



Selma-Caroline Kannengiesser
Matthias Kannengiesser

Adobe Flash CS3

Powerworkshops

Exklusive
Flashstar Workshops
für Windows und Mac OS X



30-Tage-Testversion von
Flash CS3, Tools, Workshops,
Tutorials usw.



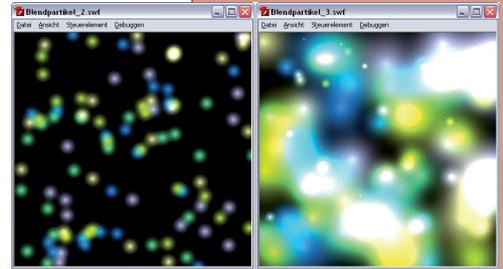
ADDISON-WESLEY

PEARSON
Education

1

EFFEKTE IN FLASH CS3

Im folgenden Kapitel werden wir Ihnen einen Überblick über die Flash-CS3-Effekte und deren Einsatzmöglichkeiten verschaffen. Sie finden zahlreiche nützliche und kompakte Workshops vor, die Ihre Arbeit mit Flash CS3 bereichern werden. Wie Sie bereits dem Vorwort entnehmen konnten, setzen wir ein fundiertes Grundlagenwissen voraus. Vor allem, da die Workshops zielgerichtet sind, können wir nicht beim Unterschied zwischen Bitmap- und Vektorgrafiken beginnen. Sie sollten sich nun anschnallen und in die Tiefen der Flash-Galaxien eintauchen.



1.1 Mischmodi im Griff – MovieClip.blendMode

Wir wollen mit Ihnen als Erstes in die Welt der Mischmodi eintauchen und Ihnen einige interessante Effekte vorstellen. Vorab sollten wir jedoch einen Blick auf die zur Verfügung stehenden Mischmodi werfen. Mithilfe des Mischmodus sind Sie in der Lage, das Erscheinungsbild eines Movieclips und der von diesem überlagerten Objekte zu beeinflussen.

Zum besseren Verständnis haben wir Ihnen eine Abbildung mit sämtlichen verfügbaren Mischmodi aufbereitet und darüber hinaus eine Tabelle mit den Einstellungen zur Movieclip-Eigenschaft `blendMode` zusammengestellt. Um die Eigenschaft `blendMode` verwenden zu können, geben Sie entweder eine Ganzzahl zwischen 1 und 14 an oder die jeweilige Bezeichnung der Mischmodi in Form einer Zeichenkette (*String*).

ABBILDUNG 1.1

Sämtliche verfügbaren Mischmodi auf einen Blick, so lassen sich Eigenarten des jeweiligen Modus besser deuten.



Zahlenwert	Stringwert	Bedeutung
1	„normal“	Der Movieclip erscheint vor dem Hintergrund. Pixelwerte des Movieclips haben Vorrang vor Hintergrundwerten. Dort, wo der Movieclip transparent ist, wird der Hintergrund sichtbar. Hierbei erfolgt keine Farbmischung zwischen den Objektinstanzen.
2	„layer“	Ermöglicht das Stapeln von Movieclip-Instanzen ohne Festlegung einer Farbmischung. Bei der Variante Löschen und Alpha liefert die Option Ebene die Grundlage zur Darstellung des Mischmodus. Der Mischmodus erzwingt darüber hinaus die Erstellung eines temporären Speichers für die vorläufige Komposition des Movieclips. Bei mehr als einem untergeordneten Objekt in einem Movieclip und der Auswahl einer anderen blendMode-Einstellung als „normal“ wird dieser Vorgang automatisch ausgeführt.
3	„multiply“	Die Grundfarbe der abgedeckten Objektinstanz wird mit der Mischfarbe des aufliegenden Instanzobjekts multipliziert. Dieses Mischungsverhältnis liefert als Ergebnis dunklere Farbtöne. Dieser Mischmodus ist auf beide Objektinstanzen anzuwenden. Diese Einstellung wird im Allgemeinen für Schatten- und Tiefeneffekte verwendet.
4	„screen“	Multipliziert die Grundfarbe der untergeordneten Objektinstanz mit der Mischfarbe der übergeordneten Objektinstanz, was als Resultat hellere Farbtöne liefert. Dieser Mischmodus ist nur auf die aufliegende Objektinstanz anzuwenden. Diese Einstellung wird in der Regel für Hervorhebungen bzw. zum Entfernen von schwarzen Bereichen aus dem Movieclip verwendet.
5	„lighten“	Ersetzt die Farbbereiche der abgedeckten Objektinstanz, die dunkler sind als die Mischfarbe der aufliegenden Objektinstanz. Alle weiteren Farbbereiche bleiben bezüglich des Mischungsverhältnisses unverändert. Dieser Mischmodus ist nur auf die aufliegende Objektinstanz anzuwenden. Diese Einstellung wird in der Regel für Überlagerungen verwendet.
6	„darken“	Ersetzt die Farbbereiche der abgedeckten Objektinstanz, die heller sind als die Mischfarbe der aufliegenden Objektinstanz. Alle weiteren Farbbereiche bleiben bezüglich der Farbmischung unverändert. Dieser Mischmodus ist auf beide Objektinstanzen anzuwenden. Diese Einstellung wird in der Regel für Überlagerungen verwendet.
7	„difference“	Subtrahiert die Mischfarbe der aufliegenden Objektinstanz von der Grundfarbe der untergeordneten Objektinstanz bzw. umgekehrt, je nachdem, welche Farbdefinition den höheren Helligkeitswert aufweist. Dies entspricht der Darstellung eines Farbnegativs in der Realität. Diese Einstellung wird in der Regel für lebendigere Farben verwendet.
8	„add“	Fügt den vorliegenden Farbton der abgedeckten Objektinstanz in Abhängigkeit von der vorliegenden Mischfarbe der aufliegenden Objektinstanz hinzu. Diese Einstellung wird in der Regel zur Animation eines heller werdenden Übergangs zwischen zwei Objekten verwendet.
9	„subtract“	Filtert die Farbe der abgedeckten Objektinstanz in Abhängigkeit von der vorliegenden Mischfarbe der aufliegenden Objektinstanz heraus. Diese Einstellung wird in der Regel zur Animation eines dunkler werdenden Übergangs zwischen zwei Objekten verwendet.
10	„invert“	Liefert als Resultat die Umkehrung der Grundfarbe.
11	„alpha“	Definiert die Farbgebung als Alphamaske. Der Mischmodus Alpha funktioniert nur dann, wenn der Mischmodus Ebene auf die aufliegende Objektinstanz angewendet wird, ansonsten verschwindet die untergeordnete Objektinstanz.
12	„erase“	Die Grundfarbe definiert den Farbwert, der bezüglich der Verwendung in den vorliegenden eingefärbten Pixeln, einschließlich denen des Hintergrundbilds, gelöscht wird. Der Mischmodus Löschen funktioniert nur dann, wenn der Mischmodus Ebene auf die aufliegende Objektinstanz angewendet wird, ansonsten verschwindet die untergeordnete Objektinstanz.

13	„overlay“	Multipliziert oder filtert die Farben der abgedeckten Objektinstanz in Abhängigkeit von der Grundfarbe der aufliegenden Objektinstanz. Ist der Hintergrund heller als 50% Grau, werden die Farben von Movieclip und Hintergrund gefiltert, was zu einer helleren Farbe führt. Ist der Hintergrund dunkler als 50% Grau, werden die Farben multipliziert, was zu einer dunkleren Farbe führt. Dieser Mischmodus ist nur auf die aufliegende Objektinstanz anzuwenden und wird in der Regel für Schattierungseffekte verwendet.
14	„hardlight“	Multipliziert oder filtert die Farben der abgedeckten Objektinstanz in Abhängigkeit von der vorliegenden Mischfarbe der aufliegenden Objektinstanz. Ist der Movieclip heller als 50% Grau, werden die Farben von Movieclip und Hintergrund gefiltert, was zu einer helleren Farbe führt. Ist der Movieclip dunkler als 50% Grau, werden die Farben multipliziert, was zu einer dunkleren Farbe führt. Dies entspricht einem Anleuchten von Objekten in der Realität und wird in der Regel für Schattierungseffekte verwendet.

Tabelle 1.1: MovieClip-blendMode (Mischmodi)

1.1.1 Mischmodi in Echtzeit

Sie wollen in der Lage sein, sämtliche Mischmodi in Echtzeit abzurufen? Kein Problem. Sie sollten hierzu folgenden Workshop genauer unter die Lupe nehmen.

Schritt 1

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Blendmode fla* speichern. Sie benötigen zur Umsetzung eine MOVIECLIP-Instanz mit dem Instanznamen `bild_mc`, eine SCHALTFLÄCHEN-Instanz mit dem Instanznamen `schalter_btn` und eine dynamische TEXTFELD-Instanz mit dem Instanznamen `display_txt`.

ABBILDUNG 1.2

Aufbau des Flash-Films zum Abruf der Mischmodi in Echtzeit

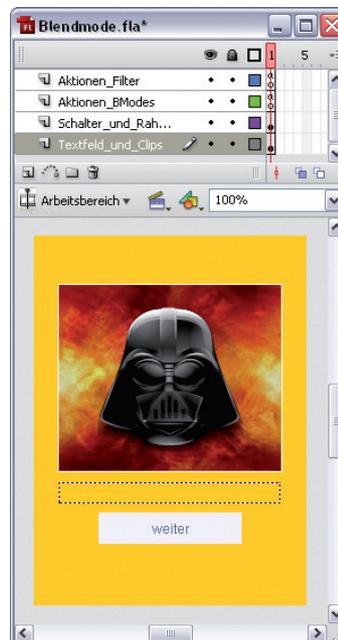



ABBILDUNG 1.3

Sie benötigen lediglich einige elementare Bestandteile, zwei Movieclips und eine Bitmap-Grafik.


