Sönke Kluth, Frederik Schricker,
Philipp Kyeck, Aron Woost

Flash CS4

Die Workshops für Einsteiger und Fortgeschrittene



Galileo Press

Web 2.0-Reflexion

Movieclips mit einem nahtlosen Übergang und Alphaverlauf spiegeln

Apple hat damit angefangen, und alle haben es kopiert: Die Spiegelung mit Alphaverlauf. In diesem Workshop lernen Sie, eine Spiegelung dynamisch auf einen Movieclip anzuwenden, auch wenn sich in dem Movieclip animiertes Material befindet.

Ziel

Den beliebten Spiegel-Effekt nutzen

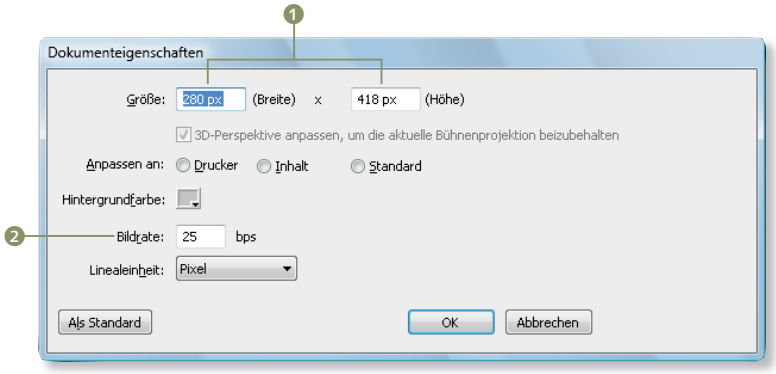
- BitmapData-Objekt anwenden
- Mit einer Matrix arbeiten
- Einen Alphaverlauf dynamisch erstellen

[**Ergebnis:** *10_ Reflexion/spiegeleffekt.swf*]



1 Dokument erstellen

Erstellen Sie ein neues AS3-Flash-Dokument mit der Grösse 280×418 Pixel ❶ und einer Bildrate ❷ von 25 bps. Speichern Sie dieses Dokument unter dem Namen »Reflection.fla« ab.



◂ **Abbildung 1**
Über Modifizieren • Dokument legen Sie die Dokumenteigenschaften fest.

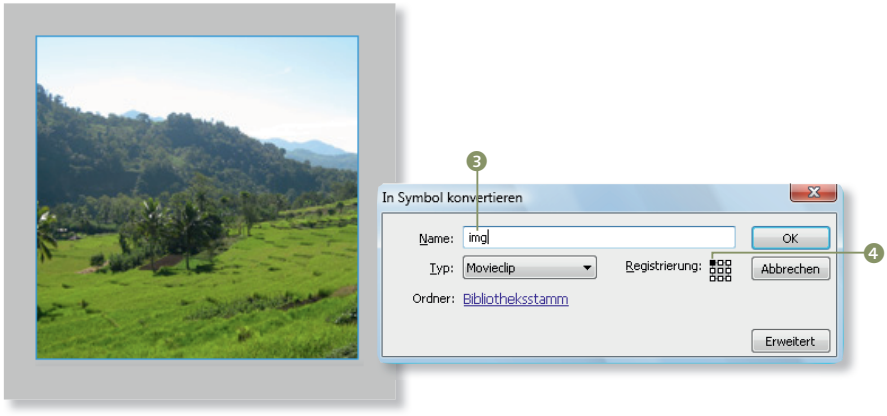
2 Quell-Movieclip erstellen

Erstellen Sie zunächst einen Movieclip, der gespiegelt werden soll. Importieren Sie dazu ein beliebiges Bild, oder erstellen Sie mit Hilfe der Flash-Werkzeuge eine Grafik. Beachten Sie, dass Ihr Movieclip höchstens 280×209 Pixel groß sein sollte, weil die Reflexion bei der eingestellten Bühnengröße sonst nicht mehr komplett sichtbar ist. In diesem Beispiel nutzen wir das Foto »img.jpg«.

Konvertieren Sie Ihre Grafik(en) mit Modifizieren • In Symbol konvertieren oder mit F8 in ein Symbol.

