konkrete Beispiele:
 - <http://www.2advanced.com/flashindex.htm> (Anfang)
 - <http://www.weatherclassroom.com>
 - <http://www.terminator3.com/>
 - <http://www.nanoreisen.de/> (Vorspann)
 - www.zdf.de / Wissen & Entdecken

4

1. Klassische Animationsformen

□ Non-lineare Animation

- Enthält interaktive Ablauf-Steuerung
 - Z.B. mit Videosteuerungs-Elementen
 - Z.B. Realisiert als Menüs
- Enthält Zugriffspunkte auf Animationsteile
 - Z.B. mit Hotspots

5

1. Klassische Animationsformen

□ Non-lineare Animation

- Beispiele:
 - [Boeing 747](#)
 - <http://www.maxfli.com/> (Golf-Bälle)
 - <http://www.2advanced.com/flashindex.htm> (Menüs)
 - <http://www.nanoreisen.de/>
 - <http://www.madonna.com/madonna/indexFlash.html>
 - www.zdf.de / Wissen & Entdecken ([Lawinen](#))

6

1. Klassische Animationsformen

□ Interaktive Animation

- Enthält interaktive Manipulationsmöglichkeiten
 - Freies Bewegen von Objekten
 - Drag & Drop
 - Drag & Drop mit Weiterlauf (werfen)
 - Vorwiegend mit Scripten verbunden
 - Begrenztes bewegen von Objekten
 - Auf vorgegebenen Pfaden
 - Objekt-Kollisionen

7

1. Klassische Animationsformen

□ Interaktive Animation

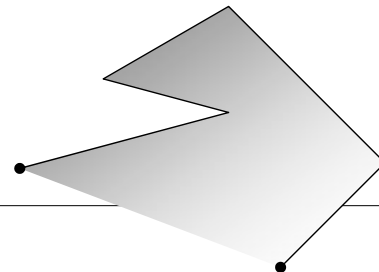
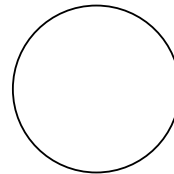
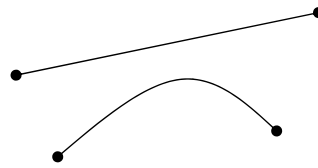
- Beispiele:
 - Quarks-Sendung

8

2. Zeichnen mit ActionScript

□ Was kann mit ActionScript grafisch erzeugt und manipuliert werden?

- Linien zeichnen
- Kurven zeichnen
 - Farbe, Dicke, Opazität, Anschlußart kann festgelegt werden
- Ellipsen und Kreise zeichnen
- Flächen zeichnen und füllen
 - Farbe, Farbverläufe, Alpha...

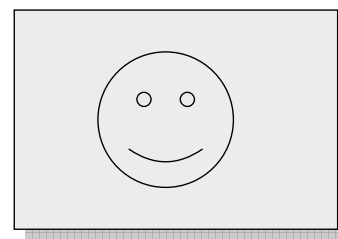
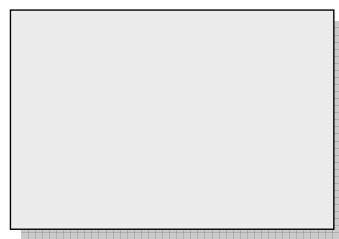


9

2. Zeichnen mit ActionScript

□ Zeichnen

- Für das Zeichnen von Objekten mit ActionScript wird generell zunächst dynamisch ein leeres MovieClip erzeugt, in das anschliessend gezeichnet wird
- Die Zeichnungsanweisungen beziehen sich damit auf ein MovieClip!



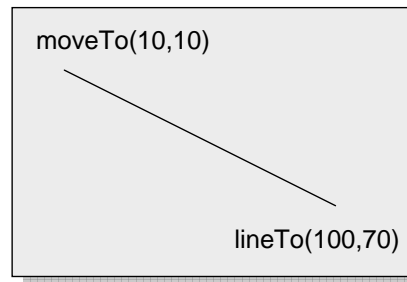
```
_root.createEmptyMovieClip("gesicht_mc", 1);
```

```
gesicht_mc.Zeichnungsanweisungen....
```

2. Zeichnen mit ActionScript

□ Linien Zeichnen

- *MovieClip.moveTo(xKoordinate, yKoordinate);*
 - Setzt den Ausgangspunkt für die anschließende *lineTo*-Methode, durch die angegebenen Koordinaten.
- *MovieClip.lineTo(xKoordinate, yKoordinate);*
 - Zeichnet eine Linie vom aktuellen 'Punkt' zu den durch die Parameter angegebenen Koordinaten.

