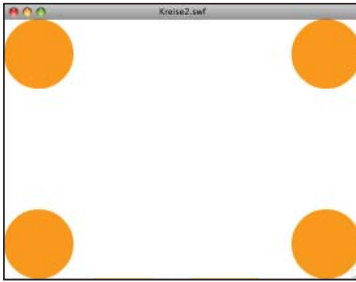


KREISE POSITIONIEREN

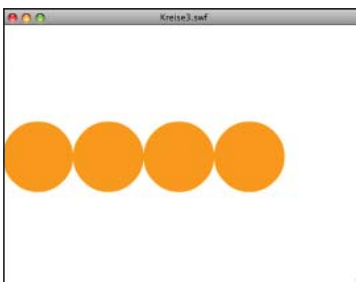
Die vier Kreise sind Instanzen eines Kreises in der Bibliothek und haben die Instanznamen kreis1, kreis2, kreis3 und kreis 4

```
kreis1.x = 0;  
kreis1.y = 0;  
kreis2.x = stage.stageWidth;  
kreis2.y = 0;  
kreis3.x = 0;  
kreis3.y = stage.stageHeight;  
kreis4.x = stage.stageWidth;  
kreis4.y = stage.stageHeight;
```



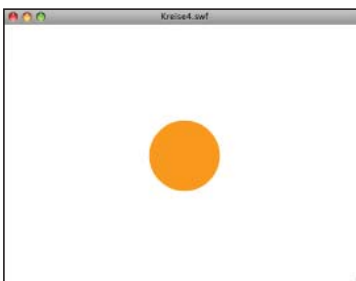
KREISE POSITIONIEREN

```
kreis1.x = 0+kreis1.width/2;  
kreis1.y = 0+kreis1.height/2;  
kreis2.x = stage.stageWidth-kreis2.width/2;  
kreis2.y = 0+kreis2.height/2;  
kreis3.x = 0+kreis3.width/2;  
kreis3.y = stage.stageHeight-kreis3.height/2;  
kreis4.x = stage.stageWidth-kreis4.width/2;  
kreis4.y = stage.stageHeight-kreis4.height/2;
```



KREISKETTE POSITIONIEREN

```
kreis1.x = kreis1.width/2;  
kreis2.x = kreis1.width+kreis2.width/2;  
kreis3.x = kreis1.width+kreis2.width+kreis3.width/2;  
kreis4.x = kreis1.width+kreis2.width+kreis3.width+kreis4.width/2;  
kreis1.y = kreis2.y = kreis3.y = kreis4.y = stage.stageHeight/2;
```



KREIS MIT NEW ERZEUGEN

In der Bibliothek liegt ein Movieclip mit dem Namen 'Kreis',
Bei den eigenschaften muss 'Export für ActionScript' aktiviert sein.

```
var kreis_mc:MovieClip;  
kreis_mc = new Kreis();  
this.addChild(kreis_mc);
```

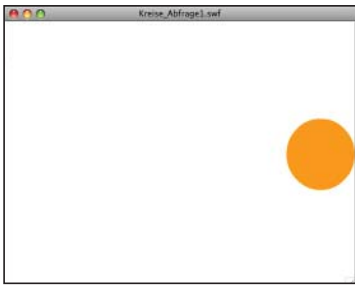
KREIS ZENTRIERT POSITIONIEREN

VAR1

```
this.getChildAt(0).x = stage.stageWidth/2;  
this.getChildAt(0).y = stage.stageHeight/2;
```

VAR2

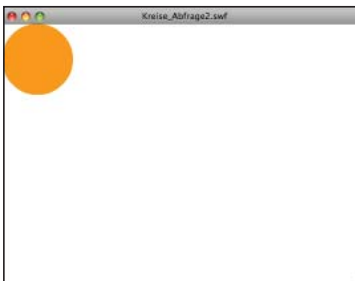
```
kreis_mc.x = stage.stageWidth/2;  
kreis_mc.y = stage.stageHeight/2;
```



KREISPOSITION ABFRAGEN 1

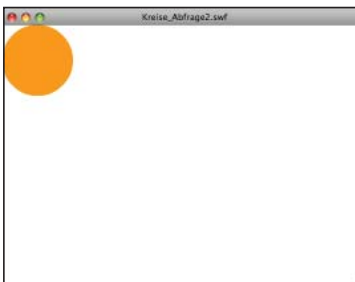
Ist der Kreismittelpunkt in der linken Hälfte, dann soll er am linken Rand platziert werden, sonst am rechten.

```
if (kreis_mc.x < stage.stageWidth/2) {  
    kreis_mc.x = kreis_mc.width/2;  
    kreis_mc.y = stage.stageHeight/2;  
} else {  
    kreis_mc.x = stage.stageWidth-kreis_mc.width/2;  
    kreis_mc.y = stage.stageHeight/2;  
}
```



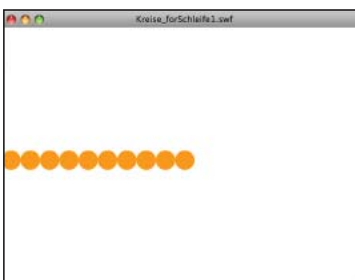
KREISPOSITION ABFRAGEN 2

```
if (kreis_mc.x <= stage.stageWidth/2 && kreis_mc.y <= stage.stageHeight/2) {  
    kreis_mc.x = kreis_mc.width/2;  
    kreis_mc.y = kreis_mc.height/2;  
}
```



KREISPOSITION ABFRAGEN 3

```
if (kreis_mc.x < stage.stageWidth/2) {  
    kreis_mc.x = kreis_mc.width/2;  
} else if (kreis_mc.x > stage.stageWidth/2) {  
    kreis_mc.x = stage.stageWidth - kreis_mc.width/2;  
}  
kreis_mc.y = stage.stageHeight/2;
```



KREISKETTE MIT FOR-SCHLEIFE

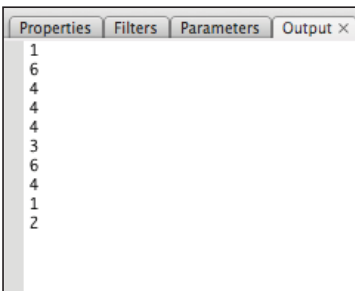
```
for (var i=1; i<=10; i=i+1) {  
    var kreis_mc:MovieClip = new Kreis();  
    this.addChild(kreis_mc);  
    kreis_mc.x = kreis_mc.width/2+(i-1)*kreis_mc.width;  
    kreis_mc.y = stage.stageHeight/2;  
}
```



KREISKETTE MIT FOR-SCHLEIFE

```
for(var i=0; i<8; i=i+1)
{
  var kreis: MovieClip = new Kreis();
  this.addChild(kreis);

  kreis.x = stage.stageWidth/2;
  kreis.y = stage.stageHeight - (kreis.height/2) - i*kreis.height;
}
```

