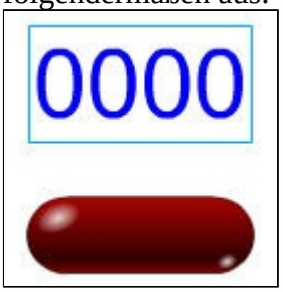
13. Zufall

Eines vorne weg. Aus Krankheitsgründen konnte ich an dieser Schulung nicht teilnehmen, wodurch die Dokumentation etwas knapper wie gewohnt ausfällt.

Zufallgenerator

- Wir basteln uns einen Zufallsgenerator, der auf Knopfdruck Ganzzahlen zwischen 0 und 10 generiert.
- Wir öffnen einen neuen Film und erstellen 2 Ebenen, die wir Grafik und Text nennen.
- In der Ebene Text erstellen wir einen dynamischen Text mit dem Variablen Inhalt Zahl

- In der Grafik Ebene platzieren wir den Standard Knopf Pill Button, die Animation sieht jetzt folgendermaßen aus:



- Beim Drücken des Knopfes wird nun über ein Actionscript dem dynamischen Feld Zahl die Zufallszahl zugewiesen:
  _root.Zahl = Math.random()*10;
- Wenn wir die Animation jetzt testen, werden Kommazahlen angezeigt. Deshalb muss die Zeile noch mit dem Befehl zum Runden erweitert werden:
  _root.Zahl = Math.round(Math.random()*10);
- Das Ergebnis sieht nun so aus: flash/13_01a.swf flash/13_01a.fla.

duplicateMovieClip

- Wir erstellen wieder einen neuen Film, mit 2 Ebenen, Grafik und Skript
- Wir fügen eine kleine Filmsequenz in der Ebene Grafik ein, als Vorlage kann aus der Bibliothek dieser FLA-Datei der Balkenfilm genutzt werden: flash/13_balken.fla
- Die Instanz der Filmsequenz nennen wir balken
- Nun erstellen wir eine Kopie der Instanz balken mit dem Befehl duplicateMovieClip und nennen diese Kopie balken1:
  balken.duplicateMovieClip("balken1", 1);
- Da die Kopie natürlich nicht an der gleichen Stelle erscheinen soll wie das Original, verändern wir die x- und y-Position mit Hilfe des Zufallgenerators:
  balken1._x = Math.random()*600;
  balken1._y = Math.random()*400;
- Damit dies auch anschaulich funktioniert benötigen wir nun zum Schluss noch einen Stop-Befehl:

```
stop();
```

- Fertig ist unsere erste Filmkopie: flash/13_01b.swf  flash/13_01b.fla  .