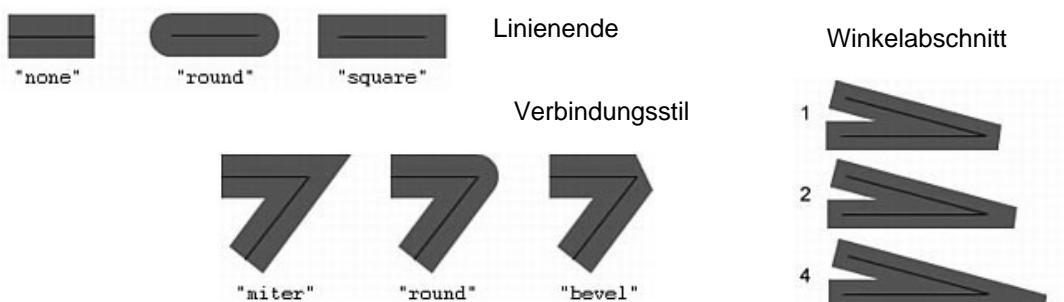


11

2. Zeichnen mit ActionScript

□ ...Linien Zeichnen

- Die Linien sowie deren Verbindungen können vielfältige Eigenschaften haben:
- *MovieClip.lineStyle(linienStaerke, linienFarbe, linienAlpha, linienSkalierung, linienEnde, Verbindungsstil, Winkelabschnitt);*

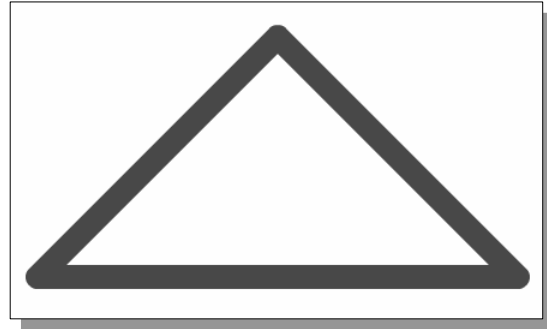


12

2. Zeichnen mit ActionScript

□ ...Linien und Objekte Zeichnen

- `this.createEmptyMovieClip("triangle_mc", this.getNextHighestDepth());`
- `triangle_mc.lineStyle(10, 0xff00ff, 100, true, "normal", "round", "round", 120);`
- `triangle_mc.moveTo(200, 200);`
- `triangle_mc.lineTo(300, 300);`
- `triangle_mc.lineTo(100, 300);`
- `triangle_mc.lineTo(200, 200);`



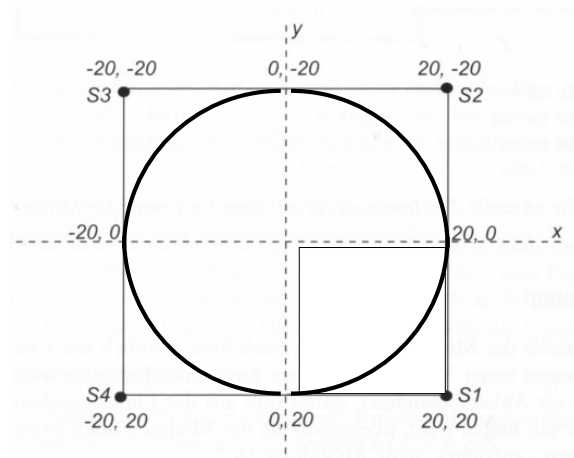
13

2. Zeichnen mit ActionScript

□ Kurven und Kreise Zeichnen

- mit Hilfe der 'curveTo'-Eigenschaft

```
moveTo(0, 20);  
curveTo( 20, 20, 20, 0);  
curveTo( 20, -20, 0, -20);  
curveTo(-20, -20, -20, 0);  
curveTo(-20, 20, 0, 20);
```