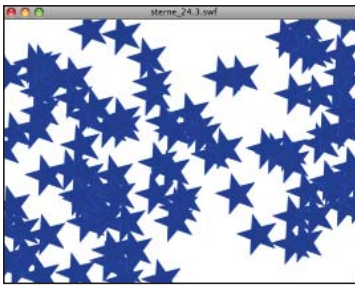


TRACE MATH.RANDOM & MATH.FLOOR

```
//trace Funktion um Ausgabewerte zu erhalten von der angegebenen random-Funk-
tion die Zufallszahlen liefert
//trace(Math.random()*6);

//floor Funktion liefert nur ganze Zahlen
//trace(Math.floor(Math.random()*6));

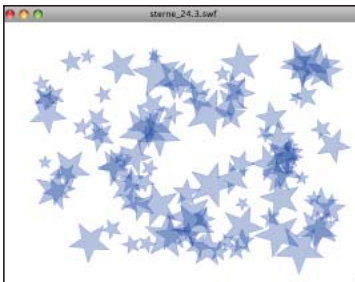
//Würfelbeispiel: Man würfelt und kann nur Zahlen zwischen 1 und 6 erhalten;
//mit +1 verschiebt man das Intervall um 1, 0 darf ja in diesem Fall nicht vorkommen;
//Immer mit Math.floor arbeiten!!
trace(Math.floor(Math.random()*6)+1);
```



MATH.RANDOM STERNFUNKTION

In der Bibliothek liegt ein Movieclip 'Stern'. 'Export für ActionScript' ist aktiviert.

```
var stern_mc:MovieClip;
for (var i=0;i<100;i++) {
    stern_mc = new Stern();
    this.addChild(stern_mc);
    stern_mc.x = Math.floor(stage.stageWidth * Math.random());
    stern_mc.y = Math.floor(stage.stageHeight * Math.random());
}
```



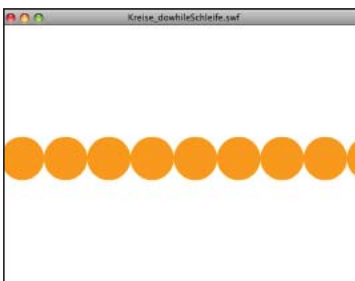
MATH.RANDOM STERNFUNKTION

*/ In einem BEGRENZUNGSRAHMEN ab 50px
Maximumwert-Minimumwert + 1 = Intervallgröße
+ den jeweiligen Startwert, um den das Intervall verschoben werden soll. */

```
var stern_mc:MovieClip;
for (var i=0;i<100;i++) {
    stern_mc = new Stern();
    this.addChild(stern_mc);
    stern_mc.x = Math.floor(Math.random() * (500-50+1))+50;
    stern_mc.y = Math.floor(Math.random() * (350-50+1))+50;
    //Deckkraft zwischen 30% und 50%, ab 30%
    stern_mc.alpha = Math.floor(Math.random() * (0.5-0.3))+0.3;
    //Skalierung zwischen 120 und 30 Prozent; Startwert:ab 30%
    stern_mc.scaleX = stern_mc.scaleY = (Math.random() * (1.2-0.3))+0.3;
    //Drehung, Startwert: -180
    stern_mc.rotation = -180 + Math.random() * 360; //180- (-180) =360
}
```

// allgemeiner formuliert:

```
stern_mc.x = Math.floor(Math.random() * (stage.stageWidth -100+1))+50;
stern_mc.y = Math.floor(Math.random() * (stage.stageHeight-100+1))+50;
```

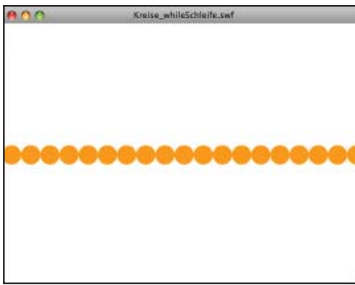


KREISKETTE MIT DO-WHILE-SCHLEIFE

*/ Lösung mit do-while-Schleife.

Das ist der beste Weg, da auf jeden Fall ein Kreis erstellt werden muss./*

```
var i:int = 0;
do {
    var kreis_mc:MovieClip = new Kreis();
    this.addChild(kreis_mc);
    kreis_mc.y = stage.stageHeight/2;
    kreis_mc.x = kreis_mc.width/2 + i*kreis_mc.width;
    i++;
} while (kreis_mc.x + kreis_mc.width/2 < stage.stageWidth);
```



KREISKETTE MIT WHILE-SCHLEIFE

```

var i:int = 0;
// Erstellen des Startkreises am linken Bühnenrand auf halber Bühnenhöhe.
var kreis_mc:MovieClip = new Kreis();
this.addChild(kreis_mc);
kreis_mc.y = stage.stageHeight/2;
kreis_mc.x = kreis_mc.width/2 + i*kreis_mc.width;
i++;

```

// Überprüfen, ob der rechte Rand des letzten Kreises noch innerhalb der Bühne liegt.

```

while (kreis_mc.x + kreis_mc.width/2 < stage.stageWidth) {
    // Erstellen des nächsten Kreises.
    kreis_mc = new Kreis();
    this.addChild(kreis_mc);
    kreis_mc.y = stage.stageHeight/2;
    kreis_mc.x = kreis_mc.width/2 + i*kreis_mc.width;
    i++;
}

```

// VARIATION 1:

```

var i:int = 0;
var distanceX:int = 0;
// Überprüfen, ob der rechte Rand vom letzten Kreis noch innerhalb der Bühne liegt.
// Wenn ja, neuen Kreis erstellen.
// Anmerkung: Es wird davon ausgegangen, dass die Kreisbreite kleiner als die
// Bühnenbreite ist.
while (distanceX < stage.stageWidth) {
    var kreis_mc:MovieClip = new Kreis();
    this.addChild(kreis_mc);
    kreis_mc.y = stage.stageHeight/2;
    kreis_mc.x = kreis_mc.width/2 + i*kreis_mc.width;
    // distanceX speichert die Position vom rechten Rand des letzten Kreises.
    distanceX = kreis_mc.x + kreis_mc.width/2;
    i++;
}

```

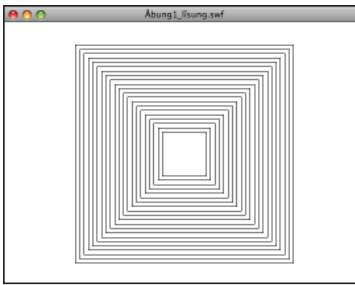
// VARIATION 2:

Elegante Lösung, die aber nur funktioniert, wenn sonst nichts auf der Bühne liegt.

```

var i:int = 0;
// stage.width liefert die totale Breite von den Objekten auf der Bühne.
while (stage.width < stage.stageWidth) {
    var kreis_mc:MovieClip = new Kreis();
    this.addChild(kreis_mc);
    kreis_mc.y = stage.stageHeight/2;
    kreis_mc.x = kreis_mc.width/2 + i*kreis_mc.width;
    i++;
}

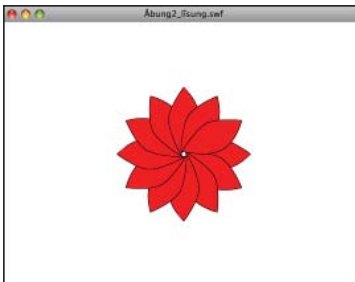
```



GENERATIVE GRAFIK

'Quadrat' ist ein Movieclip in der Bibliothek, 'Export for ActionScript' ist aktiviert.

```
var quad:MovieClip;
for (var i=0; i<=20; i++) {
    quad = new Quadrat();
    this.addChild(quad);
    quad.x = stage.stageWidth/2;
    quad.y = stage.stageHeight/2;
    quad.scaleX = quad.scaleY = 1 + i*0.2;
}
```



GENERATIVE GRAFIK

'Blatt' ist ein Movieclip in der Bibliothek, 'Export for ActionScript' ist aktiviert.
Der Registrierungspunkt ist an einer der beiden Spitzen.