ABBILDUNG 1.4

Hierarchie und Instanznamen im Überblick: bild_mc, display_txt und schalter_btn

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```

var modi = ["normal", "layer", "hardlight", "overlay", "alpha", "invert", "subtract", "add", "difference", "darken", "lighten", "screen", "erase", "multiply"];
var i = 0;
setzeMode();

function pressBtn(){
    i = (i >= (modi.length-1)) ? 0 : i+1;
    setzeMode();
}

function setzeMode(){
    bild_mc.blendMode = modi[i];
    display_txt.text = "Blendmodus: " + bild_mc.blendMode;
}

schalter_btn.onPress = function(){
    schalter_btn._xscale = schalter_btn._yscale = 99;
}

```

Schritt 2

```

schalter_btn.onRelease = function(){
    pressBtn();
    schalter_btn._xscale = schalter_btn._yscale = 100;
}

```

Schritt 3

Sie können den Film testen und sollten beim Durchklicken die Besonderheiten und Auswirkungen der diversen Mischmodi genau beobachten. Wir sind uns sicher, dass diverse Mischmodi genutzt werden können, um interessante Effekte umzusetzen. Einige haben wir in den folgenden Workshops für Sie zusammengestellt.

ABBILDUNG 1.5

Durchlauf sämtlicher Mischmodi und deren visuelle Darstellung im Flash Player

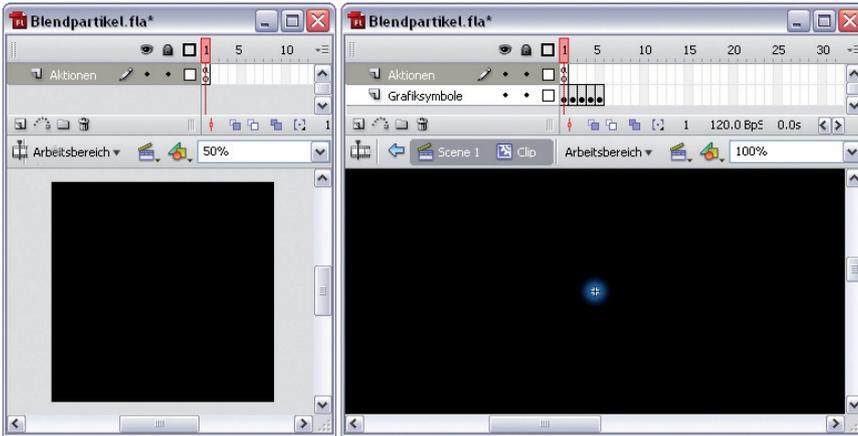


1.1.2 Partikelsimulation und Mischmodi

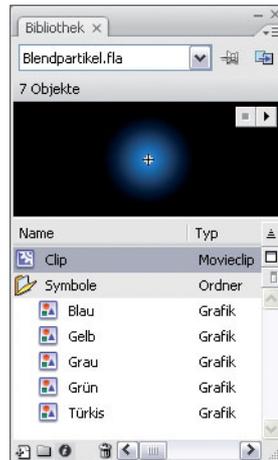
Sollten Sie sich fragen, welche Möglichkeiten sich beim Einsatz der Mischmodi ergeben, dann dürfte Sie unsere Partikelsimulation sicher beeindrucken. Vor allem wenn man bedenkt, dass lediglich der `MovieClip.blendMode` *add* (*Addition*) zum Einsatz kommt.

Schritt 1

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Blendpartikel fla* speichern. Dem `MOVIECLIP`-Symbol *Clip* aus der Bibliothek muss der Verknüpfungsbezeichner *partikel* zugewiesen werden. Sowohl in der Hauptzeitleiste als auch in der Zeitleiste des *Clip*-Movieclips befindet sich jeweils eine *Aktionen*-Ebene.


ABBILDUNG 1.6

Aufbau des Flash-Films und des Clip-Movieclips


ABBILDUNG 1.7

Sämtliche Grafiksymbole sind im Clip-Movieclip enthalten bzw. befinden sich in dessen Zeitleiste.

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
var cx = 170;
var cy = 170;
var i = 0;
while(i < 20){
var mc = this.attachMovie("partikel", ("p" + i), i, {_x:(cx + Math.
random() * 60), _y:(cy + Math.random() * 60)});
    i++;
}
```

Im ersten Schlüsselbild der Zeitleiste des *Clip*-Movieclips platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

Schritt 2

Schritt 3

```

#initclip 0
o = Partikel = function (Void) {
    this.blendMode = "add";
    this.tx = random(360);
    this.ty = random(360);
    this.xtempo = Math.random() / 10;
    this.ytempo = Math.random() / 10;
    this.xd = Math.random() * 10 + 1;
    this.yd = Math.random() * 10 + 1;
    this.x0 = this._x;
    this.y0 = this._y;
    this.gotoAndStop(random(5) + 1);
};
o = o.prototype = new MovieClip();
o["onEnterFrame"] = function (Void){
    this.tx += this.xtempo;
    this.ty += this.ytempo;
    this._x = this.x0 + Math.sin(this.tx) * this.xd;
    this._y = this.y0 + Math.cos(this.ty) * this.yd;
}
delete o;
Object.registerClass("partikel", Partikel);
#endinitclip

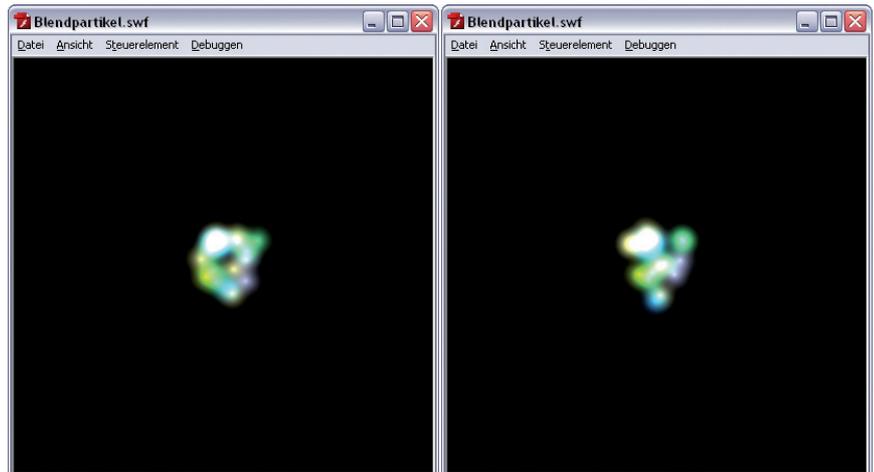
```

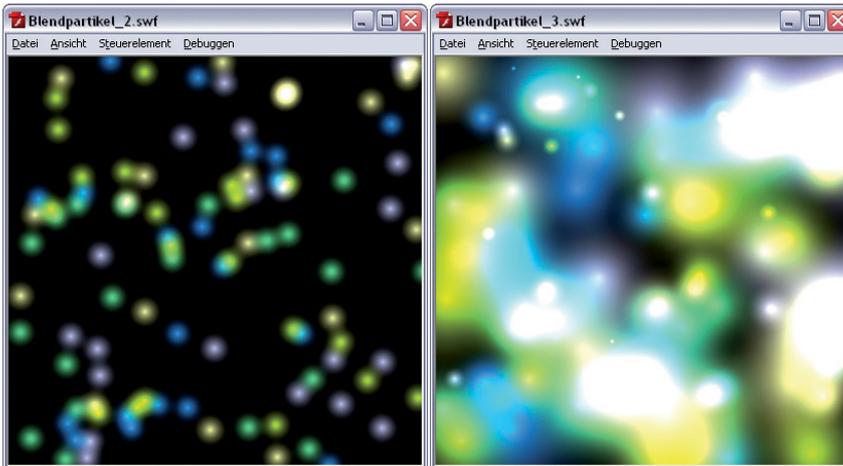
Schritt 4

Sie können den Film testen und werden sehen, dass durch den Einsatz des Mischmodus *Addition* vollkommen neue visuelle Eindrücke entstehen, die bisher in Flash nur mit einem großen Aufwand zu realisieren gewesen wären.

ABBILDUNG 1.8

Partikelsimulation durch
Mischmodus *Addition*




ABBILDUNG 1.9

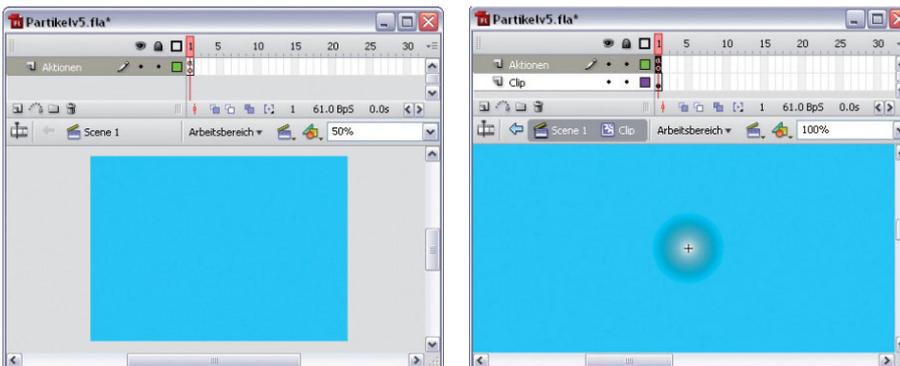
Wie wäre es mit einem ganz besonderen Sternenhimmel?