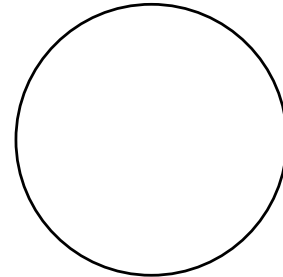
14

2. Zeichnen mit ActionScript

□ Kurven und Kreise Zeichnen

- mit Hilfe der eigenen Sinus-, Cosinus und Tangent-Funktionen, um den Kreiseffekt perfekter zu erhalten
- Nutzung der *Math-Klasse*:

```
this.createEmptyMovieClip("circle2_mc", 2);  
circle2_mc.lineStyle(0, 0x000000);  
drawCircle(circle2_mc, 100, 100, 100);  
function drawCircle(mc:MovieClip, x:Number, y:Number, r:Number):Void  
{ mc.moveTo(x+r, y);  
  mc.curveTo(r+x, Math.tan(Math.PI/8)*r+y, Math.sin(Math.PI/4)*r+x,  
  Math.sin(Math.PI/4)*r+y);  
  ...  
}
```

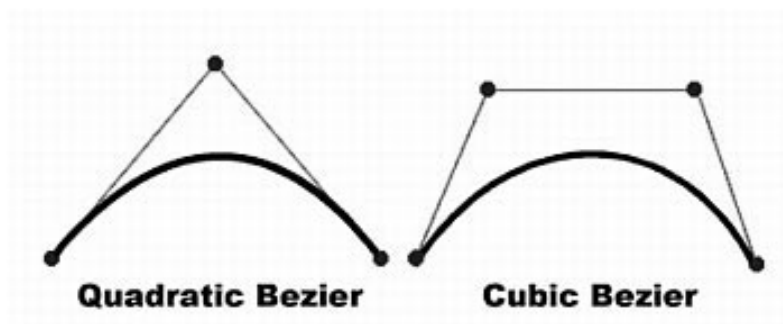


15

2. Zeichnen mit ActionScript

□ ...Kurven und Kreise Zeichnen

- mit curveTo werden quadratische Bezier-Kurven gezeichnet



16

2. Zeichnen mit ActionScript

□ Flächen füllen

- `mein_mc.beginFill(0xFF0000);`
- `this.createEmptyMovieClip("square_mc",
this.getNextHighestDepth());
square_mc.beginFill(0xFF0000);
square_mc.moveTo(10, 10);
square_mc.lineTo(100, 10);`






17

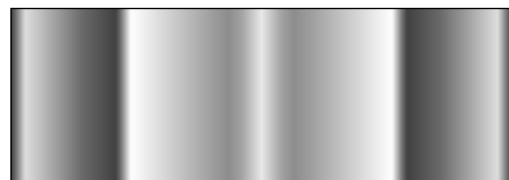
2. Zeichnen mit ActionScript

□ ...Flächen füllen mit Effekten

- linearer 'einfacher' Verlauf
- `beginGradientFill(fillType:String, colors:Array)`

ratios	Verlauf
[0, 127]	
[0, 255]	
[127, 255]	

- linearer 'mehrfacher' Verlauf
- `beginGradientFill(fillType, colors, alphas, ratios,
matrix, spreadMethod);`