A) MIT WHILE- SCHLEIFE

```
var blatt:MovieClip;
var winkel:Number = 90;
while (winkel < 90 + 360) {
    blatt = new Blatt();
    this.addChild(blatt);
    blatt.x = stage.stageWidth/2;
    blatt.y = stage.stageHeight/2;
    blatt.rotation = winkel;
    winkel = winkel + 30;
}
```

B) MIT FOR- SCHLEIFE

```
var blatt:MovieClip;
for (var i=0; i<12; i++) {
    blatt = new Blatt();
    this.addChild(blatt);
    blatt.x = stage.stageWidth/2;
    blatt.y = stage.stageHeight/2;
    blatt.rotation = 90 + i*30;
}
```



ZUFALLSZAHLEN MIT SWITCH ANWEISUNG

Zufällig wird immer eines der vier farbigen Quadrate aus der Bibliothek ausgewählt.

```
var square: MovieClip;
var select: int;
for (var i=0; i<50; i=i+1) {
    select = Math.floor( Math.random()*4);
    switch (select) {
        case 0 :
            square = new SquareBlue();
            break;
        case 1 :
            square = new SquareGreen();
            break;
        case 2 :
            square = new SquareRed();
            break;
        case 3 :
            square = new SquareYellow();
            break;
    }
    addChild(square);
    square.x = Math.random() * stage.stageWidth;
    square.y = Math.random() * stage.stageHeight;
    square.alpha = 0.2 + Math.random()*0.5;
    square.rotation = 45;
}
```

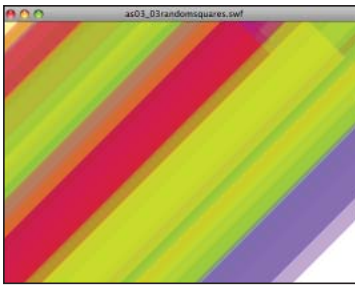
VARIANTE MIT IF - ELSE IF

```
var square: MovieClip;
var select: int;

for (var i=0; i<50; i=i+1) {

    select = Math.floor( Math.random()*4);
    if (select==0) {
        square = new SquareBlue();
    } else if (select==1) {
        square = new SquareGreen();
    } else if (select==2) {
        square = new SquareRed();
    } else {
        square = new SquareYellow();
    }
    this.addChild(square);

    square.x = Math.random() * stage.stageWidth;
    square.y = Math.random() * stage.stageHeight;
    square.alpha = 0.2 + Math.random()*0.5;
    square.rotation = 45;
}
```



ZUFAELLIGES VERTEILEN VON 4 VERSCHIEDENEN QUADRATEN
Die Skalierung und Deckkraft wird anschließend beim ENTER_FRAME-Ereignis über die EventListener-Funktion 'moving' laufend zufällig geändert.

```
var square: MovieClip;
var select: int;

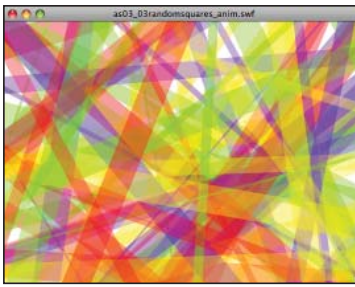
for (var i=0; i<50; i++) {

    select = Math.floor( Math.random()*4);

    switch (select) {
        case 0 :
            square = new SquareBlue();
            break;
        case 1 :
            square = new SquareGreen();
            break;
        case 2 :
            square = new SquareRed();
            break;
        case 3 :
            square = new SquareYellow();
    }
    //if (select==0) {
    //square = new SquareBlue();
    //} else if (select==1) {
    //square = new SquareGreen();
    //} else if (select==2) {
    //square = new SquareRed();
    //} else {
    //square = new SquareYellow();
    //}

    this.addChild(square);

    square.x = Math.random() * stage.stageWidth;
    square.y = Math.random() * stage.stageHeight;
    square.alpha = 0.2 + Math.random()*0.5;
    square.rotation = 45;
    square.addEventListener(Event.ENTER_FRAME, moving);
}
function moving(e:Event):void {
    e.target.alpha = 0.3 + Math.random()*0.1;
    e.target.scaleX = 0.8+Math.random()*0.4;
    e.target.scaleY = 10+Math.random()*0.4;
}
```

ZUFAELLIGES VERTEILEN VON 4 VERSCHIEDENEN QUADRATEN

```
for (var i=0; i<80; i++) {
    var square: MovieClip;
    var select;
    select = Math.floor( Math.random()*4);

    switch ( select ) {
        case 0 :
            square = new SquareBlue();
            break;

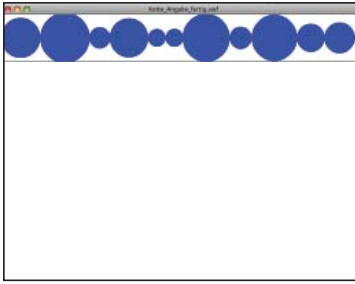
        case 1 :
            square = new SquareGreen();
            break;

        case 2 :
            square = new SquareRed();
            break;

        case 3 :
            square = new SquareYellow();
            //break;
    }
    /*
    if(select==0){
        square = new SquareBlue();
    }
    else if(select==1){
        square = new SquareGreen();
    }
    else if(select==2){
        square = new SquareRed();
    }
    else{
        square = new SquareYellow();
    }*/

    this.addChild(square);

    square.x = Math.random() * stage.stageWidth;
    square.y = Math.random() * stage.stageHeight;
    square.alpha = 0.2 + Math.random()*0.5;
    square.rotation = Math.random()*360;
    square.scaleX = Math.random()*0.6;
    square.scaleY = Math.random()*100;
    square.addEventListener(Event.ENTER_FRAME, moving);
}
function moving(e:Event):void {
    e.target.rotation++;
    e.target.alpha = 0.3 + Math.random()*0.1;
    e.target.scaleX = Math.random()*0.5;
    e.target.scaleY = Math.random()*200;
}
```



UNREGELMÄSSIGE KREISKETTE MIT WHILE-SCHLEIFE

```

var kreis_mc : MovieClip;
var skalierung: Number;
var nextX:int=0;

function zufallszahl(a:int,b:int):int {
    return a + Math.floor(Math.random() * b - a + 1);
}

while (nextX<stage.stageWidth) {
    kreis_mc = new Kreis2();
    this.addChild(kreis_mc);
    kreis_mc.y= stage.stageHeight/2;
    skalierung = Math.random()+0.5;
    kreis_mc.scaleX=kreis_mc.scaleY= skalierung;
    kreis_mc.x=nextX+kreis_mc.width/2;
    nextX=nextX+kreis_mc.width;
}

```

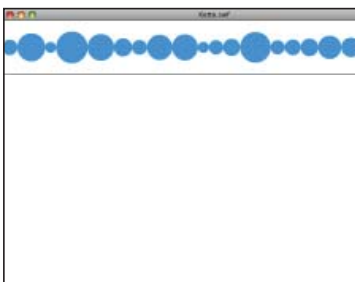


KREISKETTE

```

var figur:MovieClip;
var nextX:int = 0;
var scale:Number;
while (nextX < stage.stageWidth) {
    figur = new Kreis();
    addChild(figur);
    scale = Math.random()+0.5;
    figur.scaleX = figur.scaleY = scale;
    figur.x = nextX + figur.width/2;
    nextX = nextX + figur.width;
    figur.y = stage.stageHeight/2;
}