for-Schleife

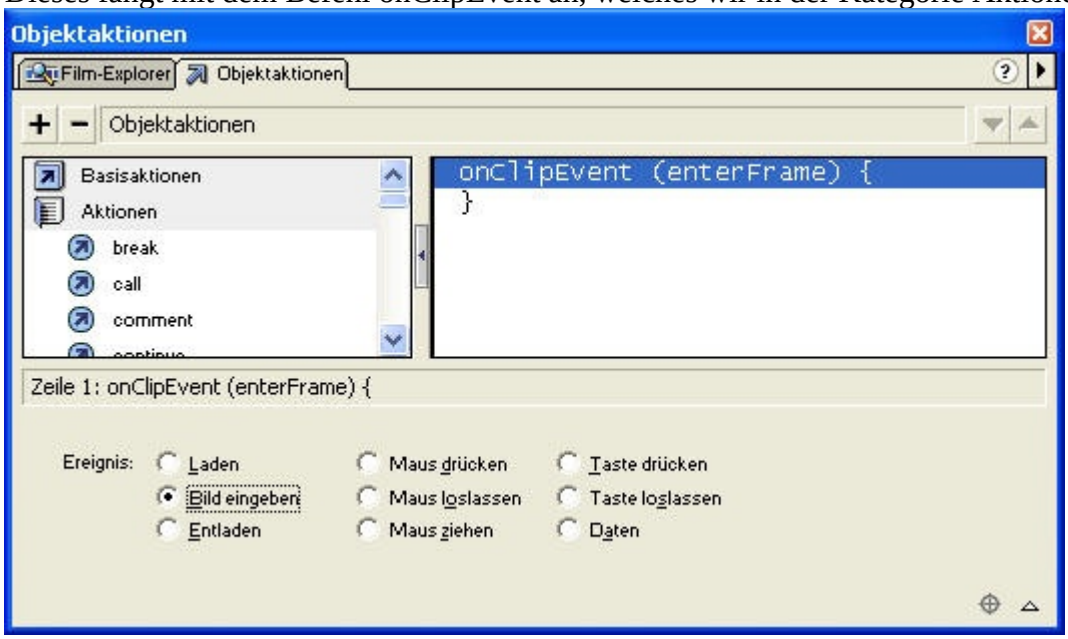
- Nun verteilen wir 20 Kopien der Instanz balken zufällig über die ganze Bühne.
- Dazu benötigen wir eine For Schleife:
  ```
  for (i = 1; i < 20; i++)
  {
  }
  ```
- Die For Schleife durchläuft den in den eckigen Klammern befindlichen Programmcode solange, bis die Variable i > 20 ist. Die Variable i startet mit dem Wert 1 und wird durch i++ immer um den Wert 1 erhöht.
- Nun übernehmen wir die 3 Programmzeilen aus dem vorherigen Beispiel.
- Da jetzt jede Instanz einen eigenen Namen benötigt, müssen wir den Namen aus dem Text balken und dem numerischen Wert der Variablen i zusammensetzen. Dazu benötigen wir einen weiteren Befehl: eval. Die 3 Zeilen aus dem vorherigen Beispiel ändern wir deshalb wie folgt ab:
  ```
  balken.duplicateMovieClip("balken" + i, i);
  eval ("balken" + i)._x = Math.random()*600;
  eval ("balken" + i)._y = Math.random()*400;
  ```
- Am Ende des Scripts benötigen wir wieder den Stop-Befehl.
- Wie wir sehen können funktioniert die for-Schleife schon: flash/13_01c.swf  flash/13_01c.fla  .
- Alle Filmkopien sehen aber noch gleich aus. Deshalb erweitern wir den Inhalt der Schleife um ein paar weitere Zufallsveränderungen an unseren Instanzkopien:
  ```
  eval ("balken" + i)._xscale = Math.random()*75+25;
  eval ("balken" + i)._yscale = eval("balken" + i)._xscale;
  eval ("balken" + i)._rotation = Math.random()*180;
  ```
- Außerdem wäre es schön eine Knopf zu haben, der die Schleife immer wieder von neuem startet, um mehrere Durchläufe mit der Zufallsverteilung besser sehen zu können. Dazu schieben wir zuerst den Inhalt der beiden Bilder 1 in Bild 2, so dass die Bilder 1 in beiden Ebenen leer sind.
- Jetzt fügen wir unseren beliebten Standardknopf Pill Button wieder in der Grafik-Ebene hinzu, links oben in der Ecke.
- Der Knopf bekommt ein einfaches Script mit Goto zum Bild 1:
  ```
  on (release) {
  gotoAndPlay (1);
  }
  ```
- Immer wenn jetzt der Knopf gedrückt wird, erfolgt die Verteilung von neuem: flash/13_01d.swf  flash/13_01d.fla  .

onClipEvent

- Wir basteln einen Ball, der sich selbstständig auf der Bühne bewegt und von den Rändern abprallt.
- Dazu starten wir mit einer neuen Flashdatei mit der Größe 550 x 400 Pixel und erstellen die 2 Ebenen Skript und Grafik.
- Dann erstellen wir eine Filmsequenz mit einer einfachen Kugel.
- Eine Instanz der Filmsequenz wir mittig auf der Bühne platziert und
- Nun geben wir im 1. Bild der Ebene Skript folgende Befehle ein:

step = 5;
stop ();

- Nun fügen wir mit einem Rechtsklick auf die Instanz und der Auswahl Aktion ein ActionScript zu. Dieses fängt mit dem Befehl onClipEvent an, welches wir in der Kategorie Aktionen finden:



Das Ereignis wird auf Bild eingeben geändert, was so viel bedeutet wie: immer wenn die Instanz auf der Bühne neu angezeigt wird.

