





# 16. Interaktion



## Tastatursteuerung

- Wir erstellen wieder einen neuen Flashfilm. Dieser enthält die 2 Ebenen Skript und Grafik.
- In der Ebene Grafik wird wieder mittig eine Animation hinterlegt. Diese erstellen wir aus einem Kreis, der sich von 100% auf 150% vergrößert und dann wieder auf 100% verkleinert.
- Diese Instanz nennen wir Kreis
- In der Ebene Grafik geht diese Kreisanimation über die Bilder 1 + 2
- In der Skript Ebene tragen wir in Bild 2 den Befehl gotoAndPlay (1); ein.
- Im Bild 1 der Skript Ebene wird folgendes Script eingetragen:  

```
xPos = getProperty(_root.Kreis, _x);
yPos = getProperty(_root.Kreis, _y);
Step = 4;
if (Key.isDown(Key.RIGHT))
setProperty (_root.Kugel, _x, xPos + Step);
```
- Der Befehl **Key.isDown(Key.RIGHT)** ist unter Objekte / Key zu finden und prüft, ob die Taste Cursor nach rechts gedrückt ist.

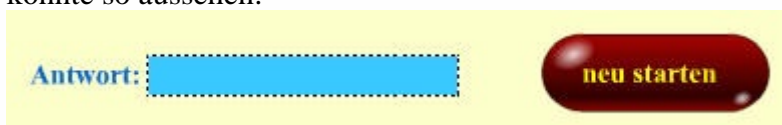
- Hier kurz mal zum Test [flash/15\\_02a.swf](#)  [flash/15\\_02a fla](#) .
- Nun ergänzen wir das Skript um die restlichen 3 Cursor Tasten.

- Und schon gehts in alle 4 Richtungen [flash/15\\_02b.swf](#)  [flash/15\\_02b fla](#) .
- Einen besonderen Effekt erzielen wir jetzt noch, wenn wir beim Drücken einer der 4 Cursor Tasten die Instanz auf der Bühne leicht versetzt kopieren. Dazu nutzen wir den duplicateMovie Befehl aus Kapitel 13 und für die Abfrage die UND-Verknüpfung von Kapitel 14.
- Am Ergebnis arbeite ich noch - Fehler gefunden, ich hatte statt dem Instanznamen Kreis den Namen

Kugel im Script an einigen Stellen verwendet: [flash/15\\_02c.swf](#)  [flash/15\\_02c fla](#) .

## Formulare



- Wir beginnen mit einem neuen Film. Die vorhandene Ebene wird als Text bezeichnet. Die Mitte der Bühne wird für die Antwort freigehalten. Im oberen Bereich wird eine Frage mit 3 Antworten vorgegeben. z.B.  
Wer baute den ersten Computer?  
- Bill Gates  
- Konrad Zuse  
- Steve Jobs
- Im unteren Bereich wird ein Antwortfeld erstellt und ein Knopf um die Animation neu zu starten. Dies könnte so aussehen:

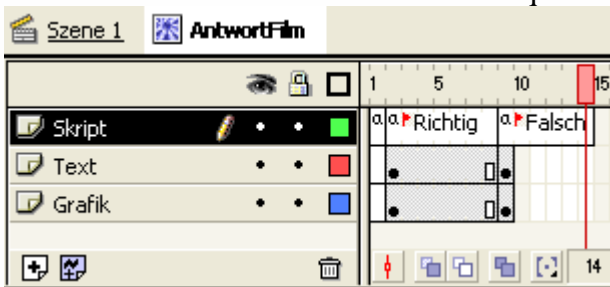


- Beim Antwortfeld handelt es sich um ein Textfeld Typ Texteingabe und wird als Variable Antwort festgelegt. Im obigen Beispiel reichen 11 Buchstaben man kann aber auch die standardmäßige 0 bei Max. Zch. stehen lassen.



- Normalerweise ist das Eingabefeld nicht sichtbar. Daher wurde es mit einem farbigen Rechteck hinterlegt in der identischen Größe der Texteingabe.

- Damit haben wir erst mal das Grundgerüst gebastelt [flash/16\\_01.swf](#)  [flash/16\\_01 fla](#) .
- Als nächstes entwerfen wir eine Filmanimation für die Antwort. Diese Filmsequenz nennen wir AntwortFilm und erstellen in der Filmsequenz 3 Ebenen wie im Beispiel:



- In der Ebene Skript wird in 1,2 und 9 jeweils nur der Stop(); Befehl eingefügt und zur besseren Übersicht wird Bild 2 als Sprungmarke Richtig und Bild 9 als Sprungmarke Falsch angelegt. Zur Erinnerung, dazu wird einfach das Bild markiert und im Bedienfeld Bild ein Name vergeben.