Geben Sie dem Movieclip in dem sich nun öffnenden Dialog den Namen »img« ❸. Achten Sie darauf, dass die Registrierung links oben ❹ liegt.

Anschließend vergeben Sie im Eigenschaftenfenster den Instanznamen »img«.



◂ **Abbildung 2**
Das Foto wird in einen Movieclip konvertiert und der Movieclip bekommt den Instanznamen »img.

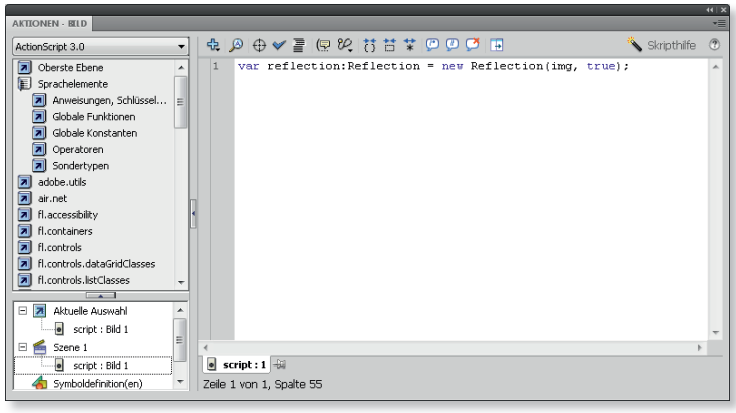
Reflexionsklasse aufrufen 3

Im Flash-Film selbst wird zunächst die (noch zu erstellende) Reflexionsklasse dynamisch von der Zeitleiste aus aufgerufen. Die Reflexionsklasse wird also nicht als Dokumentklasse erstellt oder fest mit einem Movieclip verbunden, sondern kann von jeder Zeitleiste eines (noch so verschachtelten) Movieclips instanziiert werden.

Erstellen Sie auf der Hauptzeitleiste eine neue Ebene mit dem Namen »script«. Öffnen Sie über FENSTER • AKTIONEN oder mit F9 bzw. Alt+F9 den ActionScript-Editor. Fügen Sie hier folgende Zeile hinzu:

```
var reflection:Reflection = new Reflection(img, true);
```

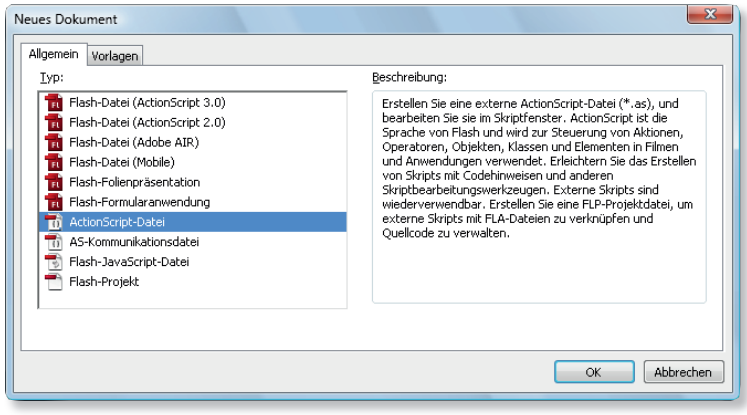
Abbildung 3 ▶
Die Reflexionsklasse wird aufgerufen.



Reflexionsklasse anlegen 4

Erstellen Sie über DATEI • NEU • ACTIONSCRIPT-DATEI eine neue Klasse. Speichern Sie diese über DATEI • SPEICHERN UNTER... als »Reflection.as« in demselben Ordner wie die .fla-Datei ab.

Abbildung 4 ▶
Eine neue ActionScript-Datei anlegen



5 Klassengrundgerüst

Fügen Sie folgenden Code in die neu erstellte Klasse ein:


Zwischenergebnis auf der Buch-DVD: *08_Reflexion/Reflection_Step05.as*

```
1: package
2: {
3:    import flash.display.Sprite;
4:    import flash.display.MovieClip;
5:    import flash.display.BitmapData;
6:    import flash.display.Bitmap;
7:    import flash.display.Shape;
8:    import flash.geom.Point;
9:    import flash.events.Event;
10:   import flash.geom.Matrix;
11:   public class Reflection extends Sprite
12:   {
13:      private var w:Number;
14:      private var h:Number;
15:      private var img:MovieClip;
16:      private var original_bd:BitmapData;
17:      private var rotate_matrix:Matrix;
18:      private var rotate_bd:BitmapData;
19:      private var output_bd:BitmapData;
20:      private var gradient_bd:BitmapData;
21:      private var display_bmp:Bitmap;
22:      public function Reflection(mc:MovieClip, video:Boolean =
         false)
23:      {
24:      }
25:   }
26: }
```

