

Hausaufgabe 3 – Animation und Interaktion mit Flash

Abgabe

- Abgabetermin: **29. Juli 2003**
- Abzugeben sind sowohl die Projektdatei aus dem Autorenwerkzeug (Dateiname: *vorname_nachname fla*), wie auch die exportierte Datei (Dateiname: *vorname_nachname swf*)

Aufgabe

Aufgabenstellung:

Zeichnen und Programmieren eines kleinen Spiels. Bob der Bauarbeiter muss Münzen in einem Labyrinth einsammeln. Dabei kann er sich nur in den Gängen bewegen. Wenn er alle fünf Münzen gefunden hat, ist das Spiel beendet.

Benötigte Funktionen:

- Zeichnen
- Bewegungs- und Form-Tweening
- Konzept der Objekte und Instanzen
- Abfangen von Userevents
- Bewegung von Objekten per Skript
- Kollisionskontrolle
- einfache Programmierkonstrukte (Bedingungen, Schleifen...)

Arbeitsschritte:

1. Einrichten der Bühne:

300px x 300px, Framerate 25, Hintergrundfarbe

2. Zeichnen des Labyrinths:

- Zeichnen eines Backsteins (25px x 20px), durch hellere und dunklere Stellen "3D-Effekt" erzeugen. Umwandeln in Movieclip (F8) "brick".
- Den Movieclip auf der Bühne so oft duplizieren und verschieben, bis aus den Backsteinen ein Labyrinth gebaut werden kann. Entwerfen eines eigenen Labyrinths.
- Alle Steine markieren, in Movieclip "maze" umwandeln.
- in der Hauptszene einen Boden für das Labyrinth malen

3. Figur zeichnen und animieren:

Bob besteht aus mehreren Einzelteilen, die alle einzeln gemalt und Movieclips gewandelt werden müssen. Anschließend können die Teile zu der Gesamtfigur (Movieclip "bob") zusammengebaut werden, wobei jedes Element in einem eigenen Layer/Ebene liegen muss. Wichtig ist, dass die gesamte Figur schmal genug sein muss, um in den Gängen laufen zu können.

Der Aufbau:

bob

-> kopf

- > körper
- > fuß1
- > fuß2

Die Teile können jetzt mit Bewegungs-Tweenings (Motion-Tween) animiert werden. Dazu im MC "bob" im 2. Frame der Zeitleiste Schlüsselbilder/Keyframes erzeugen (mit F6). In diesen Frames müssen jetzt die Körperteile in ihre Startposition gebracht werden:

- Körper um 10 Grad rotieren lassen
- einen Fuß etwas nach vorne, den anderen nach hinten

Nach den Änderungen auch im 9. und 17. Frame Schlüsselbilder erzeugen. Im 9. Frame werden die Teile in die Endposition gebracht:

- Körper um -10 Grad drehen
- vorne/hinten der Füße tauschen
- Kopf in vorne-hinten-Richtung etwas stauchen und nach vorne schieben

Den 1. Frame in jeder Ebene auswählen und "Bewegungstween" wählen, ebenso im 9. Frame. Dann den 16. Frame in Keyframes umwandeln und den 17. Frame löschen (so wird die Bewegung flüssiger beim Zurückspringen).

Zum Schluss muss noch eine Aktion hinzugefügt werden, die den MC im 1. Frame anhält, da die Figur sonst sofort losläuft.

4. Münzen

Die Münze besteht aus einem einfachen Kreis, der so animiert wird, dass die Münze sich scheinbar im Raum dreht. Dazu die Münze in einen MC "coin" umwandeln, im 10. und 20. Frame ein Schlüsselbild anlegen und zwischen den Keyframes Form-Tweenings erzeugen. Im 10. Frame kann jetzt die Form (-> Transformieren-Palette) und evtl. die Farbe der Münze verändert werden. Wie oben den vorletzten Frame in ein Schlüsselbild umwandeln und den letzten löschen.