```



KREISKETTE

```

var kreis:MovieClip;
var posX:Number = 0;
while (posX<stage.stageWidth) {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.y=stage.stageHeight/2;
    kreis.scaleX = kreis.scaleY = 0.5+Math.random();
    if (posX+kreis.width>=stage.stageWidth) {
        kreis.width=kreis.height=stage.stageWidth-posX;
    }
    kreis.x = posX+kreis.width/2;
    posX=posX+kreis.width;
}

```



TRACE EVENTHANDLER
OUTPUT: "HELLO WORLD"

BEISPIEL 1

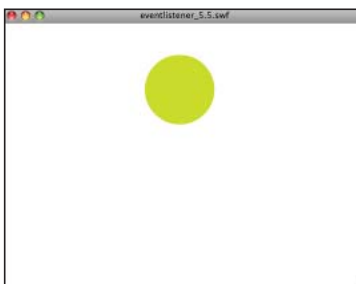
```
function enterFrameListener(e:Event):void {
    trace("Hello World");
}
this.addEventListener(Event.ENTER_FRAME,enterFrameListener);
```

this bezieht sich auf die Maintimeline.

BEISPIEL 2

```
function enterHandler(e:MouseEvent):void {
    trace("Hello World");
}
this.stage.addEventListener(MouseEvent.MOUSE_DOWN,enterHandler);
```

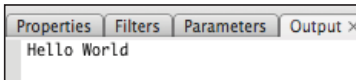
Mit jedem Mausklick auf die stage/Bühne wird Hello world ausgegeben.
In der Maintimeline ist noch nichts drinnen, darum kann sie nicht reagieren.
Man kann aber über die Maintimeline auf die Bühne zugreifen



BEISPIEL 3

```
function enterHandler(e:MouseEvent):void {
    trace("Hello World");
}
this.addEventListener(MouseEvent.MOUSE_DOWN,enterHandler);
```

Jetzt liegt ein Kreis auf der Maintimeline. Wenn ich auf den Kreis-movieclip klicke
wird hello world ausgegeben- Auf die Bühne kann immer zugegriffen werden!!!!

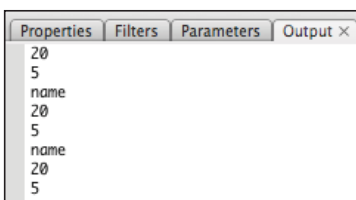


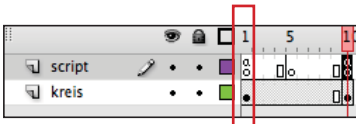
TRACE- FRAME-ABFRAGEN

```
trace (this.totalFrames);
```

Liefert die Anzahl der Frames - Wie lang ist die timeline?
Bei Sprüngen nie Zahlen reinschreiben, sondern benennen und über Namen anfragen.

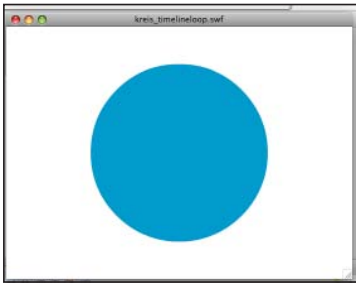
```
trace (this.currentFrame); //Abfrage eines bestimmten Frames
trace (this.currentLabel);
```





ANIMATION: PULSIERENDER KREIS DURCH EINE TIMELINELOOP

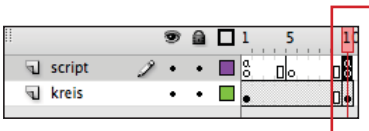
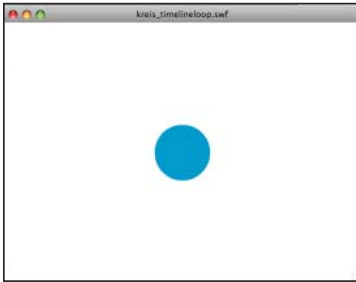
(1) STARTFRAME



```
var maxscale:Number = 8;  
var step = 0.3;  
var richtung:int
```

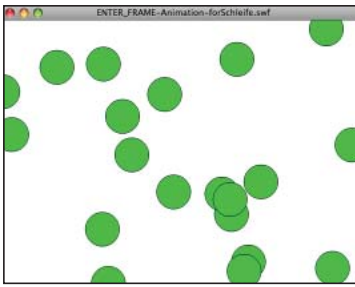
```
if (kreis_mc.scaleY >= maxscale)  
{  
    richtung = -1;  
}  
else if (kreis_mc.scaleY <= 1)  
{  
    richtung = 1;  
}
```

```
kreis_mc.scaleX = kreis_mc.scaleY = kreis_mc.scaleX + step * richtung;
```



(2) ENDFRAME

```
scaleX = scaleY;  
if (kreis_mc.scaleX >= maxscale) {  
    kreis_mc.scaleX = kreis_mc.scaleX - step * richtung;  
    kreis_mc.scaleX = kreis_mc.scaleY;  
} else {  
    kreis_mc.scaleX = kreis_mc.scaleX + step * richtung;  
}
```



ENTER-FRAME-EVENT MIT FOR-SCHLEIFE

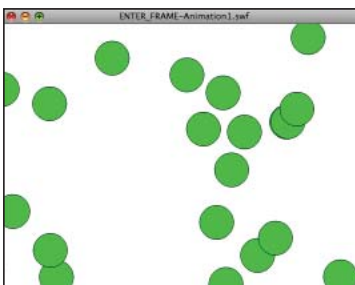
```

var kreis:MovieClip;
var i:int = 0;

this.addEventListener(Event.ENTER_FRAME, drawCircle);

function drawCircle(e:Event):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = Math.random()*stage.stageWidth;
    kreis.y = Math.random()*stage.stageHeight;
    i++;
    if (i >= 20) {
        this.removeEventListener(Event.ENTER_FRAME, drawCircle);
    }
}

```



ENTER-FRAME-EVENT

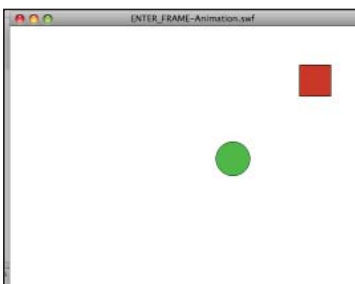
```

var kreis:MovieClip;

this.addEventListener(Event.ENTER_FRAME, drawCircle);

function drawCircle(e:Event):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = Math.random()*stage.stageWidth;
    kreis.y = Math.random()*stage.stageHeight;
    if (this.numChildren == 20) {
        this.removeEventListener(Event.ENTER_FRAME, drawCircle);
    }
}

```



ENTER-FRAME-EVENT

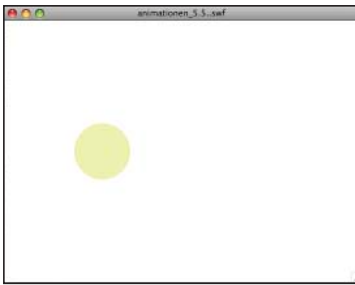
```

var kreis:MovieClip = new Kreis ();
this.addChild(kreis);
kreis.x = kreis.width/2;
kreis.y = stage.stageHeight/2;
var speed:Number = 10;

kreis.addEventListener(Event.ENTER_FRAME, moveCircle);
quadrat.addEventListener(Event.ENTER_FRAME, moveCircle);

function moveCircle(e:Event):void {
    e.target.x += speed;
    if (e.target.x > stage.stageWidth) {
        e.target.removeEventListener(Event.ENTER_FRAME, moveCircle);
    }
}

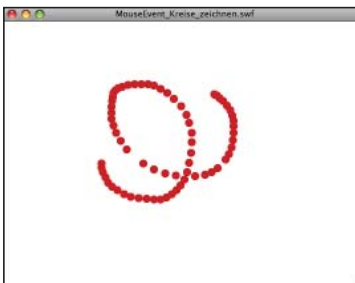
```