In den Zeilen 3 bis 10 importieren Sie die für das Skript benötigten Klassen. Die erforderlichen Variablen deklarieren Sie in den Zeilen 13 bis 21.

Dem Konstruktor in Zeile 22 wird bei Aufruf mindestens ein Parameter übergeben, nämlich der Movieclip, der reflektiert werden soll. Als zweiter optionaler Parameter `video` kann festgelegt werden, ob sich in dem Movieclip ein Video oder animierter Inhalt befindet, der in jedem Frame ausgelesen werden muss.

Mit `video:Boolean = false` geben Sie im Konstruktor an, dass der Parameter, wenn er nicht übergeben wird, `false` ist.

Referenz und Größe speichern 6


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step06.as*

Fügen Sie im Konstruktor nun folgenden Code hinzu:

```
img = mc;
w = mc.width;
h = mc.height;
```

Für die Erstellung des Spiegeleffektes benötigen Sie später die Pixelinformationen des Ausgangs-Movieclips. In der ersten Zeile speichern Sie eine Referenz auf den im Konstruktor als ersten Parameter übergebenen Movieclip ab.

Für die Berechnungen benötigen Sie die Größe des zu reflektierenden Movieclips. Dessen Breite und Höhe speichern Sie in den beiden folgenden Zeilen.

Quellbild als BitmapData speichern 7


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step07.as*

Um die Bitmapdaten des Quellbildes für das Skript verfügbar zu machen, schreiben Sie diese in ein BitmapData-Objekt. Ergänzen Sie Ihren Code dazu direkt unter den in Schritt 6 eingefügten Codezeilen wie folgt:

```
original_bd = new BitmapData(w, h);
original_bd.draw(mc);
```

In der ersten Zeile erzeugen Sie das BitmapData-Objekt `original_bd`, das die Größe des Quell-Movieclips hat.

Ein Abbild des Quell-Movieclips zeichnen Sie in der zweiten Zeile in die Variable `original_bd`.

Rotationsmatrix erstellen 8


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step08.as*

Natürlich muss die Spiegelung des Bildes auf dem Kopf stehen. Zu diesem Zweck erstellen Sie ein Matrix-Objekt. Ergänzen Sie im Konstruktor folgende Codezeilen:

```
rotate_matrix = new Matrix();
rotate_matrix.scale(1, -1);
rotate_matrix.translate(0, h);
```

Zuerst erstellen Sie das neue Matrix-Objekt `rotate_matrix` .

In der nächsten Zeile skalieren Sie dieses mit dem Faktor -1 in der Vertikalen. Die Matrix wird also »auf den Kopf« gestellt.

Anschließend müssen Sie die Matrix mit Hilfe von `translate` um die Höhe des Quell-Movieclips in der Vertikalen nach unten verschieben. Ansonsten läge sie genau über dem Quell-Movieclip – und eine Spiegelung liegt ja immer unter einem Objekt.

9 Rotations-Bitmap-Data erstellen

Nun drehen Sie die `original_bg`. Ergänzen Sie folgende Zeilen ganz unten im Konstruktor:

```
rotate_bd = new BitmapData(w, h, false, 0);
rotate_bd.draw(original_bd, rotate_matrix);
```

In der ersten Zeile erstellen Sie ein neues BitmapData-Objekt mit den Maßen des Ausgangsbildes.

Jetzt zeichnen Sie `original_bd` unter Angabe der `rotate_matrix` in `rotate_bd`. In `rotate_bd` liegt das Bild jetzt in gedrehter und verschobener Version vor.


