

Actionscript

caad:arch:ethz

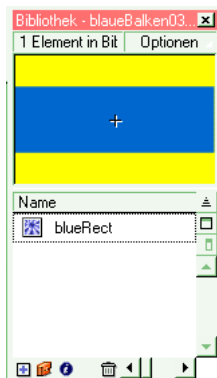
Symbole und Instanzen

Es gibt zwei Arten programmierbarer Symbole: Filmsequenzen und Buttons.
Erstellen eines Symbols:

- Im Menü 'Einfügen' -> 'Neues Symbol' ein neues Symbol erzeugen.
- Symbolnamen eintragen und als Verhalten Filmsequenz oder Button eintragen.



- Das Symbol kann nun im Editor gezeichnet werden. Es besitzt eine eigene Zeitleiste und eigene Layers.
- Über 'Fenster' -> 'Bibliothek' wird die Bibliothek mit allen bisher erzeugten Symbolen aufgerufen.
- Wenn man im Editor zurück zu Szene 1 geht, kann man die selbstgestellten Symbole mit drag and drop in den aktuellen Film ziehen.
- Die Objekte, die man in den Film gezogen hat nennt man Instanzen der Symbole aus der Bibliothek
- Man kann auch Symbole in andere Symbole hineinziehen, und somit verschachteln.
- Um Symbole im nachhinein zu verändern, muss man sie in der Bibliothek doppelklicken.



Szene bearbeiten Symbol bearbeiten



Schaltflächen- (Button-) Symbol

Button-Symbole besitzen im Gegensatz zu Filmsequenz-Symbolen vier sichtbare Zustände:
Auf - So sieht die Schaltfläche im Normalzustand aus.
Darüber - So wird die Schaltfläche, wenn man mit der Maus darüber fährt.
Drücken - So sieht die Schaltfläche aus, wenn man sie drückt
Aktiv - beschreibt den (unsichtbaren) Bereich in dem der Schalter den Mauszeiger registriert.
Button-Symbole sind in der Lage, auf Mausclicks zu reagieren und dadurch Aktionen auszulösen.

programmieren von SchaltflächenSymbolen

Schaltflächen eignen sich dafür, auf Knopfdruck Filme anzuhalten oder abzuspielen.
Bsp.:
Erzeugt ein Schaltflächen-Symbol und zieht eine Instanz aus der Bibliothek auf die Arbeitsfläche.
Wählt die Instanz an und öffnet das 'Aktionen' Fenster.
Wählt bei 'Basisaktionen' 'Go To'.
Im Skripteditor erscheint folgender Text:

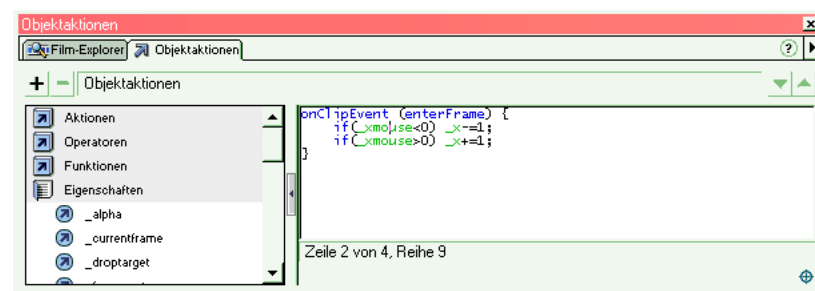
```
on (release){
    gotoAndPlay(1);
}
```

Instanzen vom Typ Schaltfläche werden durch die on()-Methoden gesteuert.
In den runden Klammern (...) steht, WANN die Anweisungen ausgeführt werden
Im Befehlsblock zwischen den geschweiften Klammern { } stehen die Anweisungen, die ausgeführt werden sollen.
Durch Anklicken kann man die Eigenschaften der Aktionen ändern:
gotoAndPlay(1) anklicken und bei Bild '2' eingeben -> Der Film springt nun zu Bild 2.
on(release) anklicken und die Markierung bei 'loslassen' löschen und bei 'darüberrollen' setzen
-> gotoAndPlay(2) wird nun ausgeführt, wenn der Mauszeiger über die Schaltfläche rollt.

