

Flash-Film selbst gedreht

Flash führt unter Linux ein Schattendasein, grafische Entwicklungsumgebungen gibt es nicht. Die Ming-Bibliothek hilft jedoch dabei, eigene Flash-Filme zu programmieren. Wie es mit PHP geht, zeigt der Artikel. Klaus Rechert



Befehl »php-config --includes« den Ort der PHP-Headerdateien liefert. Im Falle eines Fehlers kann dieser auch manuell über die Umgebungsvariable »PHPSRC« angegeben werden, zum Beispiel mit:

```
export PHPSRC=/usr/src/php-4.3
```

Die Befehle »make« und »make install« besorgen die Übersetzung und Installation in bekannter Weise. Folgende Zeile in der »php.ini« bindet das Modul in die PHP-Installation ein und sorgt dafür, dass es automatisch geladen wird.

```
extension=php_ming.so
```

Alternativ lädt in PHP-Skripten die Anweisung »dl("php_ming.so")« das Modul dynamisch.

Ein einfacher Film

Jeder Flash-Film beginnt damit, ein »SWFMovie«-Objekt zu erzeugen (siehe **Listing 1**, Zeile 3). Dieses Objekt stellt Methoden bereit, um zuerst allgemeine Filmeigenschaften zu definieren wie Hintergrundfarbe, SWF-Version und Framerate (Zeilen 4 bis 6). Sind die grundlegenden Eigenschaften definiert, fügt Zeile 12 dem ersten Frame mit der »add()«-Methode neue Elemente hinzu. Diese Methode nimmt ein SWF-Objekt als Argument und liefert ein »SWFDisplayItem« zurück, siehe **Tabelle 1**.

Ein »SWFDisplayItem«-Objekt repräsentiert eindeutig jenes Filmelement, das sich gerade auf der Displayliste befindet und so lange sichtbar bleibt, bis man es wieder entfernt. Wird das gleiche Element einem Film mehrfach hinzugefügt, steht dafür jedes Mal ein neues, eindeutiges Display-Item-Objekt. Die Metho-

den, die das Display-Item-Objekt bereitstellt, können angezeigte Filmelemente nachträglich manipulieren, etwa drehen, verschieben, verzerren und skalieren. Die Funktion »setName()« weist Elementen eindeutige Namen zu, um sie über »ActionScript« anzusprechen.

Alle Modifikationen von Display-Items laufen im Kontext eines Frame ab. Ruft das Skript die »nextFrame()«-Methode auf (**Listing 1**, Zeile 15), werden die Änderungen am Film wirksam und ein neuer Frame steht bereit. Der ist zunächst identisch mit dem vorigen, da sich die Displayliste durch den Vorgang nicht verändert. Sollen Filmelemente ab einem bestimmten Zeitpunkt verschwinden, muss der Programmierer sie mit der »remove()«-Methode des »SWFMovie«-Objekts entfernen (Zeile 21).

Abschließend kann der Film auf zwei Arten ausgegeben werden. Mit PHP besteht die Möglichkeit, den entsprechenden HTTP-Header zu setzen (Zeile 24) und dann den fertigen Flash-Film direkt ohne Zwischenspeichern auszugeben (Zeile 25). Alternativ schreibt die Library den Film als SWF-Datei.

Um Platz und Bandbreite zu sparen, lassen sich Flash-Filme seit SWF-Version 6 komprimieren. Die »save()«- und die »output()«-Methode verarbeiten die Kompressionsrate als optionalen Parameter. Erlaubt sind Werte von -1 (keine Kompression) bis 9 (Maximalwert). **Abbildung 1** zeigt das sichtbare Ergebnis im Browser.

Grafikoperationen

Die einfachsten Elemente von Flash-Filmen sind die »SWFShape«-Objekte, mit denen sich einfache Umrisse zeichnen lassen. Ein neues »SWFShape« wird

Das Format von Flash-Dateien ist kein Geheimnis. 1998 gab Macromedia die Spezifikationen frei und machte damit alternative Implementierungen von Autorensoftware für Flash möglich, die Ming-Bibliothek ist ein Beispiel dafür. Sie ist in der Lage, mit wenigen Ausnahmen standardkonforme SWF-Dateien bis einschließlich Version 6 und 7 zu erstellen. Die Bibliothek ist in C geschrieben, bietet aber Schnittstellen für die populären Skriptsprachen PHP und Perl sowie für C++ und Java.