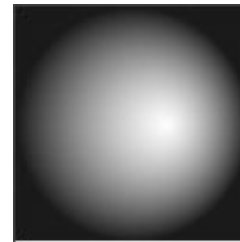


18

2. Zeichnen mit ActionScript

□ ...Flächen füllen mit Effekten

- radialer Verlauf
- *`beginGradientFill("radial", [0xFFFFFFFF, 0x000000])`*

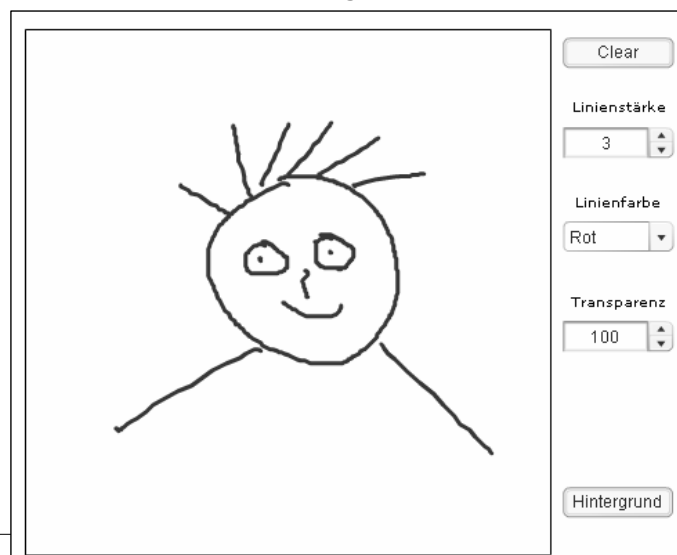


19

3. Interaktives Zeichnen mit ActionScript

- ### □ Mit Hilfe der 'draw'- und 'lineTo'-Anweisung, verbunden mit der ständigen Abfrage der Mausposition kann ein kleines Malprogramm erstellt werden:

```
onMouseMove = function(){  
  if(isInPA()){  
    lineStyle(_root._thickness,  
              _root._color,  
              _root._alpha);  
    if(draw){  
      lineTo(_xmouse, _ymouse);  
    }  
  }  
}
```



4. Animieren mit ActionScript

- Mit Hilfe mathematischer Berechnungen können weiche Animationen, Wurfstechniken oder 'Gummiband'-Verhalten simuliert werden.
- Flash bietet integrierte Klassen, mit denen *Bewegungsabläufe* gesteuert werden können:
 - die Tween-Klasse:
 - (*ActionScript-Klassenname: mx.transitions.Tween*)
 - Bewegungsabläufe
 - Beschleunigungs- und Abbremsverhalten
 - *var tweenObjekt = new mx.transitions.Tween(obj, prop, func, begin, finisch, duration)*

21

4. Animieren mit ActionScript

...die Tween-Klasse

- *var tweenObjekt = new mx.transitions.Tween(obj, prop, func, begin, finisch, duration, [useSeconds])*
 - *obj*: Objekt
 - *prop*: Eigenschaft des Tweens (z.B.: *_x*, *_yscale*, *_alpha*)
 - *func*: Verweis auf die Methode der 'Easing'-Klasse
 - *begin*: Ausgangswert der 'prop'-Eigenschaft
 - *finisch*: Endwert der 'prop'-Eigenschaft
 - *duration*: legt die Bewegungsdauer
 - *useSeconds*: Festlegung, ob 'duration' in Frames (false) oder in Sekunden (true) ausgeführt wird

22

4. Animieren mit ActionScript

Übersicht: Methoden der Tween-Klasse

In der folgenden Tabelle sind die Methoden der Tween-Klasse aufgelistet:

Methoden	Beschreibung
<code>Tween.yoyo()</code>	Weist die getweente Animation an, vom aktuellen Wert bis zu einem neuen Wert fortzufahren.
<code>Tween.fforward()</code>	Bringt die getweente Animation direkt zum Ende der Animation.
<code>Tween.nextFrame()</code>	Bringt die getweente Animation zum nächsten Bild.
<code>Tween.prevFrame()</code>	Leitet die getweente Animation zum Bild vor dem aktuellen Bild.
<code>Tween.resume()</code>	Nimmt eine getweente Animation ab dem Anhaltepunkt in der Animation wieder auf.
<code>Tween.rewind()</code>	Bringt eine getweente Animation zum Anfang der getweenten Animation.
<code>Tween.start()</code>	Startet die getweente Animation vom Anfang an.
<code>Tween.stop()</code>	Beendet die getweente Animation an der aktuellen Position.
<code>Tween.toString()</code>	Gibt den Klassennamen, „[Tween]“, zurück.
<code>Tween.yoyo()</code>	Bewirkt, dass die getweente Animation umgekehrt wie bei der letzten Inkrementierung der getweenten Eigenschaft abgespielt wird.

23

4. Animieren mit ActionScript

Übersicht: Eigenschaften der Tween-Klasse

In der folgenden Tabelle sind die Eigenschaften der Tween-Klasse aufgelistet.