ANIMATION:
KREIS BEWEGT SICH VON LINKS NACH RECHTS
DECKKRAFT NIMMT AB

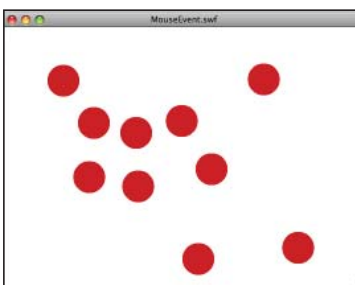
```
var speed: Number = 5;
var deckkraft: Number = 0.01;
var drehen: Number = 0.10;

kreis_mc.x = kreis_mc.x + speed;
kreis_mc.alpha = kreis_mc.alpha - deckkraft;
kreis_mc.rotation = kreis_mc.rotation + drehen;
if (kreis_mc.x >= stage.stageWidth - kreis_mc.width/2) {
    kreis_mc.x = stage.stageWidth - kreis_mc.width/2;
    stop();
} else {
    gotoAndPlay(1);
}
```



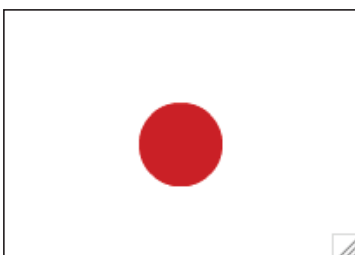
MOUSE-DOWN/UP/MOVE-EVENT - KREISE ZEICHNEN

```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, startDrawing );
function startDrawing(e:MouseEvent):void {
    this.stage.addEventListener(MouseEvent.CLICK, drawCircle);
}
function drawCircle(e:MouseEvent):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = e.stageX;
    kreis.y = e.stageY;
}
this.stage.addEventListener(MouseEvent.CLICK, stopDrawing);
function stopDrawing(e:MouseEvent):void {
    this.stage.removeEventListener(MouseEvent.CLICK, drawCircle);
}
}
```



MOUSE-DOWN/UP-EVENT

```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, drawCircle );
function drawCircle(e:MouseEvent):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = e.stageX;
    kreis.y = e.stageY;
}
}
```



SKRIPT IM MOVIECLIP KREIS:

```
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    this.startDrag(true);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    this.stopDrag();
}
}
```

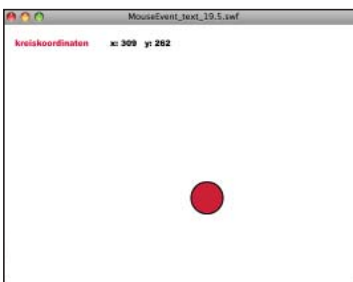
MOUSE-DOWN-EVENT

```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, drawCircle );
function drawCircle(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        kreis = new Kreis();
        this.addChild(kreis);
        kreis.x = e.stageX;
        kreis.y = e.stageY;
    }
}
```

**/ Abfrage: Nur wenn target und current target gleich sind, hat man auf die leere Bühne geklickt.
Wenn ich auf einen Kreis klicke, wird kein neuer Kreis erzeugt,
erst wenn ich auf die Bühne klicke wird ein neuer erzeugt. **/**

VARIANTE: Abfrage der Eventphasen

```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, drawCircle );
function drawCircle(e:MouseEvent):void {
    if (e.eventPhase == EventPhase.AT_TARGET) {
        kreis = new Kreis();
        this.addChild(kreis);
        kreis.x = e.stageX;
        kreis.y = e.stageY;
    }
}
```



ENTER-FRAME-EVENT - DYNAMISCHER TEXT

'koordinaten' ist der Instanzname des dynamischen Textfeldes.

```
koordinaten.addEventListener(Event.ENTER_FRAME, kreiskoordinaten);
function kreiskoordinaten(e:Event):void {
    e.target.text = "x: "+Math.floor(kreis.x)+" y: "+Math.floor(kreis.y);
}
```

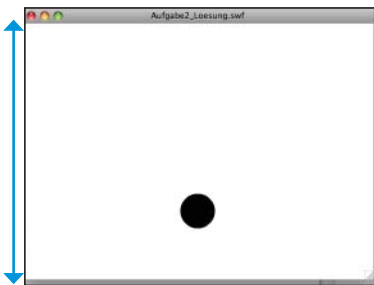
// Mit function Math.floor() oder round () werden nur ganze Zahlen geliefert



MOUSE-MOVE-EVENT - DYNAMISCHER TEXT

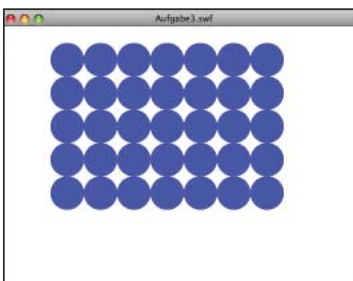
```
// Funktionsdefinition
function checkMousePos():void {
    if (stage.mouseX <= stage.stageWidth/2) {
        if (stage.mouseY <= stage.stageHeight/2) {
            mauspos_txt.text = "LINKS OBEN";
        } else {
            mauspos_txt.text = "LINKS UNTEN";
        }
    } else {
        if (stage.mouseY <= stage.stageHeight/2) {
            mauspos_txt.text = "RECHTS OBEN";
        } else {
            mauspos_txt.text = "RECHTS UNTEN";
        }
    }
}

// Funktionsaufruf
checkMousePos();
this.stage.addEventListener(MouseEvent.CLICK, position);
function position(e:MouseEvent):void {
    checkMousePos();
}
```



MOUSE-DOWN- EVENT- PENDEL - KREIS

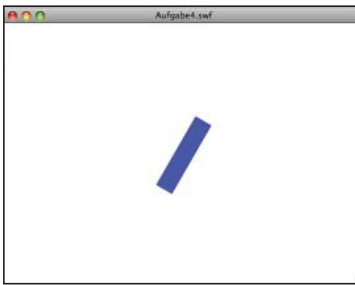
```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, drawCircle );
function drawCircle(e:MouseEvent):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = e.stageX;
    kreis.y = e.stageY;
}
```



KREISREIHEN -WHILE / FOR-SCHLEIFEN

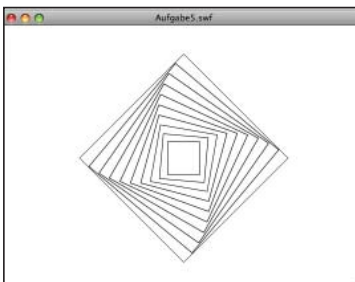
```
function kette (xPos:int, yPos:int, n:int):void {
    var kreis:MovieClip;
    var i:int = 0;
    while (i<n) {
        kreis = new Kreis();
        this.addChild(kreis);
        kreis.x = xPos + i*kreis.width;
        kreis.y = yPos;
        i++;
    }
}

for (var j:int = 0; j<5; j++) {
    kette (100, 50+j*50, 7);
}
```