Installation

Die Ming-Sourcen können von der Projektseite als gepacktes Tar-Archiv heruntergeladen werden **[1]**. Ming benötigt die Bibliotheken Libungif **[2]**, Libpng **[3]** und die Programme Flex und Bison, für die Übersetzung des PHP-Moduls zusätzlich die PHP-Headerdateien. Vor dem Kompilieren des PHP-Moduls sollte man zunächst überprüfen, ob der

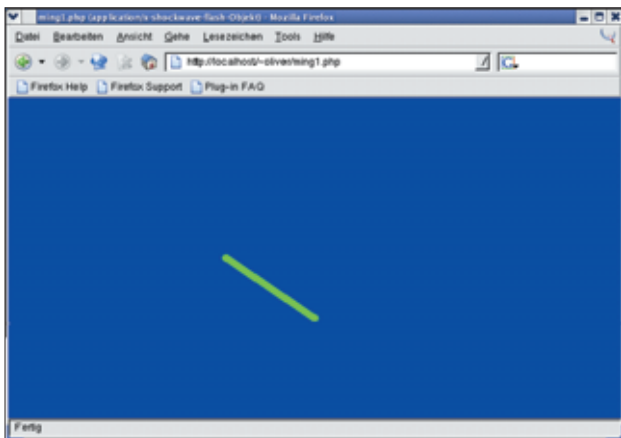


Abbildung 1: Das Beispiel aus Listing 1 lässt eine Linie um einen ihrer Endpunkte rotieren. Die Darstellung übernimmt hier das Flash-Plugin des Firefox-Browsers.

ohne Parameter erzeugt und bietet eine Reihe von Methoden, um Vektorgrafiken zu erstellen. Die Methode »setLine()« bestimmt Stifffarbe und -breite. Es gibt Methoden, deren Aufgabe es ist, den Stift zu positionieren (»movePen()«, »movePenTo()«), außerdem zum Zeichnen von Linien (»drawLine()«, »drawLineTo()«, »drawCurveTo()«) und Kreisen (»drawCircle()«). Komplexere Geometrien setzen sich aus solchen einfachen Grundformen zusammen.

Sollen ausgefüllte Flächen entstehen, muss der Flash-Programmierer zunächst ein »SWFFill«-Objekt erzeugen, und zwar mit der »addFill()«-Methode des »SWFShape«-Objekts, die dem Objekt einen neuen Füllstil zuweist. Sie erwartet als Argument entweder einen RGB-Wert mit optionalem Alphawert, ein »SWFBitmap«-Objekt, das für eine Bitmap-Grafikdatei steht, oder ein »SWFGradient«-Objekt zur Darstellung von Farbverläu-

eines Linienzugs gefüllt werden soll. So füllt beispielsweise »setLeftFill()« das Innere eines Objekts genau dann aus, wenn es gegen den Uhrzeigersinn gezeichnet wurde.

Formveränderung

Der fließende Übergang zweier Objekte ineinander, das so genannte Morphing, funktioniert mit zwei Shape-Objekten. Ein neu erzeugtes »SWFMorph«-Objekt enthält zwei leere Shape-Objekte, eins beschreibt den Anfangszustand, das andere den Endzustand. Zugriff auf die beiden Shape-Objekte erhält man mit Hilfe der Methoden »getShape1()« beziehungsweise »getShape2()«.

Sind beide Shape-Objekte fertig gestaltet, kann das Morph-Objekt dem Film hinzugefügt werden. Den aktuellen Zustand des Verlaufs beschreibt das »Ratio«-Feld des Morph-Objekts. Zu Beginn

fen. Zurück gibt »addFill()« dann ein »SWFFill«-Objekt.