1.1.3 Erweiterte Partikelsimulation

Sollte Ihnen bereits die Partikelsimulation aus dem vorangegangenen Workshop zugesagt haben, wird Sie diese erweiterte Variante sicher genauso beeindruckend finden. Auch in diesem Fall kommt lediglich der Mischmodus *Addition* zum Einsatz. Zusätzlich wurde zur Wiedergabeoptimierung auf die Movieclip-Eigenschaft `cacheAsBitmap` zugegriffen. Wird diese Eigenschaft auf `true` gesetzt, speichert der Flash Player eine interne Bitmap-Darstellung des Movieclips im Cache (Zwischenspeicher). Dies erhöht bei Movieclips mit komplexen Vektorgrafiken die Leistung.

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Partikel.fla* speichern. Dem MOVIECLIP-Symbol *Clip* aus der Bibliothek muss der Verknüpfungsbezeichner `Clip` zugewiesen werden. Sowohl in der Hauptzeitleiste als auch in der Zeitleiste des *Clip*-Movieclips befindet sich jeweils eine *Aktionen-*Ebene.

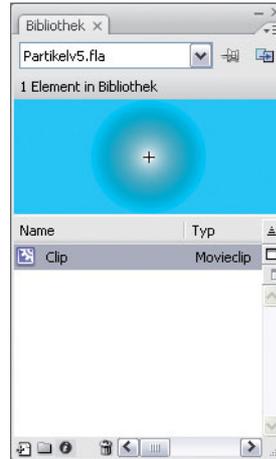
Schritt 1


ABBILDUNG 1.10

Aufbau des Flash-Films und des Clip-Movieclips

ABBILDUNG 1.11

Es wird lediglich der Clip-Movieclip benötigt.



Schritt 2

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
_quality = "LOW";
Stage.scaleMode = "noScale";

var i = 0;
while (i<100) {
    attachMovie("Clip","kugel"+i, i);
    i++;
}
```

Schritt 3

Im ersten Schlüsselbild der Zeitleiste des *Clip*-Movieclips platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
this._x = (Math.random()*1000)-500;
this._y = (Math.random()*1000)-500;

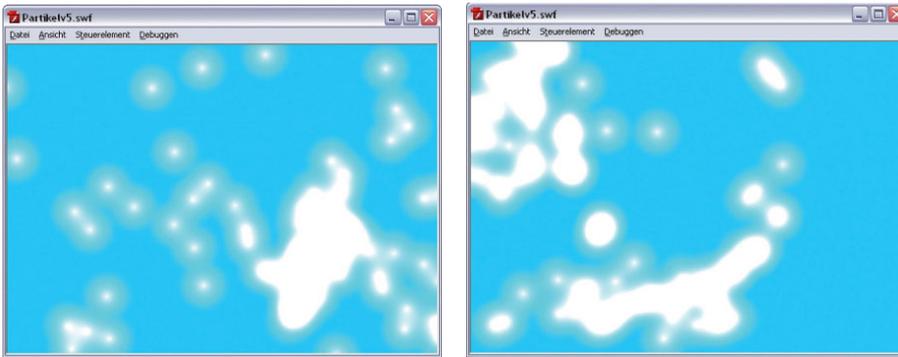
this.cacheAsBitmap = true;
this.blendMode = "add";

var tempoX = 0;
var tempoY = 0;

this.onEnterFrame = function() {
    var faktor = Math.sqrt((this._xmouse*this._xmouse)+(this._ymouse*this._ymouse))+0.1;
    tempoX = tempoX+((this._xmouse/faktor)*0.99);
    tempoY = tempoY+((this._ymouse/faktor)*0.99);
    this._x += (tempoX=tempoX*0.99);
    this._y += (tempoY=tempoY*0.99);
};
```

Schritt 4

Sie können den Film testen. Auch diese Partikelsimulation ist sicher ein interessanter Ansatz für eigene Kreationen. Wir empfehlen Ihnen, einmal ein wenig an den Zufalls-werten zu schrauben.

**ABBILDUNG 1.12**

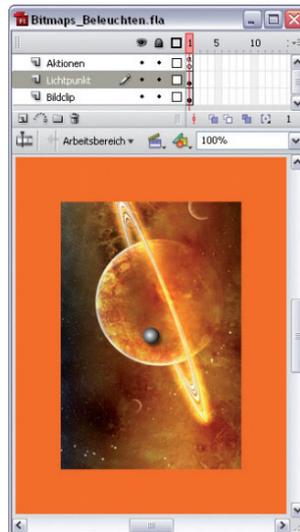
Verschmelzung von Strukturen – durch Einsatz der Mischmodi kein Problem

1.1.4 Bitmaps und Beleuchtungseffekte

Durch den Einsatz der Mischmodi lassen sich nicht nur abstrakte Effekte realisieren. Wie wäre es mit einem handfesten Beleuchtungseffekt, so wie man ihn sonst nur aus Hollywood-Streifen kennt?

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Bitmaps_Beleuchten fla* speichern. Sie benötigen zur Umsetzung zwei MOVIE-CLIP-Instanzen mit den Instanznamen *licht_mc* und *mc*. Innerhalb des *BildClip*-Movieclips platzieren Sie eine beliebige Bitmap-Grafik.

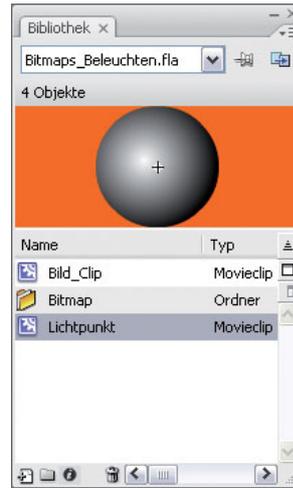
Schritt 1

**ABBILDUNG 1.13**

Der Aufbau des Flash-Films stellt sich recht übersichtlich dar.

ABBILDUNG 1.14

Die Bibliothek hält die wesentlichen Bestandteile parat.

**ABBILDUNG 1.15**

Hierarchie und Instanznamen ausgehend von der Hauptzeitleiste im Überblick



Schritt 2

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
function positionieren (posx, posy, maxanzahl){
    var zaehler = 0;
    while (zaehler < maxanzahl)
    {
        clip = this[("m" + zaehler)];
        clip._x = -zaehler * (licht_mc._x - posx) / 100 + posx;
        clip._y = -zaehler * (licht_mc._y - posy) / 100 + posy;
        zaehler++;
    }
}

function duplizieren(maxanzahl, faktor){
    var i = 0;
    while (i < maxanzahl)
    {
        mc.duplicateMovieClip ("m" + i), i);
        clip = this[("m" + i)];
    }
}
```

```

        clip._alpha = maxanzahl / 2 - i / 2;
        clip._xscale = clip._yscale = 100 + i * faktor;
        clip.blendMode = "add";
        i++;
    }
}

var clip;
positionieren (150, 200, 20);
duplizieren(20, 4);

licht_mc.swapDepths (50);

onMouseMove = function (){
    if (signal)
    {
        licht_mc._x = _xmouse;
        licht_mc._y = _ymouse;
        positionieren (150, 200, 20);
    }
};

onMouseDown = function (){
    licht_mc._x = _xmouse;
    licht_mc._y = _ymouse;
    positionieren (150, 200, 20);
    signal = true;
};

onMouseUp = function (){
    signal = false;
};

```

Sie können den Film testen. Der Beleuchtungseffekt wird durch die Position der Bezugs-MOVIECLIP-Instanz (mc) bestimmt.

Schritt 3

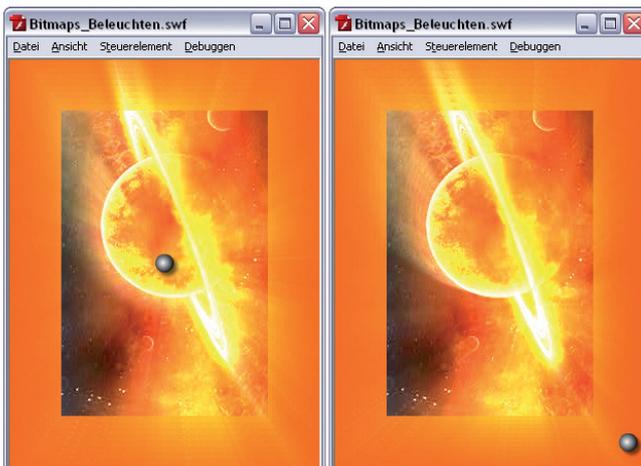
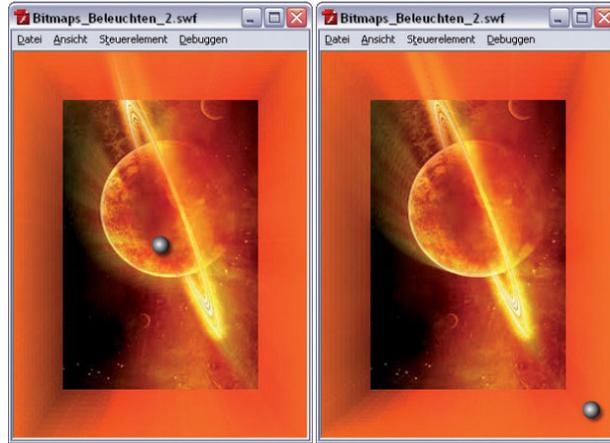


ABBILDUNG 1.16

Was die Mischmodi nicht alles können – da strahlt das Bild.

ABBILDUNG 1.17

Wie wäre es mit dem Mischmodus *hardlight*?



1.1.5 Mischmodi – Kombination von Vektor- und Bitmap-Grafik

Wie wirkt sich die Kombination von Vektor- und Bitmap-Grafiken in Verbindung mit den Mischmodi aus, diese Frage beantworten wir im folgenden Workshop. Ganz nebenbei erfährt man auch noch, wie man eine MOVIECLIP-Instanz in Drehung versetzt, und zwar nutzerabhängig.

Schritt 1

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Drehen.fla* speichern. Sie benötigen zur Umsetzung eine MOVIECLIP-Instanz mit dem Instanznamen *logo_mc* und eine dynamische TEXTFELD-Instanz mit dem Instanznamen *display_txt*. Innerhalb des *BitmapClip*-Movieclips platzieren Sie eine beliebige Bitmap-Grafik.

ABBILDUNG 1.18

Aufbau des Flash-Films – die wesentlichen Bestandteile befinden sich bereits auf der Bühne.



**ABBILDUNG 1.19**

Die Bibliothek hält die wesentlichen Bestandteile parat, vor allem die Movieclips werden zur Umsetzung benötigt.

**ABBILDUNG 1.20**

Hierarchie und Instanznamen im Überblick ausgehend von der Hauptzeitleiste

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
logo_mc.leseMausRotation = function() {
    var x = this._parent._xmouse-this._x;
    var y = this._parent._ymouse-this._y;
    return Math.atan2(y, x)*180/Math.PI;
};
logo_mc.positionzurMaus = function() {
    this._letztedrehung = this._rotation;
    this._rotation = this.leseMausRotation()-this._clickrotation;
};
logo_mc.drehen = function() {
    this._rotation += this.drehTempo;
    this.drehTempo *= .9;
};
logo_mc.onPress = function() {
    this._clickrotation = this.leseMausRotation()-this._rotation;
    this.onEnterFrame = this.positionzurMaus;
```