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step09.as*

10 BitmapData-Objekt für die Ausgabe erstellen

Fügen Sie Ihrem Skript folgende Zeilen hinzu (wieder ganz unten im Konstruktor, unter den eben eingefügten Codezeilen):

```
output_bd = new BitmapData(w, h, true, 0);
output_bd.copyPixels(rotate_bd, rotate_bd.rect, new Point(0, 0));
```

Zuerst erstellen Sie ein neues BitmapData-Objekt `output_bd`, das die Breite und Höhe des Quell-Movieclips hat. Der dritte Parameter (`true`) legt fest, dass das BitmapData-Objekt Transparenzen unterstützen soll, da Sie der Spiegelung später einen transparenten Verlauf hinzufügen werden.

Anschließend kopieren Sie die Bilder in der zweiten Zeile von `original_bd` in `output_bd`.


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step10.as*

11 Ausgabe-Bitmap erstellen

Nun benötigen Sie noch ein Bitmap-Objekt, das die Darstellung der Bilddaten übernimmt. Ergänzen Sie Ihren Code wie folgt:

```
1: display_bmp = new Bitmap();
2: display_bmp.bitmapData = output_bd;
3: display_bmp.y = h;
4: addChild(display_bmp);
```

Erstellen Sie dieses Bitmap-Objekt in der ersten Zeile, und weisen Sie es in Zeile 2 an, die Bitmapdaten von `output_bd` anzuzeigen.

Außerdem verschieben Sie `display_bmp` in Zeile 3 auf die y-Position, die exakt der Unterkante des Quell-Movieclips entspricht.

Mit `addChild()` in Zeile 4 sorgen Sie dafür, dass das Bitmap-Objekt auf der Bühne dargestellt wird.


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step11.as*

Reflexionsklasse anzeigen 12

Nun muss noch die Reflexionsklasse selbst angezeigt werden. Ergänzen Sie (wieder direkt unter den eben eingefügten Codezeilen) folgende Anweisung:

```
mc.parent.addChild(this);
```

Mit dem `addChild`-Befehl, der in derselben Hierarchie-Ebene wie der Quell-Movieclip aufgerufen wird (`mc.parent`), zeigen Sie `display_bmp` auf der Hauptzeitleiste an.

Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step12.as*

Flash-Film testen 13

Wenn Sie Ihr Dokument nun testen, wird unterhalb des Quell-Movieclips die Reflexion angezeigt.

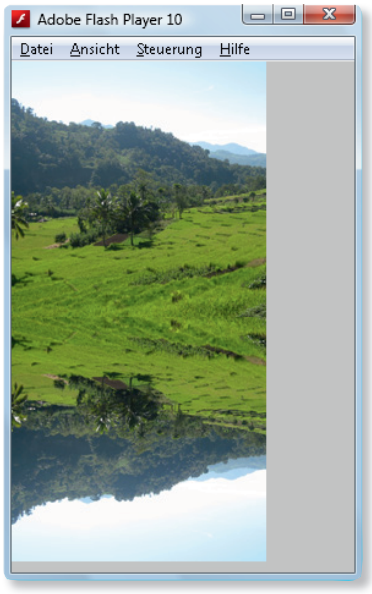


Abbildung 5 ▸
Das Bild wird gespiegelt.

Alphaverlauf hinzufügen 14

Die Reflexion soll einen Verlauf beinhalten, der von 100 % Alpha (also voll sichtbar) nach 0 % (unsichtbar) ausfadet.

Dazu erstellen Sie zunächst eine Funktion, die Ihnen einen Verlauf zurückgibt. Ergänzen Sie Ihr Skript unter dem Konstruktor um folgenden Code:

```
1: private function createGradientBitmapData(w:Number, h:Number):
   Shape
2: {
3:    var colors:Array = [0x000000, 0x000000];
```

Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step14.as*

```
4:   var fillType:String = "linear";
5:   var alphas:Array = [100, 0];
6:   var ratios:Array = [0, 0xFF];
7:   var rot:Number = 90 * Math.PI / 180;
8:   var matrix:Matrix = new Matrix();
9:   matrix.createGradientBox(w, h, rot, 0, 0);
10:  var b:Shape = new Shape();
11:  b.graphics.beginGradientFill(fillType, colors, alphas,
     ratios, matrix);
12:  b.graphics.moveTo(0, 0);
13:  b.graphics.lineTo(w, 0);
14:  b.graphics.lineTo(w, h);
15:  b.graphics.lineTo(0, h);
16:  b.graphics.lineTo(0, 0);
17:  b.graphics.endFill();
18:  return b;
19: }
```

In Zeile 1 legen Sie den Funktionsnamen `createGradientBitmapData` fest. Die Methode erwartet die zwei Parameter `w` und `h`. Dabei handelt es sich um die Breite und die Höhe des zu erstellenden Verlaufes. Die Methode gibt ein `Shape`-Objekt zurück, das später von einem BitmapData-Objekt gezeichnet wird.