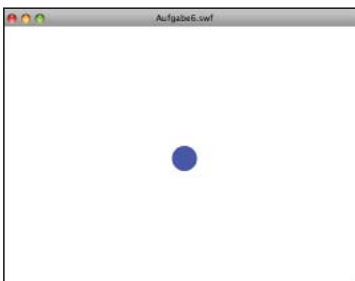
MOUSE-UP/DOWN-EVENT / ENTER FRAME : DREHBALKEN (Ü4)

```
rechteck.addEventListener(MouseEvent.CLICK, startRotate);
function startRotate(e:MouseEvent):void {
    rechteck.addEventListener(Event.ENTER_FRAME, rotate);
}
function rotate(e:Event):void {
    e.target.rotation += 5; // auch rechteck.rotation += 5; möglich
}
this.stage.addEventListener(MouseEvent.CLICK, stopRotate);
function stopRotate(e:MouseEvent):void {
    rechteck.removeEventListener(Event.ENTER_FRAME, rotate);
}
```



GENERATIVE GRAFIK

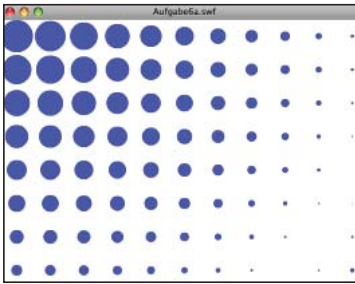
```
var quadrat:MovieClip;
for (var i:int = 0; i<10; i++) {
    quadrat = new Quadrat ();
    this.addChild(quadrat);
    quadrat.x = stage.stageWidth/2;
    quadrat.y = stage.stageHeight/2;
    quadrat.scaleX = quadrat.scaleY = 1 + i*0.4;
    quadrat.rotation = i * 5;
}
```



MOUSE-MOVE-EVENT - KREISSKALIERUNG-MAUS

SKRIPT IM MOVIECLIP KREIS:

```
this.stage.addEventListener(MouseEvent.CLICK, scaleCircle);
function scaleCircle(e:MouseEvent):void {
    var distance:Number = Point.distance(new Point(stage.mouseX, stage.
mouseY), new Point(this.x, this.y));
    this.scaleX = this.scaleY = 1 - distance / stage.stageWidth;
}
```

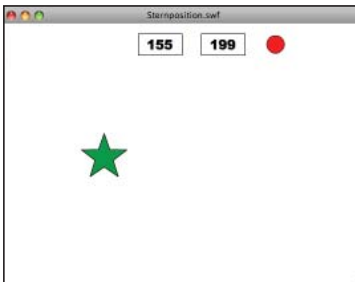


MOUSE-MOVE-EVENT - Wabbelkreise

```
function kette (xPos:int, yPos:int, n:int):void {
    var kreis:MovieClip;
    var i:int = 0;
    while (i<n) {
        kreis = new Kreis();
        this.addChild(kreis);
        kreis.x = xPos + i*kreis.width;
        kreis.y = yPos;
        i++;
    }
}
for (var j:int = 0; j<8; j++) {
    kette (25, 25+j*50, 11);
}
```

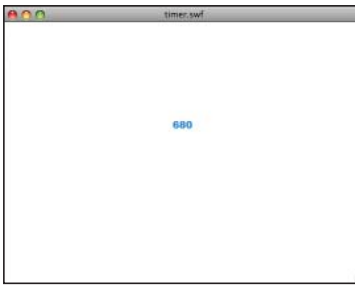
SKRIPT IM MOVIECLIP KREIS:

```
this.stage.addEventListener(MouseEvent.MOUSE_MOVE, scaleCircle);
function scaleCircle(e:MouseEvent):void {
    var distance:Number = Point.distance (new Point(stage.mouseX, stage.
mouseY), new Point(this.x, this.y));
    this.scaleX = this.scaleY = 1 - distance / stage.stageWidth;
//    this.scaleX = this.scaleY = 1 - distance / Math.sqrt(Math.pow(stage.
stageWidth,2)+Math.pow(stage.stageHeight,2));
}
```



STERNPOSITION - EINGABETEXT+BUTTON DURCH USER

```
xPos.text = Math.round(stern.x).toString();
//xPos.text = ""+stern.x;
yPos.text = Math.round(stern.y).toString();
//xPos.addEventListener(Event.ENTER_FRAME, setXPos);
//function setXPos (e:Event):void {
//stern.x = Number(xPos.text);
//}
//yPos.addEventListener(Event.ENTER_FRAME, setYPos);
//function setYPos (e:Event):void {
//stern.y = Number(yPos.text);
//}
ok_btn.addEventListener(MouseEvent.MOUSE_UP, setPos);
function setPos(e:MouseEvent):void {
    stern.x = Number(xPos.text);
    stern.y = Number(yPos.text);
}
```



TIMEREVENT

```
var timer:Timer = new Timer(10,1000);
// Wenn es nur einen Zeitparameter gibt, lauft timer endlos: new Timer (10)

timer.addEventListener(TimerEvent.TIMER, timerListener);
function timerListener(e:TimerEvent):void {
    timer_txt.text = (1000 - timer.currentCount).toString();
    e.updateAfterEvent;
}

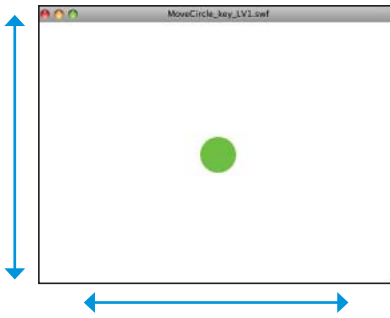
timer.start();

/* weitere Funktionen
timer.reset();
timer.stop(); */
```



TIMEREVENT- Stoppuhr mit RESET/STOP/PLAY BUTTON

```
zeit_txt.text = "0";
var timer:Timer = new Timer(10);
timer.addEventListener(TimerEvent.TIMER, timerListener);
function timerListener(e:TimerEvent):void {
    zeit_txt.text = timer.currentCount.toString();
    e.updateAfterEvent();
}
start_btn.addEventListener(MouseEvent.CLICK, startZeit);
function startZeit(e:MouseEvent):void {
    timer.start();
}
stop_btn.addEventListener(MouseEvent.CLICK, stopZeit);
function stopZeit(e:MouseEvent):void {
    timer.stop();
}
reset_btn.addEventListener(MouseEvent.CLICK, resetZeit);
function resetZeit(e:MouseEvent):void {
    timer.reset();
    zeit_txt.text = "0";
}
```



KEYBOARD-EVENT MIT SWITCH-ANWEISUNG

```

this.stage.addEventListener(KeyboardEvent.KEY_DOWN, bewegeKreis);
function bewegeKreis(e:KeyboardEvent):void {
    //trace (e.keyCode);
    switch (e.keyCode) {
        case Keyboard.LEFT :
            kreis.x--;
            break;
        case Keyboard.RIGHT :
            kreis.x++;
            break;
        case Keyboard.UP :
            kreis.y--;
            break;
        case Keyboard.DOWN :
            kreis.y++;
            break;
    }
}