Die beiden Methoden »setLeftFill()« und »setRightFill()« des SWFShape-Objekts versehen einen existierenden Füllstil mit einer Richtung und wenden ihn auf einen Umriss an. Die Füllrichtung entscheidet darüber, welche Seite

steht es auf »0«, was den unveränderten Zustand des Ausgangsobjekts beschreibt. Der Wert »1« repräsentiert den endgültigen Zustand des Zielobjekts, alle Werte dazwischen einen Zwischenzustand beim Übergang des Ausgangsobjekts in das Zielobjekt.

Der Übergang der beiden Objekte ineinander lässt sich also steuern, indem man den Wert des Ratio-Felds mit »setRatio()« kontinuierlich verändert. Listing 2 zeigt ein Skript, das einen Kreis in einen Punkt überführt.

Textelemente einbauen

Die Darstellung von Textobjekten setzt Fonts im Flash-internen Format FDB voraus. TrueType-Fonts (TTF) sind also, bevor sie in einem Flash-Film erscheinen, in zwei Schritten in dieses Format umzuwandeln. Für Schritt eins findet sich auf der Ming-Website [1] das Programm »ttf2fft«, das die TTF-Dateien in das Zwischenformat FFT konvertiert:

```
ttf2fft -e 1 Truetype.ttf -o Outfile.fft
```

Die verfügbaren Schriftschnitte der TTF-Datei zeigt:

```
ttf2fft -l Truetype.ttf
```

Listing 1: »simple.php«

```
01 <?
02
03 $movie = new SWFMovie();
04 $movie->setBackground(0x0,0x00,0xff);
05 $movie->setRate(2);
06 $movie->setDimension(400,300);
07
08 $shape = new SWFShape();
09 $shape->setLine(5, 0x0, 0xff, 0x00);
10 $shape->drawLine(50,50);
11
12 $dItem = $movie->add($shape);
13 $dItem->moveTo(200,200);
14
15 $movie->nextFrame();
16 for($deg = 0; $deg < 360; $deg += 10) {
17     $dItem->rotateTo($deg);
18     $movie->nextFrame();
19 }
20
21 $movie->remove($dItem);
22 $movie->nextFrame();
23
24 header('Content-type: application/x-shockwave-flash');
25 $movie->output();
26 ?>
```

Tabelle 1: SWF-Objekte

Objekt	Aufgabe
SWFShape	Container für primitive Grafikobjekte wie Linien, Kreise und Polygone
SWFBitmap	Objekt für eine Grafikdatei im DBL-, GIF- oder JPEG-Format
SWFSound	Objekt für eine Musikdatei im MP3-Format
SWFSoundStream	Enthält eine Musikdatei im MP3-Format als Stream
SWFFont	Repräsentant für eine Font-Datei im FDB-Format
SWFText	Stellt eine Zeichenkette im gegebenen Font dar
SWFVideoStream	Objekt zur Darstellung von Videostreams
SWFAction	Repräsentant für in Byte-Code kompilierten Actionscript-Programmcode; der Konstruktor erhält den Quelltext
SWFFill	Repräsentant für einen Füllstil
SWFGradient	Erzeugt einen Farbverlauf
SWFMorph	Repräsentant eines fließenden Übergangs zweier Shape-Objekte
SWFButton	Klickbares Objekt mit Event-Modell

Im zweiten Schritt macht das Programm »makefdb« (im Unterverzeichnis »util« zu finden) aus der FFT-Datei einen FDB-Font, den Ming dann in den Flash-Film einbaut:

```
makefdb Outfile.fff
```

Auf der Ming-Webseite steht schon eine Auswahl von FDB-Fonts für den Download zur Verfügung.

Eine Alternative zu diesen Dateien sind Browser-Fonts. Der Client, der den Flash-Film vom Server lädt, verwendet in diesem Fall eigene Schriften. Diese Option wirkt sich zwar vorteilhaft auf die resultierende Dateigröße des Flash-Films aus, da die Font-Daten nicht enthalten sind, dafür hängt sie aber von der Client-Konfiguration ab. Die Fonts »_serif«, »_sans« und »_typewriter« sollten aber bei allen Browsern und Plattformen vorhanden sein.

Schriften kontrollieren

Das »SWFFont«-Objekt repräsentiert in einem Flash-Film einen spezifischen Font. Der Objektkonstruktor erwartet die Angabe des Namens einer FDB-Datei oder eines Browser-Fonts. Einmal erstellte Font-Objekte können anschließend für beliebig viele Text-Objekte verwendet werden.