Schritt 2

```

};
logo_mc.onRelease = logo_mc.onReleaseOutside=function () {
    this.drehTempo = this._rotation-this._letztedrehung;
    if (this.drehTempo>180) {
        this.drehTempo -= 360;
    } else if (this.drehTempo<-180) {
        this.drehTempo += 360;
    }
    this.onEnterFrame = this.drehen;
};

var blendModusIndex = 0;
setInterval(wechselmodus, 2000);

function wechselmodus() {
    logo_mc.blendMode = blendModusIndex%14+1; // values 1 - 14
    display_txt.text = (blendModusIndex%14+1)+": "+logo_mc.blendMode;
    blendModusIndex++;
}

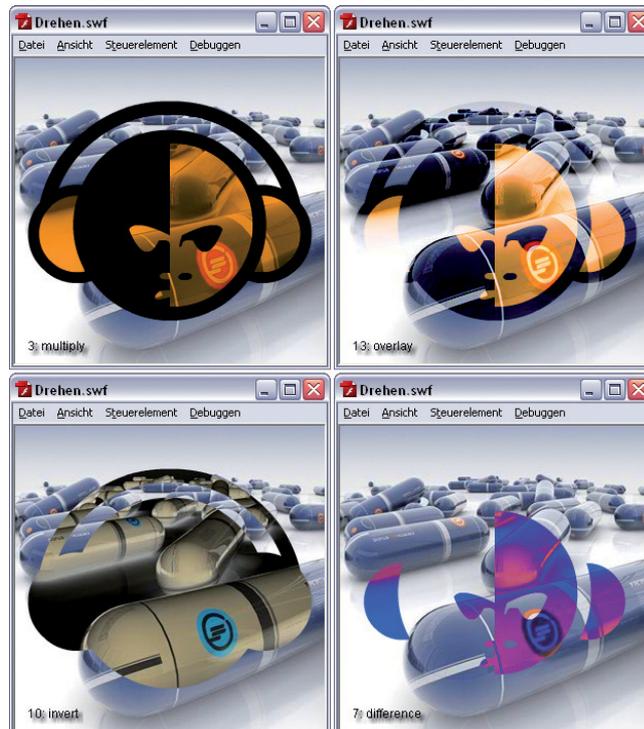
```

Schritt 3

Sie können den Film testen und werden erstaunt sein, was die Mischmodi bei der Vektor- und Bitmap-Kombination für Ergebnisse liefern. Hierdurch eröffnen sich vollkommen neue Möglichkeiten.

ABBILDUNG 1.21

Mischmodi multiply, overlay, invert und difference können sich sehen lassen.



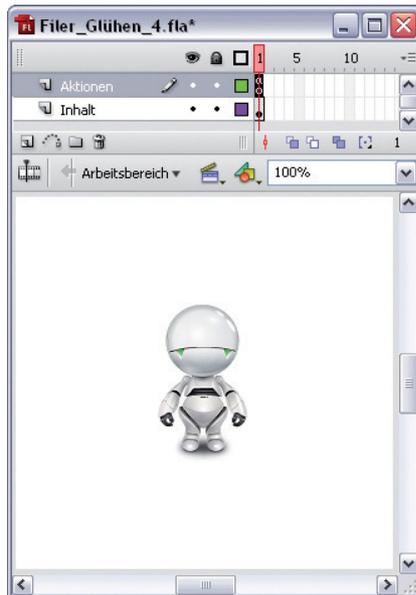
1.2 Filter im Griff – Vielfalt der Filter in Flash CS3

Das `flash.filters`-Paket enthält Klassen für die Bitmap-Filtereffekte, die erstmals in Flash Player 8 eingeführt wurden. Mit Filtern können ausdrucksstarke visuelle Effekte auf `TEXT`-, `MOVIECLIP`- und `SCHALTFLÄCHEN`-Instanzen angewendet werden, wie beispielsweise Weichzeichnung, Schliff, Glühen und Schlagschatten. In Flash gibt es neun Filtereffekte, von denen in Flash CS3 allerdings nur sieben über die Benutzeroberfläche zugänglich sind. Der `ConvolutionFilter` und der `DisplacementMapFilter` stehen nur via ActionScript zur Verfügung.

1.2.1 Glühen-Filter in Echtzeit

Mithilfe der `GlowFilter`-Klasse können verschiedene Objekte in Flash mit einem Glüheffekt versehen werden. Der enthaltene Algorithmus basiert auf dem gleichen Box-Filter, der auch im Weichzeichnen-Filter zum Einsatz kommt. Sie haben verschiedene Möglichkeiten für den beim Glühen verwendeten Stil, beispielsweise inneres oder äußeres Glühen oder den Aussparungsmodus. Der Glühen-Filter ähnelt im Übrigen sehr dem Schlagschatten-Filter, wenn die Eigenschaften `distance` und `angle` des Schlagschattens auf 0 gesetzt sind.

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Filter_Glühen.fla* speichern. Sie benötigen zur Umsetzung eine `MOVIECLIP`-Instanz mit dem Instanznamen `bild_mc`. Innerhalb des *MarvinClip*-Movieclips platzieren Sie eine beliebige Bitmap-Grafik.



Schritt 1

ABBILDUNG 1.22

Aufbau des Flash-Films – lediglich eine Aktionen- und eine Inhalt-Ebene werden benötigt.

ABBILDUNG 1.23

Die Bibliothek hält die wesentlichen Bestandteile parat, vor allem die Movieclips werden zur Umsetzung benötigt.

**ABBILDUNG 1.24**

Hierarchie und Instanznamen im Überblick



Schritt 2

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

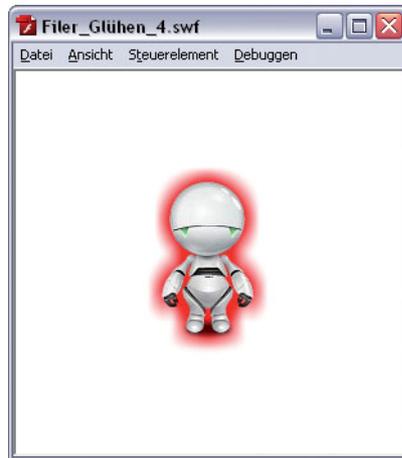
```

var richtung:Number = 1;

bild_mc.filters = [new flash.filters.GlowFilter ()];
bild_mc.blur = 10;
bild_mc.onEnterFrame = function (){
    bild_mc.blur += richtung;
    if ((bild_mc.blur >= 30) || (bild_mc.blur <= 10))
    {
        richtung *= -1;
    }
    var filterListe:Array = bild_mc.filters;
    filterListe[0].blurX = bild_mc.blur;
    filterListe[0].blurY = bild_mc.blur;
    bild_mc.filters = filterListe;
};

```

Sie können den Film testen und werden sehen, Ihr Objekt erstrahlt in einem vollkommenen neuen Licht.



