



Slideshow - Guide

Denne guide gennemgår grundigt og forklarende hvordan man i Flash laver et slideshow med simpel fade-effekt med billeder hentet dynamisk. MX og MX 2004

Skrevet den **10. Feb 2009** af **pyroman** | kategorien **Flash / Generelt** | ★★★★★

Det har været meget efterspurgt her i Flash kategorien hvordan man laver et slideshow med billeder hentet dynamisk.

Jeg har derfor sat mig ned en kedelig søndag aften og givet mig til at hjælpe de stakkels kræ, der endnu måtte sidde og rode med dette problem :)

I denne guide vil jeg forklare hvordan man stort set udelukkende med actionscript laver et billedeslideshow. Det er forudsat at man kender en smule til actionscript og brugen af funktioner, samt brugen af Flash's symboler som MovieClips og Buttons.

Koden i denne guide vil kun virke i Flash MX 2003 og MX 2004 - ikke ældre versioner!

FOR DE DOVNE - START MED AT LÆSE HER!:

Forberedelse:

De billeder vi vil have med i slideshowet ligger vi i en mappe med navnet "images". Billederne skal have navnene "1","2","3" osv...

VIGTIGT! Billeder der hentes ind i flash kan kun være af typen JPEG og fil-endelserne skal derfor være ".jpg" altså fx "1.jpg"

Vi vil oprette vores slideshow i et MC (MovieClip) - dette gør det nemmere at flytte rundt på ændre i. Så...

1. Lav et nyt MovieClip symbol og giv det navnet "Slideshow"

2. Åbn Actions panelet for første frame i det nye MC

Vi vil nu lave en funktion der henter alle billederne ind i MC'et. Når man henter et billede ind, henter man det ind i et MC og der kan kun være ét billede i hvert MC. Alt dette kan heldigvis gøres i Actionscript - så man slipper for at lave andet end at trykke på tastaturet...

```
this.loadImages = function(num){
    for(i=1;i<=num;i++){ //For-løkke - kører efterfølgende script igennem
'num' antal gange
        this.createEmptyMovieClip("img"+i, i); //Laver et nyt tomt MC
        this["img"+i].loadMovie(path+i+".jpg"); //Henter billedet ind i det
nye MC
    }
}
```

Variablen *path* skal sættes udenfor funktionen og skal indeholde en streng med stien til mappen hvori vores billeder ligger i - i vores tilfælde ville det være: *path = "images/"* - den sidste "/" er vigtig, ellers ved flash ikke at det er en mappe.

Hvis man kører denne funktion vil alle vores billeder blive vist efterhånden som de er blevet hentet og lægges ovenpå hinanden. Vi vil godt have at alle billeder er blevet hentet før der sker noget og billederne skal først skjules:

```
this.loadImages = function(num) {
    this._visible = false; //Skjuler 'Slideshow' MC'et så man ikke kan se
    billederne
    for (i=1; i<=num; i++) {
        this.createEmptyMovieClip("img"+i, i);
        this["img"+i].loadMovie("images/"+i+".jpg");
    }
    this.onEnterFrame = function() { //Efterfølgende kode gentages hvert frame
        loadedBytes = 0;
        totalBytes = 0;
        gotTotal = true;
        for (i=1; i<=num; i++) {
            loadedBytes += this["img"+i].getBytesLoaded();
            //Finder hvormeget af det aktuelle billede der er hentet
            totalBytes += this["img"+i].getBytesTotal();
            //Finder hvor meget billedet fylder i alt
            if (this["img"+i].getBytesTotal()<1) {
                gotTotal = false;
                //Finder ud af om flash har fundet ud af hvor meget fylder
            }
        }
        if (loadedBytes == totalBytes && gotTotal) { //Tjekker om alle
billeder er hentet
            delete this.onEnterFrame; //Sletter denne funktion så den ikke
gentages
            for (i=1; i<=num; i++) {
                this["img"+i]._alpha = 0; //Skjuler alle billederne
            }
            this._visible = true; //Viser 'Slideshow' MC'et igen
            //Følgende variabler bruges i funktion til at vise billeder
            this.lastNum = 0;
            this.currentNum = 0;
            showImage(1); //Viser første billede
        }
    };
};
```

Så! Nu har vi fatisk hentet alle billederne ind - nu kan vi komme til selve Slideshow effekten. Det skal lige siges at det er svært at lave en preloader til alle billederne samlet. Dette skyldes at flash ikke finder ud af hvor meget et billedet fylder før den begynder at hente på dette, og flash kan kun hente på 2-3 billeder ad gangen, man vil derfor opnå en ret upræcis loading process der fx kan finde på at springe fra 90% til 50%

fordi den fandt ud af hvor meget næste billede fyldte. Når flash endnu ikke ved hvor meget billedet fylder bliver *getBytesTotal()* bare 0, og da den så heller ikke har hentet noget af billedet bliver *loadedBytes* og *totalBytes* det samme og flash tror at den har hentet alle billederne og går derfor videre... Men den har jo ikke hentet alle billederne - derfor er det nødvendigt at lave variabelen *gotTotal*, der hvis bare ét af billederne ikke er en kendt filstørrelse sættes til *false*, hvorved flash ikke går videre i funktionen...