Ein »SWFText«-Objekt definiert die Darstellung einer Zeichenkette. Nach dem Festlegen von Font und darzustellendem Text mit Hilfe von »setFont()« und »add-

String()«, lassen sich noch weitere Darstellungseigenschaften definieren. Das Objekt bietet außerdem Methoden, um Textfarbe und -größe zu setzen.

Ebenfalls zur Darstellung von Text geeignet, aber insbesondere für die Eingabe konzipiert, sind die »SWFTextField«-Objekte. Sie bieten nicht so viel Flexibilität wie Text-Objekte, da sie nicht gedreht oder asymmetrisch skaliert werden können, eignen sich aber für Benutzereingaben. »TextField«-Objekte akzeptieren optionale Parameter, die das Erscheinungsbild und das Verhalten des Eingabefelds beeinflussen, siehe [Tabelle 2](#).

Alle Parameter lassen sich durch Oder-Verknüpfungen kombinieren. Eingabefelder brauchen wie Text-Objekte einen Font (»setFont()«-Methode), eventuell einen Text (»addString()«) und verarbeiten genauso Schriftfarbe und -größe. Hingegen können nur Eingabefelder den Text mit »align()« ausrichten oder einem Variablennamen zuordnen, um den Inhalt des Formulars zu versenden oder mit Actionscript zu bearbeiten.

Interaktive Elemente

Flash-Filme eignen sich nicht nur zur Darstellung statischer Inhalte, sondern auch für interaktive Multimedia-Präsentationen und sogar zur Programmierung kleiner Spiele. Die einfachsten interaktiven Elemente heißen SWF-Buttons. Ist ein neues Button-Objekt erzeugt, können ihm für verschiedene Zustände Shape-Objekte zugeordnet werden. Die »addShape()«-Methode erwartet neben einem Shape-Objekt noch ein oder mehrere durch Oder verknüpfte Zustandskonstanten. Mögliche Zustände oder Events zeigt [Tabelle 3](#).

Die dem Button zugeordneten Shape-Objekte werden angezeigt, wenn die ihnen zugeordneten Ereignisse eintreten. Die Ausnahme ist »SWFBUTTON_HIT«, dessen einzige Aufgabe es ist, die aktive Region zu markieren. Diese Region wird durch die Ausmaße des Shape-Objekts bestimmt, das mit diesem Zustands-Attribut versehen ist.

Events steuern nicht nur die Darstellung der Buttons, sondern lösen auch Actionscript-Programmcode aus, die Skript-Programmiersprache für Flash-Filme. In dem PHP-Programm repräsentiert ein

»SWFAction«-Objekt diesen Code, der vor der Ausführung in Bytecode zu übersetzen ist. Bei der hier beschriebenen PHP-Variante kümmert sich der Objekt-Konstruktor um den Übersetzungsvorgang. Der Programmierer übergibt lediglich den auszuführenden Programmcode im Quellcode als Argument. Dem Button liefert er daraufhin mit der »addAction()«-Methode das »SWFAction«-Objekt. Diese Methode erwartet zusätzlich eine oder mehrere der Event-Konstanten, die der **Kasten „Event-Konstanten“** aufführt.

Ein etwas komplexeres Beispiel interaktiver Elemente benutzt ein »SWFSprite«. Das komplette Listing dazu findet sich auf dem Server des Linux-Magazins [\[4\]](#). Ein Sprite ist eine Art eigenständiger Flash-Film mit eigener Zeitachse, verhält sich aber sonst genauso wie andere Flash-Objekte. Da ein Sprite einem Movie-Objekt ähnlich ist, besitzt es auch ähnliche Methoden. So können dem Sprite neue Flash-Elemente mit »add()« hinzugefügt und mit »remove()« wieder aus ihm entfernt werden. Die Funktion »nextFrame()« baut ein neues Frame in das Sprite ein.