Schritt 3

ABBILDUNG 1.25

Pulsierendes Gluehen

Bevel-Filter

Mithilfe der `BevelFilter`-Klasse können Sie für verschiedene Objekte in Flash einen Geschliffen-Effekt erzeugen. Durch diesen Geschliffen-Effekt erhalten die Objekte ein dreidimensionales Aussehen. Dabei haben Sie die Möglichkeit, diesen Effekt mithilfe von Hervorhebungen und Schattenfarben zu ändern. Darüber hinaus können Sie die Weichzeichnung, den Winkel und die Lage der geschliffenen Kanten und des Aussparungseffekts einstellen.

```
import flash.filters.BevelFilter;

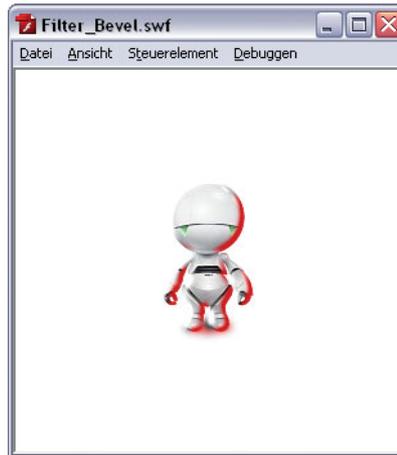
// Bevel-Filter definieren
var bevel:BevelFilter = new BevelFilter (4, 45, 0xFFFFFFFF, 1, 0xCC0000, 1,
10, 10, 2, 3);

// Eine neue bild_mc-Instanz erstellen
// Klicken mit der Maus verstärkt die Intensität
bild_mc.onPress = function ():Void {
    bevel.strength += 2;
    bild_mc.filters = [bevel];
};

// Listener erstellen, um den Filter beim Bewegen der Maus zu ändern
var mausListener:Object = new Object ();
mausListener.onMouseMove = function ():Void {
    bevel.distance = (_xmouse / Stage.width) * 10;
    bevel.blurX = (_ymouse / Stage.height) * 10;
    bevel.blurY = bevel.blurX;
    bild_mc.filters = [bevel];
};
Mouse.addListener (mausListener);
```

ABBILDUNG 1.26

Etwas Plastizität, kein Problem für den Bevel-Filter



Convolution-Filter

Die `ConvolutionFilter`-Klasse wendet einen Matrix-Convolution-Filtereffekt an. Bei einer *Convolution* werden die Pixel in einem von Ihnen angegebenen Eingabebild mit ihren benachbarten Pixeln kombiniert, um ein neues Bild zu erzeugen. Mit dem Convolution-Filter können zahlreiche Bildoperationen durchgeführt werden, einschließlich Weich- bzw. Scharfzeichnung, Kantenerkennung, Prägung und Schliff. Sie können den Convolution-Filter sowohl auf Bitmaps als auch auf Movieclip-Instanzen anwenden.

```
import flash.filters.ConvolutionFilter;
import flash.display.BitmapData;

this.createEmptyMovieClip ("bild_mc", 1);
bild_mc.createEmptyMovieClip ("halter_mc", 1);

var bild_mc1:MovieClipLoader = new MovieClipLoader ();
bild_mc1.loadClip ("bilder/space.jpg", bild_mc.halter_mc);

var matrixListe:Array = [1, 4, 6, 4, 1, 4, 16, 24, 16, 4, 16, 6, 24, 36,
24, 6, 4, 16, 24, 16, 4, 1, 4, 6, 4, 1];
var convolution:ConvolutionFilter = new ConvolutionFilter (5, 5,
matrixListe);
bild_mc.filters = [convolution];

var mausListener:Object = new Object ();
mausListener.onMouseMove = function ():Void {
    convolution.divisor = (_xmouse / Stage.width) * 300;
    convolution.bias = (_ymouse / Stage.height) * 150;
    bild_mc.filters = [convolution];
};
Mouse.addListener (mausListener);
```


ABBILDUNG 1.27

Convolution-Filter auf eine dynamische Bitmap angewandt

Farbmatrix-Filter

Mithilfe der `ColorMatrixFilter`-Klasse können Sie eine 4 x 5-Matrixtransformation auf den ARGB-Farb- und Alphawert jedes Pixels eines Eingabebilds anwenden, um ein neues ARGB-Farb- und Alphamuster zu erhalten. Der Filter ermöglicht beispielsweise das Drehen des Farbtons, das Ändern der Sättigung bzw. der Intensität eines bestimmten Farbtons, das Ändern der Luminanz bzw. Helligkeit oder Intensität einer Farbe in Alpha und weitere Effekte. Sie können den `ColorMatrixFilter` sowohl auf Bitmaps als auch auf Movieclip-Instanzen anwenden.

```
import flash.filters.ColorMatrixFilter;
import mx.transitions.Tween;
import mx.transitions.easing.*;

var mcl_obj:Object = new Object ();
mcl_obj.onLoadInit = function (ziel_mc:MovieClip):Void {
    // Movieclip-Instanz auf der Bühne zentrieren
    ziel_mc._x = (Stage.width - ziel_mc._width) / 2;
    ziel_mc._y = (Stage.height - ziel_mc._height) / 2;
    ziel_mc.watch ("helligkeit", helligkeitWatcher, ziel_mc);
    // den Movieclip ziel_mc in einem Helligkeitsbereich zwischen -100
    und +100 animieren
    var meinTween:Object = new Tween (ziel_mc, "helligkeit", Elastic.
    easeOut, 100, -100, 3, true);
    meinTween.onMotionFinished = function ()
    {
        this.yoyo ();
    };
};

this.createEmptyMovieClip ("bild_mc", 10);
var bild_mcl:MovieClipLoader = new MovieClipLoader ();
bild_mcl.addListener (mcl_obj);
bild_mcl.loadClip ("bilder/space5.jpg", bild_mc);
```

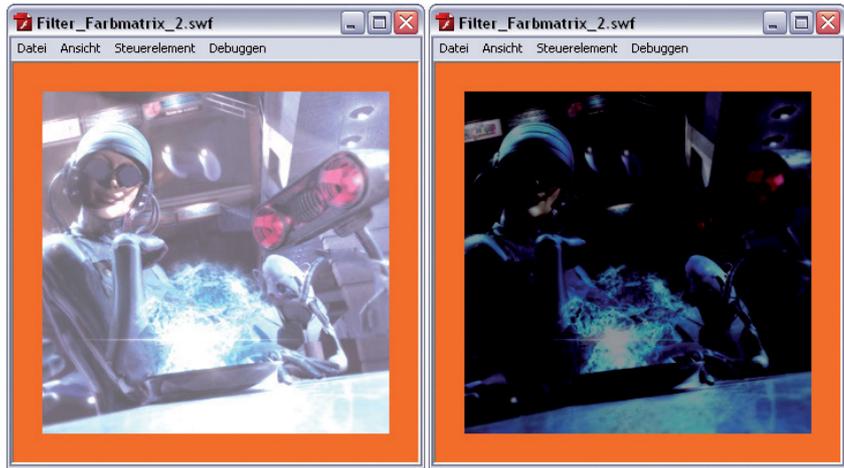
```

function helligkeitWatcher (prop:String, altWert:Number, neuWert:Number,
ziel_mc:MovieClip):Number {
    var helligkeitListe:Array = [1, 0, 0, 0, neuWert, 0, 1, 0, 0,
neuWert, 0, 0, 1, 0, neuWert, 0, 0, 0, 1, 0];
    ziel_mc.filters = [new ColorMatrixFilter (helligkeitListe)];
    return neuWert;
}

```

ABBILDUNG 1.28

Die Szene wirkt dank
Farbmatrixfilter mal hell und
mal dunkel.



Geschliffen-Filter

Mithilfe der `GradientBevelFilter`-Klasse können Sie auf Objekte in Flash einen geschliffenen Farbverlaufseffekt anwenden. Ein geschliffener Farbverlauf ist eine geschliffene Kante, die außen, innen und oberhalb eines Objekts durch Farbverläufe erweitert wurde. Geschliffene Kanten verleihen Objekten einen dreidimensionalen Effekt.

```

import flash.filters.GradientBevelFilter;

bild_mc._x = (Stage.width - bild_mc._width) / 2;
bild_mc._y = (Stage.height - bild_mc._height) / 2;

var farben:Array = new Array (0xFFFFFF, 0xCCCCCC, 0x000000);
var alphas:Array = new Array (1, 0, 1);
var ratios:Array = new Array (0, 128, 255);
var gradientBevel:GradientBevelFilter = new GradientBevelFilter (10, 45,
farben, alphas, ratios, 4, 4, 5, 3);

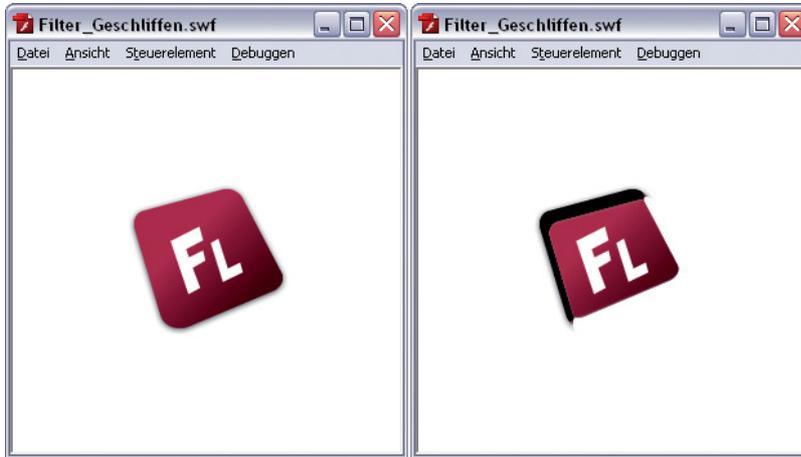
var mausListener:Object = new Object ();
mausListener.onMouseDown = function (){
    gradientBevel.strength++;
    bild_mc.filters = [gradientBevel];
};

```

```

mausListener.onMouseMove = function (){
    gradientBevel.blurX = (_xmouse / Stage.width) * 255;
    gradientBevel.blurY = (_ymouse / Stage.height) * 255;
    bild_mc.filters = [gradientBevel];
};
Mouse.addListener (mausListener);