- Für die Anzeige erstellen wir eine grüne und eine rote Kugel jeweils mit der Beschriftung Richtig und Falsch. Die Grafik liegt auf der Grafik Ebene, die Beschriftung in der Ebene Text. Hier als Beispiel die Anzeige für Falsch:



- Die Filmsequenz wird mittig auf der Bühne platziert und die Instanz als AntwortFilm bezeichnet.
- Nun folgt als letzter Punkt die Skriptprogrammierung für die Auswertung. Dazu fügen wir im Hauptfilm eine zweite Ebenen hinzu, die wir Skript nennen.
- Im Bild 1 erfolgt die Initialisierung. Damit das Antwort Feld leer ist wird es entsprechend mit "" vorbelegt. Außerdem richten wir die Variable RichtigeAntwort ein, die als Wert auch die Richtige Antwort in Großbuchstaben enthält  

```
_root.Antwort = "";
RichtigeAntwort = "KONRAD ZUSE";
```
- Im Bild 3 wird für eine Endlosschleife einfach nur zum Bild 2 zurück gesprungen.  

```
gotoAndPlay(2);
```

- Die eigentliche Auswertung erfolgt nun im Bild 3 der Ebene Skript. Dazu benötigen wir die Befehle:

```
if
else
tellTarget
gotoAndPlay
```



- Wenn mit den obigen Befehlen eine Lösung gebastelt wurde könnte diese wie folgt aussehen:

```
if (Antwort != "")
{
  if (RichtigeAntwort == Antwort)
  {
    tellTarget ("AntwortFilm")
    {
      gotoAndPlay ("Richtig");
    }
  }
  else
  {
    tellTarget ("AntwortFilm")
    {
      gotoAndPlay ("Falsch");
    }
  }
}
```

- Theoretisch könnte man auch mit dem gotoAndStop Befehl arbeiten und sich dadurch in der Filmsequenz AntwortFilm die Stop() Befehle sparen. Die Praxis zeigt aber, dass die obige Programmierung besser funktioniert. Manchmal wird von Flash 5 beim Befehl gotoAndStop erst im Feld nach der Sprungadresse gestoppt.

### toUpperCase()

- Leider funktioniert die richtige Antwort nur, wenn diese in Großbuchstaben eingegeben wird. Um einfach jede Antwort in Großbuchstaben umzuwandeln gibt es den Befehl toUpperCase(). Dieser wird einfach an das Objekt, in diesem Fall eine Variable, mit einem Punkt angehängt:  
if (RichtigeAntwort == Antwort.toUpperCase())

- Jetzt funktioniert auch die Auswertung [flash/16\\_01a.swf](#)  [flash/16\\_01a fla](#) .
- Was jetzt noch fehlt ist die Funktion für den Knopf. Durch den Knopf soll der Inhalt des Antwortfeldes und die Kugel auf der Bühne gelöscht werden. Eine Lösung dafür könnte so aussehen:

```
on (press)
{
  tellTarget ("AntwortFilm")
  {
    gotoAndPlay (1);
  }
  gotoAndPlay (1);
}
```

- Jetzt funktioniert auch die der Knopf [flash/16\\_01c.swf](#)  [flash/16\\_01c fla](#) .

### Textdatei auswerten

- Wir beginnen wieder mit einer neuen Flashdatei. Diese bekommt 2 Ebenen, die Ebene Skript und die Ebene Text.
- In der Ebene Skript wird bei Bild 1 ein Stop(); Befehl eingebaut.
- In der Ebene Text werden fangen wir mit einem Textfeld an, welches wir wie folgt formatieren:





- Als zweites Textfeld erstellen wir dieses:



- Jetzt benötigen wir noch einen Knopf. Am besten einfach den gewohnten Pill Button, den wir mit dem Text "laden" beschriften.
- In dem Verzeichnis, in dem die swf-Datei veröffentlicht wird, erstellen wir eine Text Datei, die wir einfach "text16.txt" nennen. Diese Datei enthält folgenden Fließtext:  
&text=Ein Frohes neues Jahr und einen Guten Rutsch!Ein Frohes neues Jahr und einen Guten Rutsch!Ein Frohes neues Jahr und einen Guten Rutsch!Ein Frohes neues Jahr und einen Guten Rutsch!Ein  
&datum=13.12.2002
- &text und &datum sind Variablen, die über das Einlesen der Datei mit den oben definierten Textfeldern verknüpft wird.
- Dazu benötigen wir noch folgendes Script auf dem Knopf-lesen:  

```
on (press) {
loadVariables("text16.txt", _root);
}
```



- Hier die dazugehörigen Dateien: [flash/16\\_04.swf](#)  [flash/16\\_04 fla](#)  [flash/text16.txt](#).
- Aber jetzt kommt erst das Interessante an dieser Geschichte. Ändern Sie während die SWF-Datei geöffnet ist den Text in der Datei Text16.txt und klicken Sie nochmals den laden-Knopf. Was passiert? Der gerade geänderte neue Text erscheint in der Flash-Animation. Damit ist es möglich ohne Änderung Flashanimation Variablenwerte zu ändern.