Eigenschaft	Beschreibung
<code>Tween.duration</code>	Die Dauer der getweenten Animation in Bildern oder Sekunden. Schreibgeschützte Eigenschaft.
<code>Tween.finish</code>	Der letzte getweente Wert für das Ende der getweenten Animation. Schreibgeschützte Eigenschaft.
<code>Tween.FPS</code>	Die Anzahl der Bilder pro Sekunde der getweenten Animation. Schreibgeschützte Eigenschaft.
<code>Tween.position</code>	Der aktuelle Wert der getweenten Eigenschaft des Ziel-Movieclips. Schreibgeschützte Eigenschaft.
<code>Tween.time</code>	Die aktuelle Zeit in der Dauer der Animation. Schreibgeschützte Eigenschaft.

24

4. Animieren mit ActionScript

Übersicht: Ereignisprozeduren der Tween-Klasse

In der folgenden Tabelle sind die Ereignisprozeduren der Tween-Klasse aufgelistet.

Ereignis	Beschreibung
<code>Tween.onMotionChanged</code>	Ereignisprozedur; wird bei jeder Änderung der animierten Eigenschaft des getweenten Objekts aufgerufen.
<code>Tween.onMotionFinished</code>	Ereignisprozedur; wird aufgerufen, wenn die Animation des Tween-Objekts beendet ist.
<code>Tween.onMotionResumed</code>	Ereignisprozedur; wird aufgerufen, wenn die Methode <code>Tween.resume()</code> aufgerufen wird und die Wiederaufnahme der getweenten Animation bewirkt.
<code>Tween.onMotionStarted</code>	Ereignisprozedur; wird aufgerufen, wenn die Methode <code>Tween.start()</code> aufgerufen wird und den Start der getweenten Animation bewirkt.
<code>Tween.onMotionStopped</code>	Ereignisprozedur; wird aufgerufen, wenn die Methode <code>Tween.stop()</code> aufgerufen wird und das Beenden der getweenten Animation bewirkt.

25

4. Animieren mit ActionScript

Die Easing-Klasse

- Stellt 16 verschiedene Bewegungsabläufe in 6 Kategorien zur Verfügung
 - **Back:** Erweiterung der Animation an den Enden
 - **Bounce:** Springeffekt am Anfang oder am Ende
 - **Elastic:** elastischer Effekt am Anfang oder am Ende
 - **Regular:** verlangsamte Bewegung am Anfang oder am Ende als Ausbremsung oder Beschleunigung
 - **Strong:** gleich wie 'Regular', jedoch viel stärker
 - **None:** gleichmäßige Bewegung

26

4. Animieren mit ActionScript

...Die Easing-Klasse

- 4 Bewegungsabläufe für Beschleunigungs- und Abbremsverhalten:
 - easeIn: Am Anfang
 - easeOut: am Ende
 - easeInOut: am Anfang und am Ende

- Syntax:
 - `mx.transitions.easing.<Kategorie>.<Bewegungsablauf>`
 - Bsp.: *tweenKugel (kugel_mc, mx.transitions.easing.Bounce.easeOut);*

27

4. Animieren mit ActionScript

- Beispiel eines animierten Menüs mit der tween-Klasse → aus dem Buch "Flash 8" von Kannengießer



28

4. Animieren mit ActionScript

TransitionManager-Klasse

□ Angebot von 10 Übergängen

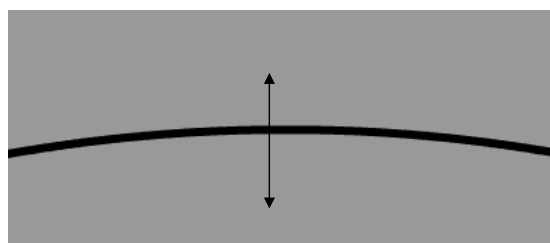
- Iris
- Verwischen
- Pixelauflösung
- Lammellen
- Ein-/Ausblenden
- Schieben
- Zoom
- Stauchen
- Drehen
- Foto

- #### □ Syntax: *TransitionManager.start(ziel_mc, {type:Fly, direction:0, duration:3, easing:Elastic.easeInOut, startPoint:6});*