Denne skal fade det gamle billede ud og det nye ind.

Vi kan lave denne funktion:

```
this.showImage = function(num){
    this["img"+lastNum]._alpha = 0; //Gør sidste bageste billede gennemsigtigt
    this["img"+lastNum]._visible = false; //Skjuler sidste bageste billede
    this.lastNum = currentNum; //Gør nuværende billede til sidste billede
    //Sætter nuværende billede:
    if(num > this.maxNum){
        this.currentNum = 1;
    }else if(num < 1){
        this.currentNum = this.maxNum;
    }else{
        this.currentNum = num;
    }
    this["img"+currentNum]._visible = true; //Viser nye billede
    this["img"+currentNum]._alpha = 0; //Gør nye billede gennemsigtigt
    this["img"+lastNum]._alpha = 100; //Gør sidste billede helt synligt
    this.onEnterFrame = function(){
        this["img"+currentNum]._alpha += speed; //Fader nye billede ind
        this["img"+lastNum]._alpha -= speed; //Fader sidste billede ud
        if(this["img"+currentNum]._alpha >= 100){ //Tjekker om nye billede er
fadet ind
            delete this.onEnterFrame;
            if(automatic){
                //tidsstyrer slideshow:
                slideshowTimer = setInterval(doSlideshow, interval);
            }
            this["img"+lastNum]._visible = false; //Skjuler sidste billede
        }
    }
}
```

Så nu har vi lavet en funktion der kan fade et nyt billede ind... man kan bare angive nummeret på det ønskede billede - så fader den automatisk det gamle ud og det nye ind.

Ved hvert frame øges det nye billedes *_alpha* med *speed* - denne variabel skal angives foroven af scriptet og er et tal mellem 0 og 100.

Linjen *slideshowTimer = setInterval(doSlideshow, interval);* bruger metoden *setInterval()*, der kan bruges til at udføre en funktion efter et specielt tidsinterval (her variabelen *interval*), der angives i millisekunder. Metoden gemmer en referance i variabelen *slideshowTimer*, der skal bruges for at nulstille et interval (ellers vil den blive ved med at kalde funktionen).

For at slideshowet kører automatisk skal man lave en variabel med navnet *automatic* med værdien *true* (eller *false* hvis slideshowet ikke skal køre).

Vi mangler faktisk kun den funktion intervallet kalder, nemlig funktionen med navnet *doSlideshow*. Denne funktion skal slette sidste interval og vise næste billede:

```
this.doSlideShow = function(){
    clearInterval(slideshowTimer); //nulstiller interval, så den ikke kører videre
    showImage(currentNum+1); //Viser næste billede
}
```

Til at vise næste billede har vi variabelen *currentNum* som funktionen *showImage* sætter. Næste billede må så være dette nummer plus én - og det er så hvad vi gør ;)

Hvis vi opstiller alle vores variabler er vi faktisk klar til at køre!

Her kommer så hele scriptet:

```
speed = 10;
path = "images/";
interval = 3000;
automatic = true;
this.loadImages = function(num) {
    this.maxNum = num;
    this._visible = false;
    for (i=1; i<=num; i++) {
        this.createEmptyMovieClip("img"+i, i);
        this["img"+i].loadMovie(path+i+".jpg");
    }
    this.onEnterFrame = function() {
        loadedBytes = 0;
        totalBytes = 0;
        gotTotal = true;
        for (i=1; i<=num; i++) {
            loadedBytes += this["img"+i].getBytesLoaded();
            totalBytes += this["img"+i].getBytesTotal();
            if (this["img"+i].getBytesTotal()<1) {
                gotTotal = false;
            }
        }
        if (loadedBytes == totalBytes && gotTotal) {
            delete this.onEnterFrame;
            for (i=1; i<=num; i++) {
                this["img"+i]._alpha = 0;
                this["img"+i]._visible = false;
            }
            this._visible = true;
            this.lastNum = 0;
            this.currentNum = 0;
            showImage(1);
        }
    };
};
```

```

};

this.showImage = function(num){
    this["img"+lastNum]._alpha = 0;
    this["img"+lastNum]._visible = false;
    this.lastNum = currentNum;
    if(num > this.maxNum){
        this.currentNum = 1;
    }else if(num < 1){
        this.currentNum = this.maxNum;
    }else{
        this.currentNum = num;
    }
    this["img"+currentNum]._visible = true;
    this["img"+currentNum]._alpha = 0;
    this["img"+lastNum]._alpha = 100;
    this.onEnterFrame = function(){
        this["img"+currentNum]._alpha += speed;
        this["img"+lastNum]._alpha -= speed;
        if(this["img"+currentNum]._alpha >= 100){
            delete this.onEnterFrame;
            if(automatic){
                slideshowTimer = setInterval(doSlideshow, interval);
            }
            this["img"+lastNum]._visible = false;
        }
    }
}

this.doSlideShow = function(){
    clearInterval(slideshowTimer);
    showImage(currentNum+1);
}

//Her startes slideshowet - med 4 billeder
loadImages(4);