```

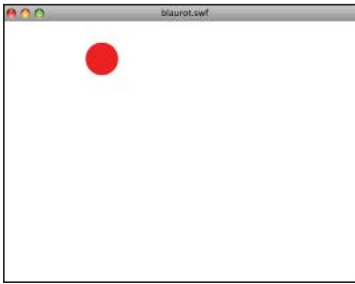
KEYBOARD-EVENT

Bei gedrückter Shift-Taste wird die Schrittweite auf 10 vergrößert.

```

this.stage.addEventListener(KeyboardEvent.KEY_DOWN, bewegeKreis);
function bewegeKreis(e:KeyboardEvent):void {
    //trace(e.keyCode);
    switch (e.keyCode) {
        case Keyboard.LEFT :
            if (e.shiftKey) {
                kreis.x -= 10;
            } else {
                kreis.x--;
            }
            break;
        case Keyboard.UP :
            kreis.y--;
            break;
        case Keyboard.RIGHT :
            kreis.x++;
            break;
        case Keyboard.DOWN :
            kreis.y++;
            break;
    }
    e.updateAfterEvent();
}

```

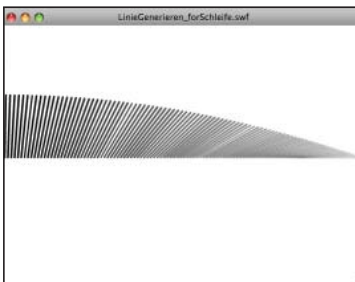
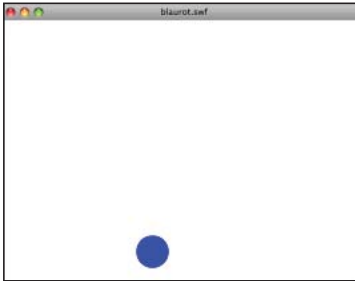


MOUSE-MOVE-EVENT: ROTER KREIS OBEN / BLAUER KREIS UNTEN

```

var blau:MovieClip = new Blau();
var rot:MovieClip = new Rot();
this.addChild(blau);
this.addChild(rot);
blau.x = rot.x = stage.mouseX;
blau.y = rot.y = stage.mouseY;
if (stage.mouseY > stage.stageHeight/2) {
    rot.visible = false;
}
this.stage.addEventListener(MouseEvent.MOUSE_MOVE, moveCircle);
function moveCircle(e:MouseEvent):void {
    blau.x = rot.x = stage.mouseX;
    blau.y = rot.y = stage.mouseY;
    if (stage.mouseY > stage.stageHeight/2) {
        rot.visible = false;
    } else {
        rot.visible = true;
    }
}

```



GENERATIVE GRAFIK

```

var linie:MovieClip;

for (var i:int=0; i<=90; i++) {
    linie = new Linie();
    linie.y=stage.stageHeight/2;
    linie.x=5*i;
    linie.rotation = i;
    linie.alpha = (100 -i + 1) /100;

    addChild(linie);
}

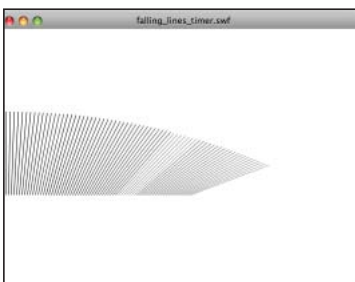
```

```

var linie:MovieClip;
var posX:int = 10;

for (var i = 0; i<=90; i++) {
    linie = new Linie();
    this.addChild(linie);
    linie.x = posX;
    linie.y = 250;
    posX +=4;
    linie.rotation = i;
    linie.alpha = 1-i*0.01;
}

```



TIMER EVENT: GENERATIVE GRAFIK baut sich mit der Zeit auf.

```

var linie:MovieClip;
var posX:int = 10;
var i:int = 0;

var timer:Timer = new Timer(1,91);
timer.addEventListener(TimerEvent.TIMER, drawLine);
timer.start();

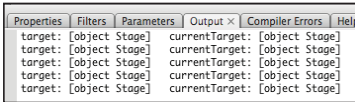
function drawLine(e:TimerEvent):void {
    linie = new Linie();
    this.addChild(linie);
    linie.x = posX;
    linie.y = 250;
    posX +=4;
    linie.rotation = i;
    linie.alpha = 1-i*0.01;
    i++;
    e.updateAfterEvent();
}

```



MOUSE-DOWN-EVENT: CREATE OBJECTS CLICK

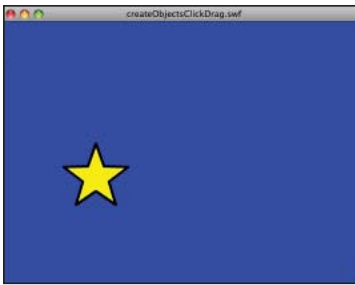
```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    var stern:MovieClip = new GelberStern();
    this.addChild(stern);
    stern.x = e.stageX;
    stern.y = e.stageY;
}
```



MOUSE-DOWN-EVENT: CREATE OBJECTS CLICK

NUR BEIM KLICK AUF DIE LEERE BÜHNE WIRD EIN STERN ERZEUGT.

```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    trace("target: " + e.target + " currentTarget: " + e.currentTarget);
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}
```



MOUSE-DOWN-EVENT: CREATE OBJECTS CLICK AND DRAG
Erzeugte Sterne lassen sich bei gedrückter Maustaste verschieben.

```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}
```

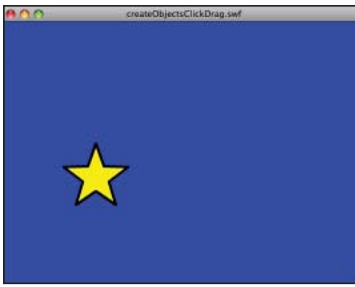
SKRIPT IM MOVIECLIP STERN:

```
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    this.startDrag(true);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    this.stopDrag();
}
```

MOUSE-DOWN-EVENT: CREATE OBJECTS CLICK AND DRAG
Beide Skripts sind in der Maintimeline platziert.

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste verschieben
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
}
```



MOUSE-DOWN/UP-EVENT: CREATE OBJECTS CLICK & DRAG & DELETE (A)

```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}
```

SKRIPT IM MOVIECLIP STERN:

```
// Stern bei gedrückter Maustaste bewegen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
}
// Stern bei gedrückter Strg-Taste durch Anklicken löschen
this.addEventListener(MouseEvent.CLICK, loeschen);
function loeschen(e:MouseEvent):void {
    if (e.ctrlKey == true) {
        e.target.parent.removeChildAt(e.target.parent.getChildIndex(e.target));
    }
}
```

MOUSE-DOWN/UP-EVENT: CREATE OBJECTS CLICK & DRAG & DELETE (B)

```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}
```

IM MOVIECLIP STERN:

```
// Stern bei gedrückter Maustaste bewegen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
}
// Stern bei gedrückter Strg-Taste durch Anklicken löschen
this.addEventListener(MouseEvent.CLICK, loeschen);
function loeschen(e:MouseEvent):void {
    if (e.ctrlKey == true) {
        parent.removeChildAt(parent.getChildIndex(this));
    }
}
```

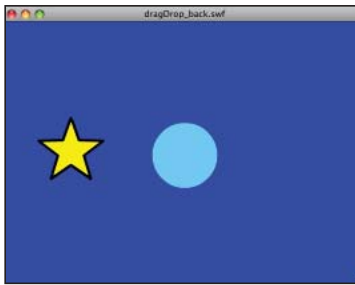



MOUSE-DOWN/UP-EVENT: CREATE OBJECTS CLICK & DRAG & DELETE & DOUBLECLICK (A)