programmieren in Flash MX

Flash bietet nicht nur die Möglichkeit, Trickfilme und Navigationen mittels Filmdrehbuch zu erstellen. mit der Skriptsprache ActionScript ist man in der Lage, selbst erstellte Symbole mit intelligenten Aktionen zu belegen. Actionscript wurde seit Flash 5.0 syntaktisch der Programmiersprache JavaScript angeglichen.
In einem Flashprojekt kommen die Scriptaktionen an zwei Stellen zum Einsatz.
Bilder innerhalb der Zeitleiste können mit Aktionen belegt werden (Bildaktionen/Frameactions)
Selbst erzeugte Symbole (Buttons und Movieclips) lassen sich programmieren.
Gezeichnete Elemente (Polygone, Rechtecke, etc.) müssen zuerst in Symbole umgewandelt werden bevor man sie mit einem Script belegt.

Der ActionScript Editor



über 'Fenster' -> 'Aktionen' ruft man den Editor für die Actionskripte auf.
Der Titel des Fensters zeigt an, ob man geraden Object- oder Bildaktionen bearbeitet.
Objektaktionen werden durch Klick auf eine Symbol-Instanz auf der Arbeitsfläche (Szene1) editiert.
Bildaktionen bearbeitet man durch Klick auf ein Bild in der Zeitleiste.

Es gibt zwei Bearbeitungsmodi um Actionskripte zu erstellen:
Sie werden rechts oben eingestellt.
'Normaler Modus': Hier baut man die Skripte mit der Maus zusammen.
Links werden die Objektaktionen oder die zu bearbeitenden Eigenschaften ausgewählt. Je nach Art des Symbol (Filmsequenz oder Button) sind nur bestimmte Aktionen anwählbar. Für die ausgewählten Aktionen werden im Editorfenster Skripte erstellt. Klickt man die Zeilen im Skript an, kann man einzelne Parameter verändern.
'Expertenmodus': Hier werden die Skripte über die Tastatur eingegeben.

Editorfenster: Hier werden die Skripte angezeigt und editiert.
_x Symboleigenschaften erscheinen immer grün
Vordefinierte Befehle und Funktionen sind blau
//... Kommentare sind violett
"..." Texte sind Hellgrau
eigen Variablen und Funktionen sind schwarz.

Filmsequenz- (Movieclip-) Symbol

Das Filmsequenz-Symbol gleicht in der Struktur der Ebenen und Zeitleisten einem kompletten Flashfilm. Prinzipiell ist jeder Flashfilm den man erstellt wie ein grossen Filmsequenz-Symbol welches alle anderen Symbole beinhaltet.
Mittels Actionscript kann man die Eigenschaften der Filmsequenz-Symbole ändern, z.B. Position, Skalierung, Farbe, etc.

programmieren von FilmsequenzSymbolen

Bsp.:
Erzeugt ein Filmsequenz-Symbol und zieht eine Instanz aus der Bibliothek auf die Arbeitsfläche.
Wählt die Instanz an und öffnet das 'Aktionen' Fenster.
Wählt bei 'Eigenschaften' '_alpha'.
Im Skripteditor erscheint folgender Text:

```
onClipEvent (load) {
    _alpha;
}
```

Klickt auf _alpha und tragt unten _alpha=50 ein
Wenn ihr den Film mit TestMovie testet ist die Transparenz des Objektes 50%.

Instanzen vom Typ Filmsequenz werden durch die OnClipEvent()-Methoden gesteuert.
Der Bezeichner in den runden Klammern, z.B. (enterFrame) bestimmt, WANN die Anweisungen in den geschweiften Klammern, z.B. { _alpha+=10} ausgeführt werden.
Geschweifte Klammern dienen dazu, mehrere Aktionen zusammenzufassen.

onClipEvent(load){}
Die Anweisungen nach dieser Funktion werden nur ein einziges Mal zu Beginn der FlashAnimation ausgeführt. In dieser Funktion kann man z.B. Anfangswerte für eigene Variablen setzen.

onClipEvent(enterFrame){}
Die Anweisungen dieser Funktion werden bei jedem neuen Bildaufbau (normalerweise 12 mal pro Sekunde) ausgeführt. Auf diese Art kann man zum Beispiel mit _x+=5; Bewegungen erzeugen.