```

I dette eksempel fader billedet frem med 10% hvert frame, for at stå synligt i 3000 millisekunder (= 3 sekunder). Billederne ligger i mappen "images".

Til sidst i scriptet kalder jeg til slut den funktion der starter det hele - nemlig *loadImages()*, hvori jeg har angivet hvor mange billeder der er i mappen. Denne henter billederne ind, og når dette er færdigt viser den det første billede, hvorefter slideshowet vil køre automatisk.

Man kunne også have udeladt at kalde funktionen her i 1. frame, hvis man ikke ville have at slideshowet skulle starte med det samme - man kunne fx også sætte den på en knap i *_root*, så skal man bare lige huske at referere til slideshow MC'et.

Man kan også bruge funktionen *showImage()* på fx knapper til at vise næste og sidste billede:

```

showImage(currentNum+1); //Viser næste billede

showImage(currentNum-1); //Viser sidste billede

```

Når man bruger disse funktioner skal man huske at referere til slideshow MC'et både på selve funktionen og på *currentNum* variabelen...

Det var det - Håber i kunne bruge det til noget!!

MVH Pyroman

Kommentar af larsholmgaard_dk d. 27. Mar 2004 | 1

Supergodt formuleret, meget professionelt og det er tydeligt at se der er lagt et stort stykke arbejde i artiklen - det hele er meget gennemarbejdet. Dét giver jeg topkarakter for. At man kan få artiklen for sølle 5 point (og kun 2,5 for PRO) er da helt i topklasse.

Kommentar af jmc d. 26. Feb 2006 | 2

Tak for en god artikel, men det var ikke lige hvad jeg søgte.

Jeg var på udkik efter en måde at loade billeder dynamisk samtidigt med at næste billede preloads i slideshowet.

Hvor det også er muligt at lave mapper der generere en menu i flashen og opdatere det hele på en nem og enkel måde.

På bejerholm.dk kan filename hedde hvadsomhelst.jpg, og det hele hentes via

<http://www.bejerholm.dk/flash.asp>

men det er ikke nemt at få preload til at virke med dynamisk opdatering.

Kommentar af alexander_j d. 02. Jan 2005 | 3

Ser fint ud :-D

Kommentar af michael-k d. 26. Mar 2004 | 4

Det ville måske være en ide at fortælle at eksemplet ikke virker i gamle versioner af flash, ingen vi "køber". Lidt trælst for os der bruger 5eren :(

Eller endnu bedre fortæl hvad der ikke virker i ældre og hvorfor, og lav et eksempel der vil virke :)

Kommentar af barklund d. 31. Mar 2004 | 5

Jeg ville personligt lave de anonyme onEnterFrame's ikke-anonyme - altså definere dem specifikt som funktioner ud for scope. Derudover finder jeg det lidt skidt praksis at blande enterFrame's og intervaller, men det er måske bare en smagssag :)

Kommentar af lyden d. 03. Apr 2004 | 6

Super

Kommentar af per-olof d. 31. Aug 2004 | 7

Supergo Med lidt hjælp virker det og man ikke bruger progressive *.jpg

Kommentar af xxgullexx d. 13. Dec 2004 | 8

"Koden i denne guide vil kun virke i Flash MX 2003 og MX 2004 - ikke ældre versioner!"

Jeg giver "michael-k" ret i dette leder til misforståelser, som i mit tilfælde.

Evt. du kunne skrive dette i din synopsis.

Ellers et godt stykke arbejde, men jeg giver "Kun" middel da jeg ikke kunne bruge artiklen på mit software.