- Nun ergänzen wir das Skript um den noch fehlenden Inhalt:

```
onClipEvent (enterFrame)
{
this._x += _root.step;
if (this._x >= 550)
{
_root.step = -_root.step;
}
}
```

Hintergrund zur Kurzschreibweise von Zuweisungen

x = x + 1 entspricht x ++
x = x + 4 entspricht x += 4

x = x * -1 entspricht x = -x
Dadurch wird im obigen Beispiel der Wert für _root.step zur Berechnung immer zwischen + und - gewechselt!!!

- Die Kugel wandert nun bis an den rechten Rand und ändert dann seine Richtung nach links

flash/14_01a.swf flash/14_01a.fla.

- Im nächsten Schritt wollen wir die Kugel auch noch dazu bringen am linken Rand wieder abzuprallen. Dazu erweitern wir die If-Abfrage um eine ODER Verknüpfung:
if ((this._x >= 550) || (this._x <= 0))

Hintergrund zur Verknüpfung von Abfragen

|| ODER-Verknüpfung
Beispiel: if x = 1 || y = 1

&& UND-Verknüpfung

Beispiel: if x = 1 && y = 1

- Jetzt wandert die Kugel schon schön zwischen links und rechts hin und her flash/14_01b.swf  flash/14_01b.fla  .
- Nun erweitern wir die Bewegung der Kugel nach oben und untern. Dazu benötigen wir nun zwei 2 step Variablen. Eine für den X-Wert und eine für den Y-Wert. Deshalb ändern wir das Initialisierungsskript im Bild eins auf:
  stepx = 5;
  stepy = 5;
  stop();
- Außerdem muss natürlich auch das Aktionscirpt der Instanz angepasst werden:

```
onClipEvent (enterFrame)
{
this._x += _root.stepx
this._y += _root.stepy

// x-Bereich Abfrage
if ((this._x >= 550) || (this._x <= 0))
{
_root.stepx = -_root.stepx;
}

// y-Bereich Abfrage
if ((this._y >= 400) || (this._y <= 0))
{
_root.stepy = -_root.stepy;
}
}
```

- Danach bewegt sich die Kugel in alle Richtungen flash/14_01c.swf  flash/14_01c.fla  .
- Der nächste Schritt wird wieder etwas komplizierter. Zur Zeit ändert die Kugel ihre Richtung erst, wenn die Mitte der Kugel den Rand erreicht hat. Deshalb müssen wir die Abfragewerte für den Richtungswechsel noch anpassen.
- Im ActionScript von Bild 1 ermitteln wir zuerst die Werte die wir für diesen Schritt benötigen. Es handelt sich um die Größe der Kugel geteilt durch 2. Das Script dazu lautet:

```
// maximale Breite ermitteln
maxx = getProperty ( _root.Kugel, _width ) / 2;
// maximale Höhe ermitteln
maxy = getProperty ( _root.Kugel, _height ) / 2;
```

- Nun müssen im Script zur Kugel die maxx-Werte bei 0 addiert und bei 550 abgezogen werden. Dann wird die Kugel nicht erst beim erreichen des Mittelpunktes Ihre Richtung wechseln, sondern schon bei der Berührung der Wand. Das Gleiche Spiel natürlich auch für maxy. Die beiden if-Abfragen dazu lauten:

```
if ((this._x >= 550 - _root.maxx) || (this._x <= 0 + _root.maxx))
```

  und

```
if ((this._y >= 400 - _root.maxy) || (this._y <= 0 + _root.maxy))
```

- Nun bewegt sich die Kugel so, wie wir sie haben wollen flash/14_01d.swf  flash/14_01d.fla  .