5. Finish

Jetzt kann das Spielfeld komplett zusammengebaut werden; dazu Bob und fünf Münzen in den Gängen platzieren (dabei sollte Bob möglichst genau in einer Kachel des Labyrinth-Gitters stehen). Um über Actionscript angesprochen werden zu können, müssen die einzelnen Elemente auf der Bühne benannt werden (Instanznamen):

Bob -> "bob"

Labyrinth -> "maze"

Münzen -> "c1" bis "c5"

Zusätzlich muss noch ein dynamisches (-> Eigenschaftspalette) Textfeld erstellt werden, dem in der Eigenschafts-Palette der Variablenname (Var:) "coins" zugewiesen wird.

6. Laufsteuerung

Die Figur soll mit den Pfeiltasten gesteuert werden können. Flash bietet dazu Listener an, die u.a. Benutzereingaben (hier: das Ereignis Tastendruck/onKeyDown) abfangen und entsprechende Methoden aufrufen. Das folgende Codestück erzeugt einen solchen Listener und bindet ihn an das Key-Objekt.

```
myListener = new Object();  
myListener.onKeyDown = function() {  
    if (Key.isDown(Key.DOWN)) {
```

```
        //Methodenaufruf: nach unten bewegen
    } else if (Key.isDown(Key.UP)) {
        //Methodenaufruf: nach oben bewegen
    } else if (Key.isDown(Key.LEFT)) {
        //Methodenaufruf: nach links bewegen
    } else if (Key.isDown(Key.RIGHT)) {
        //Methodenaufruf: nach rechts bewegen
    }
};
Key.addListener(myListener);
```

Die Laufmethode(n) bewegt Bob jetzt jeweils um eine Kachelbreite nach links/rechts oder um eine Kachelhöhe nach oben/unten. Außerdem soll sich Bob auch immer in die Laufrichtung drehen und die Animation abspielen. Zusätzlich muss geprüft werden, ob er eine Münze einsammelt. Da Bob nicht durch Wände laufen kann, darf all dies nur dann durchgeführt werden, wenn er nach dem Verschieben nicht den "maze"-MC berührt (-> 7.). Die Methode hat also folgenden Aufbau:

```
if (<no collision>) {
    //verschieben
    //drehen
    //animation abspielen
    //münze gesammelt? [collectCoins()]
}
```

Zum Ändern der Position/Rotation eines MCs muss die entsprechende MC-Eigenschaft verändert werden. Der Zugriff erfolgt über <Instanzname>.<Eigenschaft>. Eine Beschreibung aller Eigenschaften gibt es in der Referenz-Palette unter dem Punkt "Eigenschaften".

7. Kollisionserkennung

Eine einfache Möglichkeit zur Erkennung von Kollisionen zwischen zwei Objekten bietet die mc.hitTest-Methode (Referenz: Objekte->Film->Movieclip->Methoden). Um Kollisionen mit komplexen Formen wie dem Labyrinth zu erkennen, muss die Variante mc.hitTest(x, y, shapeFlag) verwendet werden, z.B.:

```
maze.hitTest(<x-Pos von Bob nach Laufen>, <y-Pos von Bob nach Laufen>, true)
```

Für einfache Objekte wie die Münzen genügt die zweite Variante mc.hitTest(target), z.B.:
c1.hitTest(bob)

8. Münzen sammeln

Beim Aufsammeln einer Münze soll die Münze vom Spielfeld verschwinden und in den Bereich links neben dem Textfeld verschoben werden. Außerdem erhöht sich die Variable coins um 1. Wenn coins den Wert 5 erreicht, wird die Variable auf "fertig" gesetzt. Der Aufbau der Funktion ist also wie folgt:

```
function collectCoins() {
    //wiederhole für die Münzen ci (i = 1..5)
    //Kollision ci mit bob? {
        //setze y-Pos von ci = 12;
        //setze x-Pos von ci = 225 - coins*12;
        //erhöhe coins um 1
    }
```

```
        }  
    }  
    //alle Münzen gesammelt? -> coins = "fertig"  
}
```

Hinweis: Der Zugriff auf den Movieclip ci (i = 1..5) erfolgt über `_root["c"+i]`.

Optional (d. h. nicht für die Hausaufgabe bewertet):

- Gegner: laufen auch nur in Gängen, bei Kollision Spielende o.ä.
- Sound: z.B. beim Einsammeln einer Münze, Ende des Spiels, Kollision
- Timer
- mehr Animationen, z.B. bei Kollision, Münze sammeln, Spielende