Actionscript Syntax

Kommentare - Zur besseren Verständlichkeit sollte man in die eigenen Skripte Kommentare einfügen. Diese beginnen immer mit //

Anweisungen werden immer durch Semikolon ; voneinander getrennt. Zur besseren Übersicht schreibt man eine Anweisung pro Zeile.

Blöcke - Mehrere Anweisungen lassen sich mit geschweiften Klammern zusammenfassen. Beispiel mit Kommentar, Block und Anweisungen:

```
// Kommentar
{
    Anweisung eins;
    Anweisung zwei;
}
```

Eigenschaften - Symbolinstanzen haben Eigenschaften, die man über die Scripte abrufen und verändern kann. Diese Eigenschaften beginnen immer mit einem Unterstrich und erscheinen im Skript in grüner Schrift.

_x :X-Koordinate
_xscale :Skalierung in x-Richtung
_rotation: Rotation in Grad
_alpha: Transparenz in Prozent
_xmouse: Mausabstand vom Nullpunkt der Instanz.

Variablen - Eine Variable ist ein Platzhalter für einen Wert. Variablen können Zahlen und Texte speichern.

`nummer = 262;`
`strasse = „heinrichstrasse“;`

Zuweisungen - Eigenschaften und Variablen kann man Werte zuweisen

```
variable = 10; // Variable erhält den Wert 10
variable += 10; // Variable wird um 10 erhöht
variable -= 10; // Variable wird um 10 vermindert
variable ++; // Variable wird 1 erhöht
```

Ausdrücke - Variablenwerte und Zahlen lassen sich beliebig miteinander verrechnen

Um Fehler zu vermeiden sollte man immer Klammern setzen.
`ziel = (start + (geschwindigkeit*100.0))/100.0;`

Abfragen - if (Bedingung) { Anweisungen; }

Die if() Anweisung funktioniert folgendermassen:

Wenn der Ausdruck in der runden Klammer wahr ist, werden die Anweisungen in der geschweiften Klammer ausgeführt:

```
if (_xmouse<0)
{
    _x-=1;
}
```

Wenn der Mauszeiger links vom Nullpunkt ist, bewegt sich auch das Objekt nach links

Bedingungen - Der Syntax für Bedingungen:

```
(_x < 10) // kleiner als
(_x > 10) // grösser als
(_x == 10) // ist gleich
(_x != 10) // ist ungleich
```

mehrere Bedingungen lassen sich mit „und“ und „oder“ zusammenfassen
&& - „und“ Verknüpfung;

|| - „oder“ Verknüpfung; (zwei senkrechte Striche)
! - „nicht“

Beispiele:

```
( (_x<10) && (_x>-10) ) // _x kleiner 10 UND grösser -10
( (_yscale==100) || (_xscale==100) ) // _yscale gleich 100 ODER _xscale gleich 100
```

Actionscript Funktionen/Klassen/Objekte

es gibt vordefinierte Funktionen, die man in seinen Skripten verwenden kann.

random()

Die Funktion random() erzeugt eine zufällige ganze Zahl. Der Wert in der Klammer gibt die Anzahl der Möglichkeiten.

random(8) erzeugt eine Zahl zwischen 0 und 7.