```


ABBILDUNG 1.29

Die dritte Dimension oder nicht

Schlagschatten-Filter

Mithilfe der DropShadowFilter-Klasse können Sie Schlagschatten zu unterschiedlichen Flash-Objekten hinzufügen. Der Schattenalgorithmus basiert auf dem gleichen Box-Filter, den auch der Weichzeichnen-Filter verwendet. Für den Stil eines Schlagschattens stehen verschiedene Möglichkeiten zur Verfügung, beispielsweise innere und äußere Schatten sowie der Aussparungsmodus.

```

import flash.filters.DropShadowFilter;

var dropShadow:DropShadowFilter = new DropShadowFilter (4, 45, 0x000000,
0.8, 10, 10, 2, 2);

var mcl_obj:Object = new Object ();
mcl_obj.onLoadInit = function (ziel_mc:MovieClip):Void {
    ziel_mc._x = (Stage.width - ziel_mc._width) / 2;
    ziel_mc._y = (Stage.height - ziel_mc._height) / 2;
};

this.createEmptyMovieClip ("bild_mc", 10);
var bild_mcl:MovieClipLoader = new MovieClipLoader ();
bild_mcl.addListener (mcl_obj);
bild_mcl.loadClip ("bilder/FLA.png", bild_mc);

var mausListener:Object = new Object ();
mausListener.onMouseMove = function ():Void {

```

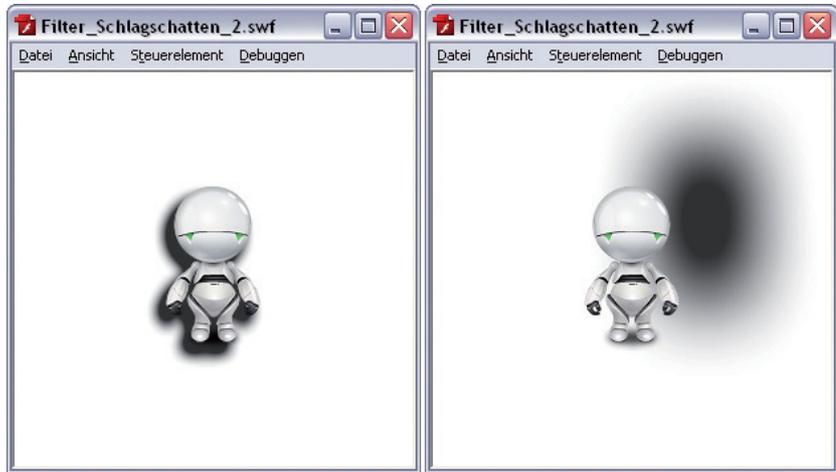
```

var p1:Number = bild_mc._y - _ymouse;
var p2:Number = bild_mc._x - _xmouse;
var degrees:Number = Math.atan2 (p1, p2) / (Math.PI / 180);
dropShadow.distance = Math.sqrt (Math.pow (p1, 2) + Math.pow (p2,
2)) * 0.5;
dropShadow.blurX = dropShadow.distance;
dropShadow.blurY = dropShadow.blurX;
dropShadow.angle = degrees - 180;
bild_mc.filters = [dropShadow];
};
Mouse.addListener (mausListener);

```

ABBILDUNG 1.30

Ein Schatten, der sich in Echtzeit verschieben lässt



Weichzeichnen-Filter

Mithilfe der `BlurFilter`-Klasse können Sie für verschiedene Objekte in Flash einen Weichzeichnungseffekt erstellen. Die Weichzeichnung verwischt die Einzelheiten eines Bilds. Dabei können Sie Weichzeichnungseffekte erstellen, die von einer geringfügig unscharfen Anzeige bis zu einem Gauss-Verwischen reichen. Der Weichzeichnen-Filter basiert auf einem Box-Pass-Weichzeichner. Der `quality`-Parameter definiert, wie oft der Weichzeichnungseffekt wiederholt werden soll, drei Anwendungen entsprechen in etwa einem Gauss-Verwischen à la *Adobe Photoshop* oder *Adobe Fireworks*.

```

import flash.filters.BlurFilter;
import mx.transitions.Tween;
import mx.transitions.easing.*;

this.createEmptyMovieClip ("halter_mc", 10);
halter_mc.createEmptyMovieClip ("bild_mc", 20);

var mcl_obj:Object = new Object ();
mcl_obj.onLoadInit = function (ziel_mc:MovieClip){
    ziel_mc._x = (Stage.width - ziel_mc._width) / 2;

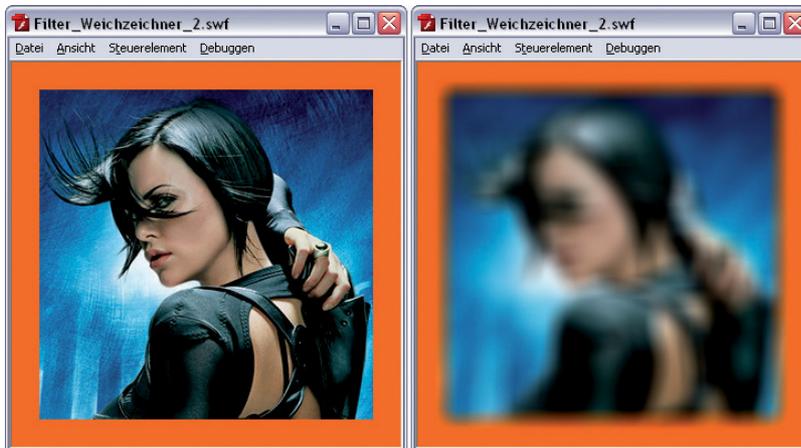
```

```

    ziel_mc._y = (Stage.height - ziel_mc._height) / 2;
    var myTween:Tween = new Tween (ziel_mc, "blur", Strong.easeInOut, 0,
20, 3, true);
    myTween.onMotionChanged = function () {
        ziel_mc._parent.filters = [new BlurFilter (ziel_mc.blur, ziel_
mc.blur, 1)];
    };
    myTween.onMotionFinished = function (){
        myTween.yoyo ();
    };
};

var bildLoader:MovieClipLoader = new MovieClipLoader ();
bildLoader.addListener (mcl_obj);
bildLoader.loadClip ("bilder/space3.jpg", halter_mc.bild_mc);

```

**ABBILDUNG 1.31**

Weichzeichnen wirkt, als würde man durch ein Milchglas schauen.

Verschiebungsmatrix-Filter

Die `DisplacementMapFilter`-Klasse führt mithilfe der Pixelwerte durch als Verschiebungsmatrix bezeichnete `BitmapData`-Objekte Verschiebungen bei Instanzen auf der Bühne aus, wie beispielsweise bei `Movieclip`- und `BitmapData`-Instanzen. Mit diesem Filter können Sie Instanzen verzerren oder mit einem Fleckenmuster versehen.

```

import flash.filters.DisplacementMapFilter;
import flash.geom.Point;
import flash.display.BitmapData;

var perlinBmp:BitmapData;
var displacementMap:DisplacementMapFilter;

var mcl_obj:Object = new Object ();
mcl_obj.onLoadInit = function (ziel_mc:MovieClip):Void {
    ziel_mc._x = (Stage.width - ziel_mc._width) / 2;
    ziel_mc._y = (Stage.height - ziel_mc._height) / 2;

```

```

        perlinBmp = new BitmapData (ziel_mc._width, ziel_mc._height);
        perlinBmp.perlinNoise (ziel_mc._width, ziel_mc._height, 10, Math.
round (Math.random () * 100000), false, true, 1, false);
        displacementMap = new DisplacementMapFilter (perlinBmp, new Point
(0, 0), 1, 1, 100, 100, "color");
        bild_mc.filters = [displacementMap];
    };

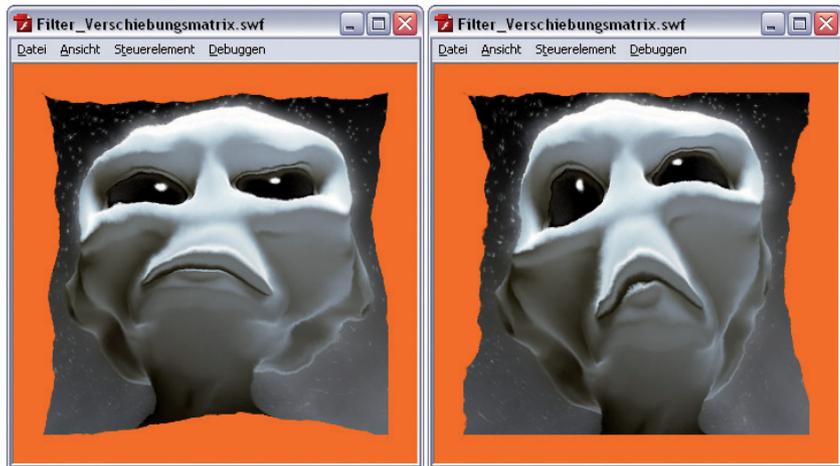
var bild_mc:MovieClip = this.createEmptyMovieClip ("bild_mc", 1);
bild_mc.createEmptyMovieClip ("halter_mc", 1);
var bildLoader:MovieClipLoader = new MovieClipLoader ();
bildLoader.addListener (mcl_obj);
bildLoader.loadClip ("bilder/space2.jpg", bild_mc.halter_mc);

var mausListener:Object = new Object ();
mausListener.onMouseMove = function ():Void {
    perlinBmp.perlinNoise (bild_mc._width, bild_mc._height, 10, Math.
round (Math.random () * 100000), false, true, 1, false);
    bild_mc.filters = [displacementMap];
};
Mouse.addListener (mausListener);

```

ABBILDUNG 1.32

Eine Echtzeit-Pixelverschiebung innerhalb von Bildern ist kein Problem.