```
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}
// Movieclips durch Doppelklick löschen
this.addEventListener(MouseEvent.DOUBLE_CLICK, loeschen);
function loeschen(e:MouseEvent):void {
    e.target.parent.removeChildAt(e.target.parent.getChildIndex(e.target));
}
}
```

SKRIPT IM MOVIECLIP STERN:

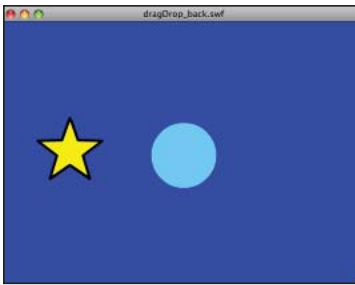
```
this.doubleClickEnabled = true;
// Stern bei gedrückter Maustaste bewegen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
}
}
```



MOUSE-DOWN/UP-EVENT: DRAG/DROP_BACK (A)

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste in den hellblauen Kreis ziehen
var startPosX: Number;
var startPosY: Number;
var hitTest:Boolean;
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
    startPosX = e.target.x;
    startPosY = e.target.y;
    if (e.target.hitTestObject(ziel)) {
        hitTest = true;
    } else {
        hitTest = false;
    }
}
this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
    if (e.target.hitTestObject(ziel)) {
        e.target.x = ziel.x;
        e.target.y = ziel.y;
    } else {
        if (hitTest == false) {
            e.target.x = startPosX;
            e.target.y = startPosY;
        }
    }
}
}
```

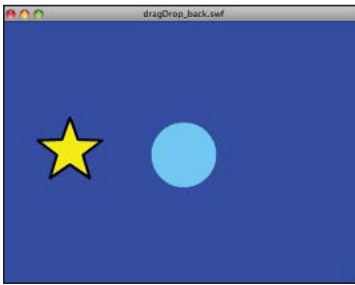


MOUSE-DOWN/UP-EVENT: DRAG/DROP_BACK (B)

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste in den hellblauen Kreis ziehen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
    e.target.startPosX = e.target.x;
    e.target.startPosY = e.target.y;
    if (e.target.hitTestObject(ziel)) {
        e.target.hitTest = true;
    } else {
        e.target.hitTest = false;
    }
}

this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
    if (e.target.hitTestObject(ziel)) {
        e.target.x = ziel.x;
        e.target.y = ziel.y;
    } else {
        if (e.target.hitTest == false) {
            e.target.x = e.target.startPosX;
            e.target.y = e.target.startPosY;
        }
    }
}
}
```

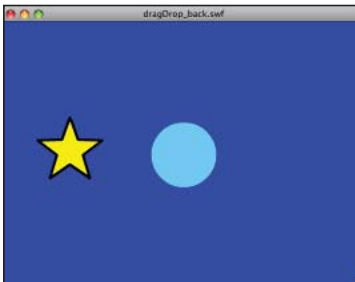


MOUSE-DOWN/UP-EVENT: DRAG/DROP (A)

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste in den hellblauen Kreis ziehen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}

this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
    if (e.target.hitTestObject(ziel)) {
        e.target.x = ziel.x;
        e.target.y = ziel.y;
    }
}
}
```

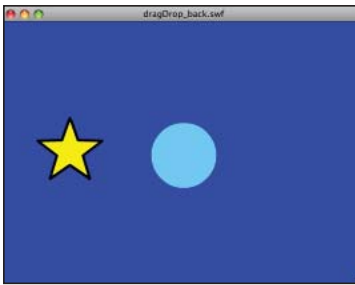


MOUSE-DOWN/UP-EVENT: DRAG/DROP (B)

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste in den hellblauen Kreis ziehen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}

this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
    if (e.target.hitTestPoint(ziel.x,ziel.y,true)) {
        e.target.x = ziel.x;
        e.target.y = ziel.y;
    }
}
}
```

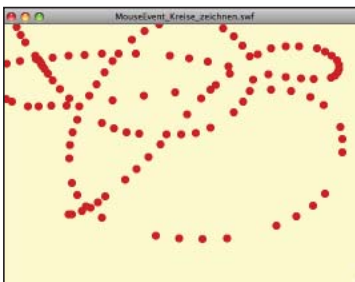


MOUSE-DOWN/UP-EVENT: DRAG/DROP (C)

```
// Stern durch Anklicken der Bühne erzeugen
this.stage.addEventListener(MouseEvent.CLICK,erzeugeStern);
function erzeugeStern(e:MouseEvent):void {
    if (e.target == e.currentTarget) {
        var stern:MovieClip = new GelberStern();
        this.addChild(stern);
        stern.x = e.stageX;
        stern.y = e.stageY;
    }
}

// Stern bei gedrückter Maustaste in den hellblauen Kreis ziehen
this.addEventListener(MouseEvent.CLICK, startZiehen);
function startZiehen(e:MouseEvent):void {
    e.target.startDrag(false);
}

this.addEventListener(MouseEvent.CLICK, stopZiehen);
function stopZiehen(e:MouseEvent):void {
    e.target.stopDrag();
    if (ziel.getRect(this.stage).containsRect(e.target.getRect(this.stage))) {
        e.target.x = ziel.x;
        e.target.y = ziel.y;
    }
}
}
```



MOUSE-DOWN/UP-EVENT: KREISE ZEICHNEN (A)

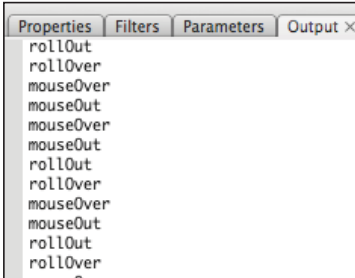
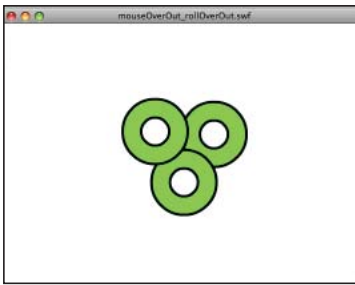
```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, drawCircle);
function drawCircle(e:MouseEvent):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = e.stageX;
    kreis.y = e.stageY;
}
}
```

MOUSE-DOWN/UP-EVENT: KREISE ZEICHNEN (B)

```
var kreis:MovieClip;
this.stage.addEventListener(MouseEvent.CLICK, startDrawing );
function startDrawing(e:MouseEvent):void {
    this.stage.addEventListener(MouseEvent.CLICK, drawCircle);
}

function drawCircle(e:MouseEvent):void {
    kreis = new Kreis();
    this.addChild(kreis);
    kreis.x = e.stageX;
    kreis.y = e.stageY;
}

this.stage.addEventListener(MouseEvent.CLICK, stopDrawing);
function stopDrawing(e:MouseEvent):void {
    this.stage.removeEventListener(MouseEvent.CLICK, drawCircle);
}
}
```



MOUSE-OVER-OUT/ROLL-OVER-OUT-EVENT:

// Stern durch Anklicken der Bühne erzeugen

```
ringe.addEventListener(MouseEvent.CLICK,mouseOverHandler);
ringe.addEventListener(MouseEvent.CLICK,mouseOutHandler);
ringe.addEventListener(MouseEvent.CLICK,rollOverHandler);
ringe.addEventListener(MouseEvent.CLICK,rollOutHandler);
function mouseOverHandler (e:MouseEvent):void {
    trace("mouseOver");
}
function mouseOutHandler (e:MouseEvent):void {
    trace("mouseOut");
}
function rollOverHandler (e:MouseEvent):void {
    trace("rollOver");
}
function rollOutHandler (e:MouseEvent):void {
    trace("rollOut");
}
```