Das Beispiel realisiert mit einem Sprite-Objekt eine visuelle Suche innerhalb eines Flash-Films. Durch Klicken und Ziehen springt der Benutzer an beliebige Stellen im Film. Da ein Sprite eine eigene Zeitachse besitzt, kann ein Skript, während das Sprite angeklickt ist, periodisch dessen Position abfragen und den darunter liegenden Film zu einem bestimmten Frame springen lassen. Dabei werden die Events nicht einem Button zugeordnet, sondern einem DisplayItem-Objekt.

Soundtrack

Der SWF-Standard kennt zwei Arten, um Audiodaten zu integrieren. Zum einen den Event-Sound, geeignet für kürzere Sequenzen, die periodisch oder aufgrund eines Ereignisses abgespielt werden. Zum anderen den Streaming-Sound, der dafür geeignet ist, einen ganzen Film mit Audiodaten zu unterlegen, insbesondere wenn Synchronisation zwischen Grafik und Ton wichtig sind. Da bei Flash-Filmen der Client viel rechnen muss, kann es sein, dass der Flash-

Listing 2: »morph.php«

```
01 $morph = new SWFMorph();
02
03 $shape1 = $morph->getShape1();
04 $shape2 = $morph->getShape2();
05
06 $shape1->setLine(2, 0xff, 0x0, 0x0);
07 $shape1->movePenTo(200, 200);
08 $shape1->drawCircle(100);
09
10 $shape2->setLine(1, 0x0, 0x0, 0xff);
11 $shape2->movePenTo(200, 200);
12 $shape2->drawLine(1, 1);
13
14 $dMorph = $movie->add($morph);
15
16 for($r=0.0; $r<=1.0; $r+=0.02) {
17     $dMorph->setRatio($r);
18     $movie->nextFrame();
19 }
```

Tabelle 2: »TextField«-Parameter

Name	Bedeutung
SWFTEXTFIELD_NOEDIT	Eingabefeld ist nicht editierbar
SWFTEXTFIELD_PASSWORD	Passwortfeld
SWFTEXTFIELD_DRAWBOX	Erzeugt einen Rahmen um das Eingabefeld
SWFTEXTFIELD_MULTILINE	Mehrzeiliges Eingabefeld
SWFTEXTFIELD_WORDWRAP	Erlaubt Zeilenumbruch
SWFTEXTFIELD_NOSELECT	Verhindert die Auswahl des Eingabefelds

Player vorübergehend die vorgegebene Framerate nicht einhalten kann, zum Beispiel wenn die Berechnung eines Frame sehr aufwändig ist. Läuft parallel dazu Streaming-Sound, verzichtet der Player auf das Zeichnen einzelner Frames zugunsten einer synchronen Audio- und Grafik-Wiedergabe.

Ming unterstützt leider noch nicht alle Sound-Optionen. Streaming-Sound ist zur Zeit nur im MP3-Format möglich. Event-Sound unterstützt ebenfalls das MP3-Format, zusätzlich ist Audio unkomprimiert oder im ADPCM-Format erlaubt. Streaming-Sound integriert man durch die »streamMP3()«-Methode des Movie-Objekts.

Event-Sound wird durch ein »SWF-Sound«-Objekt repräsentiert, das als Parameter den Namen der Audiodatei erwartet. Die Methode »startSound()« akzeptiert dieses »SWFSound«-Objekt wiederum als Argument und baut es in den Film ein. Sie gibt außerdem noch ein »SWFSoundInstance«-Objekt zurück, mit dem die Eigenschaften beim Abspielen der Datei beeinflussbar sind. Anhalten lässt sich die Ausgabe mit der Methode »stopSound()«.

Videostreaming

Seit SWF Version 6 verarbeitet Flash Videodaten, die allerdings in dem nicht freien Sorenson-H.263-Format vorliegen müssen. Ming erwartet das Video als FLV-Datei, einem von Macromedia ge-

Event-Konstanten

SWFBUTTON_MOUSEOVER
SWFBUTTON_MOUSEOUT
SWFBUTTON_MOUSEUP
SWFBUTTON_MOUSEUPOUTSIDE
SWFBUTTON_MOUSEDOWN
SWFBUTTON_DRAGOUT
SWFBUTTON_DRAGOVER

Träger für Audio- und Videostreams dient. Mehr Informationen zu FLV bietet die Macromedia-Website [5].

