

Tid programmeret med Actionsript

Ligesom vi kan styre tiden ved hjælp af timeline, så kan vi gøre noget ud fra tiden i actionsript. Det er selvfølgelig en helt anden måde at gøre det på, så vi vil starte med at lave noget der lige så godt kunne laves i timeline. Det er udelukkende for at lære princippet.

Opgave 14a

Tegn en lille dims og konverter den til et symbol af typen Movie Clip.

Giv den navnet min (Instance Name).

Opret et lag ved navn Actionsript og skriv følgende kode:

```
var speed:Number = 10; // Dette er en variabel, der angiver hastigheden

min.addEventListener(Event.ENTER_FRAME , boldAnimation);

function boldAnimation(event:Event):void
{
    min.y = min.y + speed;
    min.x = min.x + 2;
}
```

Test det.

Opgave 14b

Det der sker i Opgave 14a er at den kører skråt ned over scenen 1 gang, og så sker der ikke mere.

Prøv at tilføje følgende kode inde i funktionen:

```
if ((min.y > 400) || (min.y < 0)) {
    speed = speed * -1;
}

if (min.x > 550) {
    min.x = 0;
}
```

Test det og se hvordan det har ændret opførsel.

Prøv at rette i koden, så dit symbol bliver inden for rammen hele tiden, og ikke går halvt ud over kanten.

Opgave 14c

Tegn et nyt objekt og konverter det til et symbol med navnet din.

Opret to nye variabler i actionsriptet under variabelen speed som følger:

```
var dinSpeedX:Number = 5;
var dinSpeedY:Number = 5;
```

Inde i funktionen tilføjes der så din også bevæger sig ved:

```
din.y += dinSpeedY;
din.x += dinSpeedX;
```

Og der tilføjes så den holder sig inden for rammerne igen:

```
if ((din.y > 400) || (din.y < 0)) {
    dinSpeedY *= -1;
}

if ((din.x > 550) || (din.x < 0)) {
    dinSpeedX *= -1;
}
```

Korriger igen så det er kanten den bøje af på.

Opgave 14d

Nu prøver vi at tilføje noget der er mere komplekst, nemlig at de reagerer på **hinanden**.

Inde i funktionen tilføjer vi følgende kode vi har flyttet med objekterne:

```
if (min.hitTestObject(din)) {
    if (din.x > min.x) {
        if (dinSpeedX < 0) {
            dinSpeedX *= -1;
        } else {
            if (din.y > min.y) {
                if (speed > 0) {
                    speed *= -1;
                } else {
                    dinSpeedY *= -1;
                }
            } else {
                if (speed < 0) {
                    speed *= -1;
                } else {
                    dinSpeedY *= -1;
                }
            }
        }
    }
} else {
    if (dinSpeedX > 0) {
        dinSpeedX *= -1;
    } else {
        if (din.y > min.y) {
            if (speed > 0) {
                speed *= -1;
            } else {
                dinSpeedY *= -1;
            }
        } else {
            if (speed < 0) {
                speed *= -1;
            } else {
                dinSpeedY *= -1;
            }
        }
    }
}
```

De ting vi har lavet før har indtil nu har også kunne være lavet traditionelt i timelinen, relativt enkelt med nogen bevægelser der tweener inden i hinanden.

Denne teknik har indtil nu bare være anderledes at vi regnede os frem til det.

Det vi lige har tilføjet kan simpelthen ikke lade sig gøre bare ved at tegne i timelinen, nemlig at få de to objekter til at reagere på **hinanden**.

Opgave 14e

I det foregående var brugeren slet ikke indblandet.

Her giver vi brugeren muligheden for at generere objekterne, ved at give muligheden for at flytte rundt med dem.

Det første vi gør er, at vi opretter to variabler, der kan stoppe tidsbevægelsen af min og din. Det kræver følgende kode:

```
var moveMin:Boolean = true;
var moveDin:Boolean = true;
```

Inde i funktionen boldAnimation skal vi så stoppe bevægelserne hvis de to variabler ikke er true.

Det gøres ved følgende rettelse – vær opmærksom på det skal ikke bare sættes ind – de 4 linier der laver bevægelse er der i forvejen, det er kun if-strukturen der er koblet uden om:

```
if (moveMin) {  
    min.y = min.y + speed;  
    min.x = min.x + 2;  
}  
if (moveDin) {  
    din.y += dinSpeedY;  
    din.x += dinSpeedX;  
}
```

For at brugeren kan trække rundt med objekterne bruges startDrag() og stopDrag() der skal kodes ind via MouseEvents på objektet. Det gøres med følgende kode:

```
min.addEventListener(MouseEvent.CLICK, dragMin);  
min.addEventListener(MouseEvent.CLICK, stopDragMin);  
  
function dragMin(event:MouseEvent):void {  
    moveMin = false;  
    min.startDrag();  
}  
  
function stopDragMin(event:MouseEvent):void {  
    moveMin = true;  
    min.stopDrag();  
}
```

Dette løser det kun for det ene objekt.

Prøv selvstændigt at kode de eventlistenere og funktioner der skal til for at trække rundt med det andet objekt.

Hvis du har problemer kan du altid søge hjælp i det eksempel jeg har med.

Læg mærke til, at selvom det er tid vi har arbejdet med i denne opgave, så er alt det der sker lavet i Frame 1