Math-Funktionen enthalten nützliche Funktionen, wie Wurzel, Sinus oder Absolutwert.

```
Math.PI // die Zahl Pi
Math.abs(-10) //der Absolutwert der Zahl (also ohne Vorzeichen)
Math.round(20.3) // die Zahl gerundet
Math.random() // Zufallswert zwischen 0.0 und 1.0
```

Color-Objekt

Um Farben zu verändern muss man in der onClipEvent(load) Funktion ein ColorObjekt für das Symbol erzeugen:

```
meineFarbe = new Color(this);
diesem kann man dann eine neuen RGB-Wert zuweisen:
meineFarbe.setRGB( rot*65536 + gruen*256 + blau);
```

RGB-Farbwerte

rot,gruen und blau müssen Ganzzahlen von 0 bis 255 sein.
(je höher der wert desto stärker ist der Farbanteil)
der RGB-Wert einer Farbe ist: rot*65536 + gruen*256 + blau

Zeichnen mit Actionscript (funktioniert seit Flash MX)

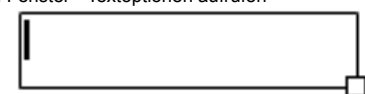
```
moveTo(x,y) // bewegt den Stift nach x,y
lineTo(x,y) // zeichnet eine Linie nach x,y
beginFill(rgb, alpha) // die folgenden Zeichenbefehle erzeugen ein Polygon
lineStyle(dicke, rgb, alpha) // Stichstärke festlegen
clear() // löscht alle gezeichneten Linien
```

dynamischer Text

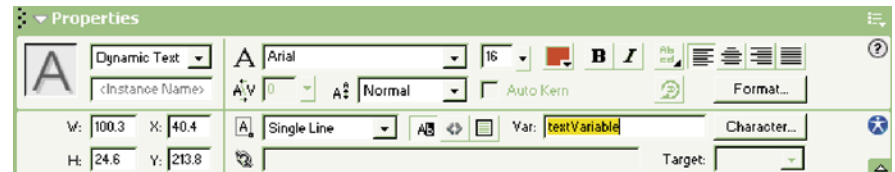
dynamische Texte kann man mit ActionScript modifizieren.

dynamischen Text erstellen:

1. ein Textfeld aufziehen (man braucht keinen Text einzugeben)
2. Fenster->Textoptionen aufrufen



3. Auf „Dynamischer Text“ einstellen.
4. Textvariable eingeben (keine Namen doppelt)



die Textvariable verhält sich ähnlich wie ein Instanzname

dynamischen Text verändern:

einfach im Skript dem Variablennamen (mit Hierarchie) einen Text in Anführungszeichen zuweisen.

Bsp.:

```
_root.TextField2 = „Hallo“;
```

man Texten auch Zahlenwert zuweisen.

mehrere Texte/Werte lassen sich mit „+“ zu einem LangenText verknüpfen:

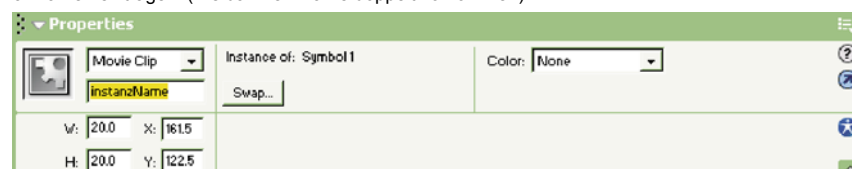
```
_root.TextField2 = „Meine X-Koordinate ist „+this._x“;
```

Symbolhierarchien

Es ist oft erforderlich, mit einem Skript verschiedenen Instanzen innerhalb des Flashfilmes anzusprechen.

Dazu muss man den Instanzen Namen geben:

1. Zu benennende Instanz auf der Arbeitsfläche anwählen.
2. Fenster -> Bedienfelder -> Instanz aufrufen
3. Namen eintragen. (Es darf kein name doppelt vorkommen)



Variablen und Eigenschaften dieser Instanz werden mit dem Punkt (.) angesprochen
`instanzName._x=10;`

Die Symbole in Flash sind hierarchisch organisiert.

Szene1 ist die **_root** Ebene.

Alle Instanzen in Szene1 werden folgendermassen Angesprochen:

```
_root.instanzName._x=10;
```

Befinden sich innerhalb der Instanzen weitere Instanzen, sieht es so aus:

```
_root.instanzName.innererInstanzname._x=10;
```