Videodaten sind entweder in der Flash-Datei enthalten oder werden von externen Quellen als Stream geladen. Das »SWFVideoStream«-Objekt repräsentiert das Video und bildet die Schnittstelle, um es zu steuern. Bekommt der Konstruktor einen Dateinamen als Parameter übergeben, baut Ming die Videodaten in den zu erzeugenden Film ein. Dabei ist zu beachten, dass die Framerate des Flash-Films mit der des Videofilms übereinstimmt, da sich immer nur ein Videoframe pro Flash-Frame schreiben lässt:

```
$stream = new SWFVideoStream(fopen
    ("video.flv", "rb"));
$stream->setDimension(100,200);
$item = $movie->add($stream);
for($i = 0; $i < $stream->getNumFrames();
    $i++)
    $movie->nextFrame();
```

Wer die Videodaten nicht in die Datei schreiben will, kann sie auch dynamisch laden. Flash-Player 7 bietet erstmals die Möglichkeit des Progressive Downloads an. In diesem Fall kann die FLV-Datei getrennt von Flash-Film gespeichert sein, der sie dann erst bei Bedarf lädt. Hierzu ist ein leeres »SWFVideoStream«-Objekt notwendig, das als Platzhalter für die Videodaten dient. Da zu diesem Zeitpunkt die Auflösung der Videodaten noch unbekannt ist, muss der Programmierer diese Größe mit »setDimension()« bestimmen. Actionscript-Programmcode lädt und steuert dann die Videodaten, siehe Listing 3.

Echtes Streaming ist mit Macromedias Flash Communication Server möglich, den es auch für Linux gibt [6]. Ohne Lizenz läuft der Server zwar in einem eingeschränkten Modus mit weniger Bandbreite und Streams, reicht für die Entwicklung aber aus. Der Actionscript-Code sieht gegenüber dem obigen Bei-

Tabelle 3: SWF-Events

Name	Bedeutung
SWFBUTTON_UP	Button wurde losgelassen
SWFBUTTON_OVER	Zeiger befindet sich über dem Button
SWFBUTTON_DOWN	Button wurde geklickt
SWFBUTTON_HIT	Markiert die klickbare Region

spiel dann anders aus. Die »connect«-Methode bekommt als Parameter die URL des Servers übergeben, etwa:

```
nc.connect("rtmp://localhost/test/audio");
```

Ausblick

Auch wenn es keine grafische Entwicklungsumgebung für Flash unter Linux gibt, lassen sich mit der Ming-Bibliothek recht einfach eigene Flash-Filme erzeugen. Die Multimedia-Fähigkeiten werden zudem laufend erweitert, sodass eigenen Experimenten mit Audio- und Videostreaming nichts im Weg steht. (ofr) ■

Infos

- [1] Ming-Homepage: [\[http://ming.sourceforge.net\]](http://ming.sourceforge.net)
- [2] Libpng: [\[http://www.libpng.org/pub/png/libpng.html\]](http://www.libpng.org/pub/png/libpng.html)
- [3] Libungif: [\[http://sourceforge.net/projects/libungif\]](http://sourceforge.net/projects/libungif)
- [4] Listings zum Artikel: [\[http://www.linux-magazin.de/Service/Listings/2004/08/Flash\]](http://www.linux-magazin.de/Service/Listings/2004/08/Flash)
- [5] Flash FLV Video: [\[http://www.macromedia.com/support/flash/ts/documents/video_resources.htm\]](http://www.macromedia.com/support/flash/ts/documents/video_resources.htm)
- [6] Flash Server: [\[http://macromedia.com/software/flashcom\]](http://macromedia.com/software/flashcom)

Listing 3: Events für Streaming

```
01 $script = " \
02     stop(); \
03     nc=new NetConnection(); \
04     nc.connect(null); \
05     NewStream=new NetStream(nc); \
06     video1.attachVideo(NewStream); \
07     NewStream.setBufferTime(10); \
08     NewStream.play("video.flv"); \
09 ";
10
11 $action = new SWFAction($script);
12 $stream = new SWFVideoStream();
13 $video_item = $movie->add($stream);
14 $video_item->setname("video1");
15 $action_item = $movie->add($action);
```