29

5. Interaktives Animieren mit ActionScript

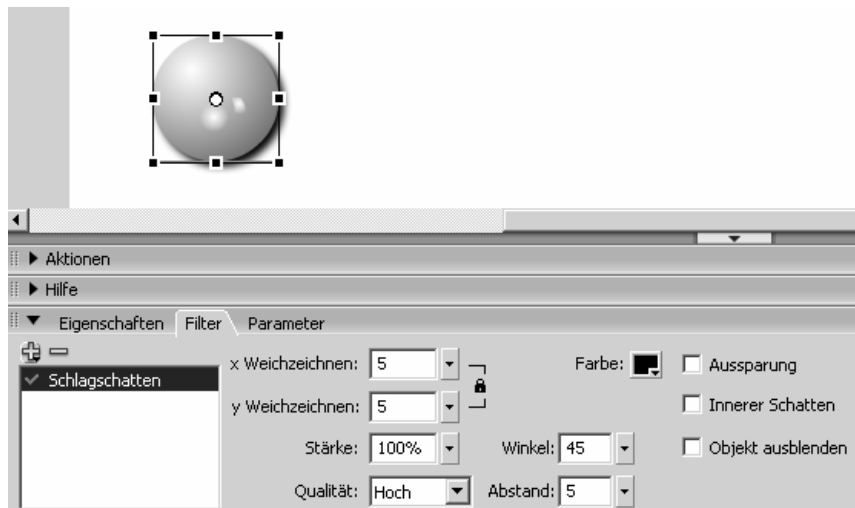
- #### □ Mit Hilfe mathematischer Berechnungen können weiche Animationen, Wurfstechniken oder 'Gummiband'-Verhalten simuliert werden.
- #### □ Diese basieren oft auf dem Ereignis *'mc.onEnterFrame = function () {...}'*



30

6. neue Grafische Effekte bei Flash 8

...endlich !



31

6. neue Grafische Effekte bei Flash 8

Filtereffekte mit ActionScript

□ z.B. 'DropShadowFilter'-Klasse

- *var meinSchatten:flash.filters.DropShadowFilter = new flash.filters.DropShadowFilter (10, 45, 0x000000)*
- mit Import der Klasse:
 - *import flash.filters.Blurfilter;*
 - *var meinBlur:Blurfilter = new Blurfilter(10, 45, 3);*
- Empfehlung: Import aller Filter-Klassen:
 - *import flash.filters.*;*

32

6. neue Grafische Effekte bei Flash 8

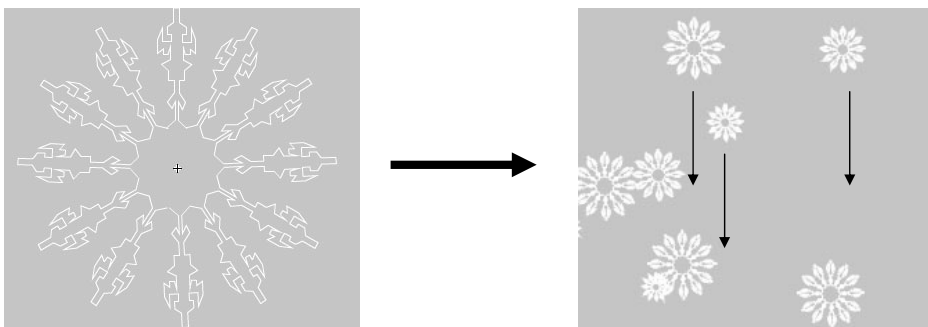
- ! Viele der Filter aus FireWorks werden beim Import in Flash als FW-PNG Datei beibehalten und können nachträglich bearbeitet werden!
- Mischmodi in Flash

33

6. neue Grafische Effekte bei Flash 8

Bitmap Surface Caching

- Umwandlung komplexer Vektor-MovieClips zur laufzeit in Bitmaps zur Beschleunigung der Abspielgeschwindigkeit und weicher Darstellung der Animationen



- Syntax: *meinClip_mc.cacheAsBitmap = true;*

34

7. 3D-Animation in Flash

- Beispiele

4. Animieren mit ActionScript

- Animation mit vorbereiteten Animationsklassen:
 - TransitionManager-Klasse
 - Tweening-Klasse
 - Easing-Klasse
 - `mx.transitions.easing.Bounce.easeOut,param1:empty,param2:empty}}`;
 - → Beispiele

6. Neue grafische Filter in Flash

□ !! Beispiel → die 3D-Galerie !!