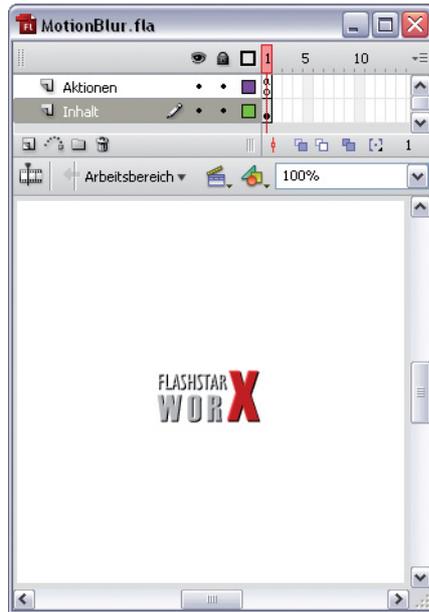


1.2.2 Simulation von MotionBlur-Bewegungen

Sie wollten schon immer eine Beschleunigung realistisch simulieren und haben bisher MotionBlur-Effekte lediglich durch das Hinzufügen unzähliger Movieclip-Instanzen realisieren können. Diese Zeiten gehören dank der BlurFilter-Klasse der Vergangenheit an.

Schritt 1

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *MotionBlur.fla* speichern. Sie brauchen zur Umsetzung eine MOVIECLIP-Instanz mit dem Instanznamen `click_mc`. Innerhalb des *FSClip*-Movieclips platzieren Sie die zu bewegenden Inhalte.

**ABBILDUNG 1.33**

Aufbau des Flash-Films – lediglich eine Aktionen- und eine Inhalt-Ebene werden benötigt.

**ABBILDUNG 1.34**

In der Bibliothek befindet sich der FSClip-Movieclip.

**ABBILDUNG 1.35**

Der Instanzname click_mc wird der Movieclip-Instanz zugewiesen.

Schritt 2

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
Stage.scaleMode = 'noScale';

import flash.geom.*;
import flash.display.BitmapData;
import flash.filters.BlurFilter;

click_mc._visible = false;

var maxBlur:Number = 40;
var easeWert:Number = .75;
var maxGroesse:Number = maxBlur + Math.sqrt(click_mc._width*click_mc._
width + click_mc._height*click_mc._height);
var offset:Number = maxGroesse/2;
var blur_bmp:BitmapData = new BitmapData(maxGroesse, maxGroesse, true,
0);

this.createEmptyMovieClip("blur_mc",1);
blur_mc.createEmptyMovieClip("image",1);
blur_mc.image.attachBitmap(blur_bmp, 1, false, true);
blur_mc.image._x = -offset;
blur_mc.image._y = -offset;

var rotate_matrix:Matrix = new Matrix();

var blur_rect:Rectangle = blur_bmp.rectangle;
var blur_punkt:Point = new Point(0, 0);
var blur_filter:BlurFilter = new BlurFilter(0,0);

var ziel_Punkt:Point = new Point(click_mc._x, click_mc._y);

onEnterFrame = function(){
    var alt_Punkt = new Point(blur_mc._x, blur_mc._y);
    var neu_Punkt = Point.interpolate(alt_Punkt, ziel_Punkt, easeWert);

    blur_mc._x = neu_Punkt.x;
    blur_mc._y = neu_Punkt.y;

    var abstand = Point.distance(alt_Punkt, neu_Punkt);
    var winkel = berechneWinkel(alt_Punkt, neu_Punkt);

    rotate_matrix.identity();
    rotate_matrix.rotate(-winkel);
    rotate_matrix.translate(offset, offset);

    blur_bmp.fillRect(blur_bmp.rectangle, 0);
    blur_bmp.draw(click_mc, rotate_matrix);
    blur_filter.blurX = Math.min(maxBlur, abstand*1.5);
    blur_bmp.applyFilter(blur_bmp, blur_rect, blur_punkt, blur_filter);
    blur_mc._rotation = winkel * 180/Math.PI;
```

```
}  
  
onMouseDown = function(){  
    ziel_Punkt = new Point(_xmouse, _ymouse);  
}  
  
function berechneWinkel(p1:Point, p2:Point):Number{  
    var dx = p2.x - p1.x;  
    var dy = p2.y - p1.y;  
    return Math.atan2(dy, dx);  
}  
}
```

Sie können den Film testen. Wie Sie sehen steht einem interessanten *MotionBlur*-Effekt nichts mehr im Wege. Verantwortlich hierfür ist die *BlurFilter*- und die *BitmapData*-Klasse.



Schritt 3

ABBILDUNG 1.36

MotionBlur in Pura-Natura

ABBILDUNG 1.37

Wie wäre es mit einer Instanz, die Sie samt MotionBlur durch die Gegend werfen können?

MotionBlur-Spuren

Wir haben noch ein interessantes Skript zur Darstellung von *MotionBlur*-Spuren umgesetzt, das wir Ihnen nicht vorenthalten wollen. Um das Skript anwenden zu können, benötigen Sie lediglich ein MOVIECLIP-Symbol mit dem Verknüpfungsbezeichner `kugel`.

```
Stage.scaleMode = 'noScale';

// Tempo der Kugeln
var kugelTempo = 3;
// Anzahl der Kugeln
var kugelAnzahl = 20;
// Weichzeichner
var spurBlur = 0;

// Weichzeichner-Filter
var blur = new flash.filters.BlurFilter(spurBlur, spurBlur, 1);
// Erzeugen eines Hintergrund-Bitmap
var bg_bmp = new flash.display.BitmapData(300,300,false,0xFF000000);

// Movieclip-Instanzen, welche die Kugeln (haupt_mc)
// und das Hintergrundbild (background_mc) enthalten
this.createEmptyMovieClip("haupt_mc", 2);
this.createEmptyMovieClip("background_mc", 1);

// Hintergrund-Bitmap der Movieclip-Instanz (background_mc) zuweisen
// und mit dem Weichzeichner versehen
background_mc.attachBitmap(bg_bmp,1);
background_mc.filters = [blur];

// Kugel erzeugen
function erzeugeKugeln(){
    var i = kugelAnzahl;
    var kugel, winkel;

    while(i--){
        // Kugel in der Bühnenmitte platzieren
        kugel = haupt_mc.attachMovie("kugel", "kugel"+i, i, {_x:
Stage.width/2, _y:Stage.height/2});
        // Zufallswinkel zuweisen
        winkel = Math.random()*Math.PI*2;
        kugel.bx = Math.cos(winkel)*kugelTempo;
        kugel.by = Math.sin(winkel)*kugelTempo;
        // Ereignis (onEnterFrame) zuweisen
        kugel.onEnterFrame = bewegeKugeln;
    }
}

// Kugeln bewegen
function bewegeKugeln(minX,maxX,minY,maxY){
    this._x += this.bx;
    this._y += this.by;
```

```

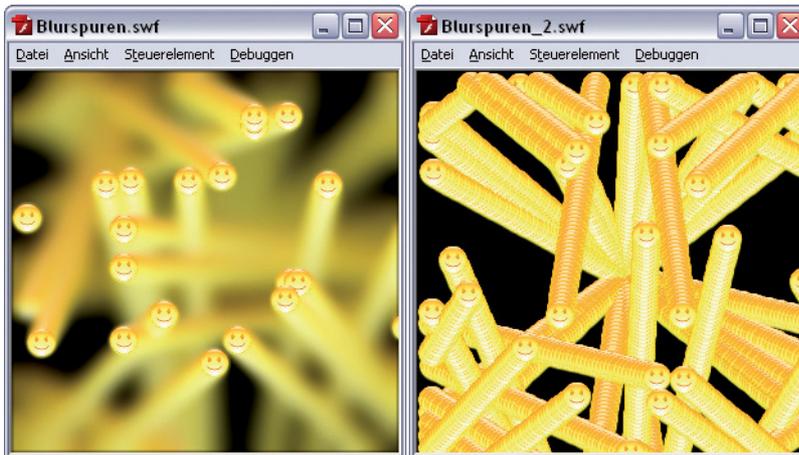
// Links/Rechts-Bewegungs-Beschränkung
if (this._x > 300) this._x = 0;
else if (this._x < 0) this._x = 300;

// Oben/Unten-Bewegungs-Beschränkung
if (this._y > 300) this._y = 0;
else if (this._y < 0) this._y = 300;
}

// Hintergrund-Bitmap aktualisieren (neu zeichnen),
// um die Spuren darzustellen
onEnterFrame = function(){
    bg_bmp.draw(this);
}

// Kugeln erzeugen (Auslöser)
erzeugeKugeln();

```


ABBILDUNG 1.38

*MotionBlur-Spuren,
auch kein Problem*

Auswahlfokus

Wie wäre es abschließend noch mit einem Auswahlfokus, der sich hervorragend für Flash-Anwendungen eignet, die dem Nutzer eine gezielte Auswahl von Objekten ermöglichen. Um das Skript anwenden zu können, benötigen Sie lediglich ein SCHALTFLÄCHEN-Symbol mit dem Instanznamen `erzeuge_btn` und ein MOVIECLIP-Symbol mit dem Verknüpfungsbezeichner `figur`.