Die Farben des Verlaufes legen Sie in Zeile 3 fest. Sie sind beliebig, da Sie nur die Alphawerte des Verlaufes benötigen.

Den Typ definieren Sie in Zeile 4 als linearen Verlauf. Die Alphawerte legen Sie in Zeile 5 fest. Diese sollen von 100 % bis 0 % reichen.

Die Variable `ratios` in Zeile 6 gibt an, dass der gesamte Farbraum von 0 bis 255 genutzt werden soll.

Außerdem muss der Verlauf rotiert werden, da er sich standardmäßig von links nach rechts vollzieht. Die Rotationsvariable legen Sie in Zeile 7 fest. Sie muss im Bogenmaß angegeben werden, weshalb die Gradzahl 90 mit Pi/180 multipliziert werden muss.

In Zeile 8 erstellen Sie dann eine neue Matrix, der Sie in Zeile 50 die Flächenangaben für den Verlauf übergeben.

Den Shape, der von der Funktion zurückgegeben wird, erstellen Sie in Zeile 10. In Zeile 11 weisen Sie den Shape an, eine Verlaufszeichnung zu beginnen, die die zuvor erstellten Parameter hat.

Die weiteren Zeilen 12 bis 17 sind für die Zeichnung des Verlaufs zuständig.

In Zeile 18 schließlich wird der soeben gezeichnete Verlauf als Shape-Objekt zurückgegeben.

Umrechnung
Sie können einen Winkel, der im Bogenmaß angegeben ist, wie folgt in das Gradmaß umrechnen:

*Grad = Radiant * (180/Pi)*

Um einen Winkel, der im Gradmaß angegeben ist, in das Bogenmaß umzurechnen, verwenden Sie folgende Formel:

*Radiant = Grad * (Pi/180)*

Tipp
Ausführliche Informationen über die Zeichnen-Engine von Flash finden Sie in der Flash-Hilfe.

Alphaverlauf anwenden 15

Nun muss der Code zur Erstellung der Spiegelung folgendermaßen ab Zeile 34 angepasst werden. Ersetzen Sie folgenden Code aus Schritt 10

```
output_bd = new BitmapData(w, h, true, 0);
output_bd.copyPixels(rotate_bd, rotate_bd.rect, new Point(0, 0));
```


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step15.as*

mit folgenden Codezeilen:

```
1: gradient_bd = new BitmapData(w, h, true, 0);
2: gradient_bd.draw(createGradientBitmapData(w, h));
3: output_bd = new BitmapData(w, h, true, 0);
4: output_bd.copyPixels(rotate_bd, rotate_bd.rect, new Point(0, 0),
   gradient_bd, new Point(0, 0), true);
5: display_bmp = new Bitmap();
6: ...
```

In Zeile 1 erstellen Sie ein BitmapData-Objekt mit Namen `gradient_bd`, das wieder die Breite und Höhe des Quell-Movieclips hat.

Innerhalb der `draw`-Methode in Zeile 2 rufen Sie `createGradientBitmapData()` auf und übergeben dieser Methode die Breite und Höhe. Das zurückgegebene Shape-Objekt wird in `gradient_bd` gespeichert.

Anschließend muss noch die `copyPixels`-Funktion in Zeile 4 angepasst werden. Hier geben Sie als vierten Parameter das eben erstellte `gradient_bd` an und legen mit dem sechsten Parameter (`true`) fest, dass für den Kopierprozess die Alphadaten von `gradient_bd` herangezogen werden sollen.

Wenn Sie Ihr Dokument nun exportieren, werden Sie sehen, dass auf der Spiegelung ein Alphaverlauf liegt.