```
Stage.scaleMode = 'noScale';

var zielfigur = null;
var figurliste = new Array();
var ausgangspunkt = new flash.geom.Point(150, 150);
var figurgrenzen = new flash.geom.Rectangle(-150, -150, 300, 300);
var blur = new flash.filters.BlurFilter(0, 0, 1);

function focusClips(maxdistanz, focuseasefactor, blurstaerke){
    var i = figurliste.length;
    var diff;
    if (focus == undefined) focus = 0;
    if (!zielfigur.getDepth())
    {
        return false;
    }
    focus += (zielfigur.getDepth()-focus)/focuseasefactor;
    while (i--){
        diff = Math.abs(focus-figurliste[i].getDepth());
        blur.blurX = blur.blurY=blurstaerke*diff/maxdistanz;
        figurliste[i].filters = [blur];
    }
}

function erzeugeFiguren(figuranzahl){
    entferneFiguren();
    var i = figuranzahl;
    while (i--){
        figurliste.push(erzeugeZufallFigur(i, 250, 1000));
    }
    figurliste[0].onPress();
    focusClips(1000, 8, 15);
}

function entferneFiguren(){
    var i = figurliste.length;
    while (i--){
        figurliste[i].removeMovieClip();
    }
    figurliste.length = 0;
}

function erzeugeZufallFigur(tiefe, fokalelaenge, maxdistanz){
    var figur = this.attachMovie("figur", "figur"+tiefe, tiefe);
    var xpos = figurgrenzen.x+Math.floor(Math.random()*figurgrenzen.width);
    var ypos = figurgrenzen.y+Math.floor(Math.random()*figurgrenzen.
height);
    var zpos = Math.floor(Math.random()*maxdistanz);
    var scaleratio = fokalelaenge/(fokalelaenge+zpos);
    figur._x = ausgangspunkt.x+xpos*scaleratio;
```

```
    figur._y = ausgangspunkt.y+ypos*scaleratio;
    figur._xscale = figur._yscale=100*scaleratio;
    var tiefe = maxdistanz-zpos;
    while (this.getInstanceAtDepth(tiefe))
    {
        tiefe++;
    }
    figur.swapDepths(tiefe);
    figur.onRollOver = figurRollOver;
    figur.onPress = figurPress;
    figur.onRollOut = figurRollOut;
    return figur;
}

function figurRollOver(){
    this.gotoAndStop(2);
}

function figurPress(){
    if (zielfigur)
    {
        zielfigur.gotoAndStop(1);
    }
    this.gotoAndStop(2);
    zielfigur = this;
}

function figurRollOut(){
    if (zielfigur != this)
    {
        this.gotoAndStop(1);
    }
}

erzeugen_btn.onRelease = function(){
    erzeugeFiguren(10);
};
onEnterFrame = function(){
    focusClips(1000, 8, 15);
};
erzeugeFiguren(10);
```

ABBILDUNG 1.39

Auswahlfokus und schon steht einer gezielten Auswahl nichts mehr im Weg.

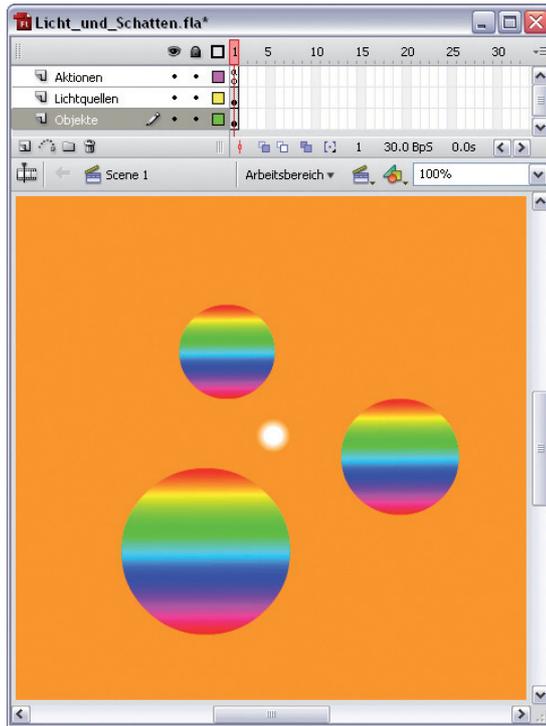


1.2.3 Simulation von Lichtquellen

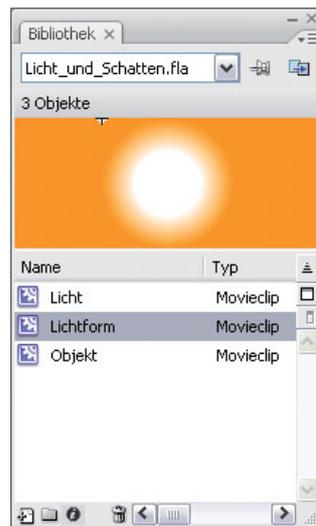
Die Simulation von Lichtquellen wird durch die Kombination aus *Bevel-* und *Schlagschatten-*Filtern ermöglicht.

Schritt 1

Wir benötigen als Erstes eine ActionScript 2.0 Flash-Datei, diese sollten Sie unter dem Namen *Licht_und_Schatten fla* speichern. Sie benötigen zur Umsetzung vier *MOVIECLIP*-Instanzen, denen Sie die Instanznamen *licht_mc*, *obj1*, *obj2* und *obj3* zuweisen. Innerhalb des *Licht*-Movieclips platzieren Sie eine Instanz des *LichtForm*-Movieclips und innerhalb des *Objekt*-Movieclips platzieren Sie einen beliebigen Inhalt.

**ABBILDUNG 1.40**

Aufbau des Flash-Films – in der Hauptzeitleiste sind sowohl die Lichtquelle als auch die beleuchteten Objekte enthalten.

**ABBILDUNG 1.41**

Sämtliche wesentlichen Bestandteile befinden sich innerhalb der Bibliothek.

ABBILDUNG 1.42

Hierarchie und Instanznamen im Überblick


Schritt 2

Im ersten Schlüsselbild der Zeitleiste platzieren Sie in der *Aktionen*-Ebene folgende ActionScript-Codezeilen.

```
function setzeSchatten(zentrum){
    px = zentrum._x
    py = zentrum._y;
    lx = licht_mc._x;
    ly = licht_mc._y
    if (lx<px && ly<py) {
        quelle = 1;
    } else if (lx>=px && ly<=py){
        quelle = 2;
    } else if (lx<=px && ly>=py){
        quelle = 3;
    } else if (lx>=px && ly>=py){
        quelle = 4;
    }
    dx = Math.abs(lx-px);
    dy = Math.abs(ly-py);
    dz = Math.sqrt((dx*dx)+(dy*dy))

    winkel = Math.atan2(dx, dy);
    winkel = Math.round(winkel/Math.PI*180);
    switch (quelle)
    {
        // Links/Oben
        case 1:
            dropShadow.angle = -1*winkel+90;
            kante.angle = -1*winkel+90;
            break;
        // Rechts/Oben
        case 2:
            dropShadow.angle = winkel+90;
            kante.angle = winkel+90;
            break;
        // Links/Unten
        case 3:
            dropShadow.angle = winkel+270;
            kante.angle = winkel+270;
            break;
```

```
// Rechts/Unten
case 4:
    dropShadow.angle = -1*winkel+270;
    kante.angle = -1*winkel+270;
    break;
}
dropShadow.alpha = Math.abs((dz/1200)-1)-.2;
kante.shadowAlpha = Math.abs((dz/1200)-1)-.2;
kante.highlightAlpha = Math.abs((dz/600)-1)-.2;
dropShadow.distance = dz/15;
dropShadow.blurX = dz/15;
dropShadow.blurY = dz/30+5;
zentrum.filters = [dropShadow, kante]
}

lichtKontrolle = setInterval(setzeSchatten, 33, obj1)
lichtKontrolle2 = setInterval(setzeSchatten, 33, obj2)
lichtKontrolle2 = setInterval(setzeSchatten, 33, obj3)

function setzeDragger(obj) {
    obj.onPress = function() {
        startDrag(obj)
    }
    obj.onRelease = function() {
        stopDrag();
    }
    obj.onReleaseOutside = function() {
        stopDrag();
    }
}

setzeDragger(obj1)
setzeDragger(obj2)
setzeDragger(obj3)
setzeDragger(licht_mc)

// Bevel-Filter
var kante = new flash.filters.BevelFilter();
kante.blurY = 2
kante.blurX = 2
kante.knockout = false
kante.strength = 1
kante.quality = 2
kante.shadowAlpha = 0
kante.highlightAlpha = 1
kante.angle = 35
kante.distance = 3;

// DropShadow-Filter
var dropShadow = new flash.filters.DropShadowFilter();
dropShadow.blurX = 4;
dropShadow.blurY = 4;
dropShadow.distance = 6;
dropShadow.angle = 35;
```

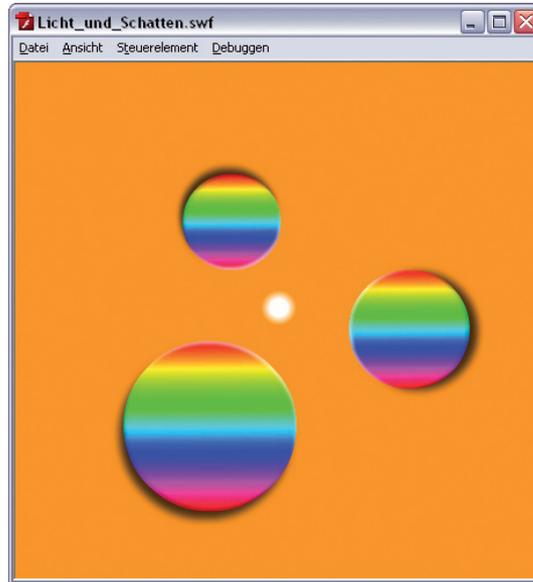
Schritt 3

```
dropShadow.quality = 2;  
dropShadow.alpha = 1;  
  
zentrum.filters = [dropShadow];
```

Sie können den Film testen und nun heißt es, die Lichtquelle oder die Objekte ins rechte Licht zu rücken.

ABBILDUNG 1.43

Lichtquelle platzieren und genießen

**ABBILDUNG 1.44**

Auch die Objekte lassen sich verschieben, versuchen Sie es einmal!