Videoaktualisierung hinzufügen 16

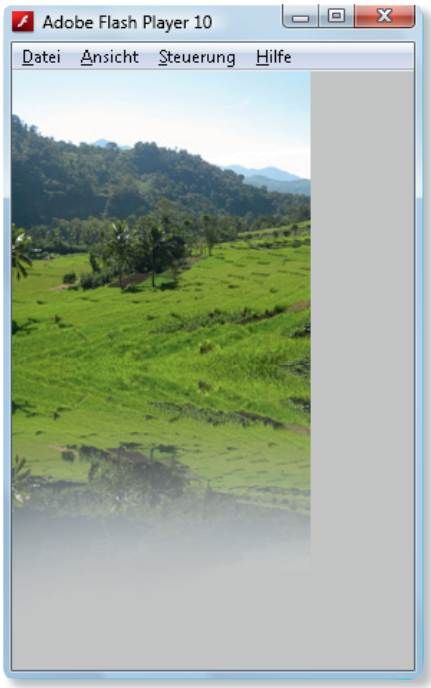
Jetzt werden Sie noch Vorkehrungen treffen für den Fall, dass sich in dem Quell-Movieclip Bewegtbilder befinden.

Zunächst müssen Sie sich beim `ENTER_FRAME`-Event anmelden. Fügen Sie folgende `if`-Anweisung ans Ende des Konstruktors ein (Zeile 43):

```
if(video){
   img.addEventListener(Event.ENTER_FRAME, onImgEnterFrame);
}
```


Zwischenergebnis auf der Buch-DVD: *10_Reflexion/Reflection_Step16.as*

Wenn im Konstruktor die `video`-Variable als `true` übergeben wurde (siehe Zeile 22 in Schritt 5), meldet sich das Skript beim `ENTER_FRAME`-Event an. Von nun an wird in jedem Frame die `onImgEnterFrame`-Methode in Ihrem Skript aufgerufen.



◀ **Abbildung 6**
Die Web 2.0-Spiegelung ist fertig.

Diese Methode erstellen Sie nun. Ergänzen Sie Ihr Skript unter dem Konstruktor um folgende Funktion (ab Zeile 47):

```
1: private function onImgEnterFrame(evt:Event):void
2: {
3:     original_bd.draw(img);
4:     rotate_bd.draw(original_bd, rotate_matrix);
5:     output_bd.fillRect(output_bd.rect, 0x00000000);
6:     output_bd.copyPixels(rotate_bd, rotate_bd.rect, new Point(0,
       0), gradient_bd, new Point(0, 0), true);
7: }
```

Die Zeilen 3 und 4 verhalten sich exakt wie im Konstruktor in den Zeilen 33 und 38.

Sie setzen den Bildinhalt von `output_bd` in Zeile 5 zurück, indem sie die Fläche mit der Farbe `0x00000000` füllen.

Zeile 6 verhält sich genau wie Zeile 37 im Konstruktor.

Damit das korrekte Verhalten getestet werden kann, sollte sich im Quell-Movieclip natürlich auch animierter Inhalt befinden. Im Beispiel auf der Buch-DVD bewegt sich, zu Testzwecken, einfach ein Kreis innerhalb des Bildes.

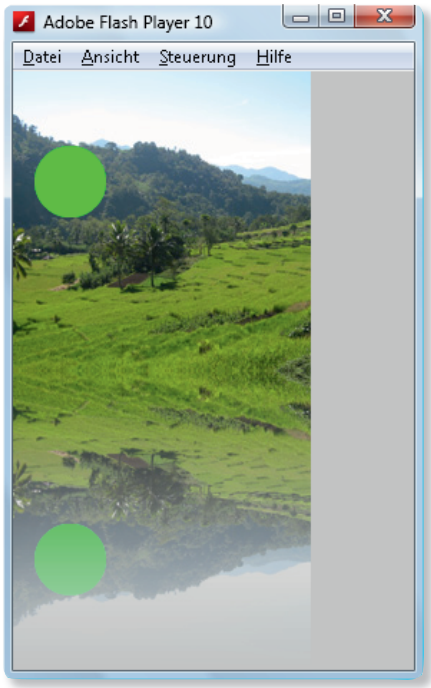


Abbildung 7 ▸
Auch animierte Inhalte können jetzt gespiegelt werden.