

## Knap med Action

Ind til nu har alt vi har lavet helt kørt af sig selv.

Det vi skal til nu er, at brugeren skal have indflydelse på hvad der skal ske – vi skal til at programmere noget interaktion.

### Opgave 12a

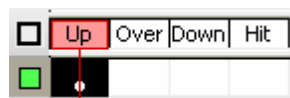
Opret en ny tegning og tegn noget der skal blive til en knap.

Konverter det tegnede til et symbol af typen button.

Test det, og konstater, at der kommer en hånd over knappen, som vi er vant til, men noget der har interaktion.

Gå ind i symbolet (dobbelklik).

Timeline ser nu helt anderledes ud.



Lav en Keyframe (F6) til Over og Down, så der kan redigeres separat i dem.

Rediger i knappen, og få den til at lave en endnu bedre knapreaktion, end bare at hånden kommer (farveskift, tekst flytte ramme ændres or whatever).

Test at det virker fornuftigt.

Hit-delen kan vi ikke teste endnu, da knappen ikke gør noget, og i mange tilfælde er den helt ligegyldig, de 3 første dele giver en rigtig god knap-funktion.

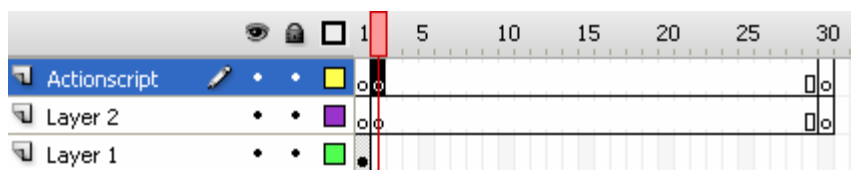
### Opgave 12b

Ude på scenen oprettes 2 nye lag.

Kald det ene for Actionscript, og

opret keyFrames, så man har følgende

timeline.



Tegn noget ind i Layer 2, der udfører en eller anden form for bevægelse – Brug noget tweening for at gøre det let for jer selv.

Test det.

Det er ikke de store geniale ting der kommer ud af det – det kommer vi til.

### Opgave 12c

Marker Actionscript-laget, stil dig i timeline frame 1 og vælg Actions nede i Properties-toolbaren (eller tryk F9)

skriv følgende linie ind:

```
stop();
```

Test hvordan det nu virker (Nu skulle den gerne vise knappen, men der sker ikke noget – heller ikke når man trykker på knappen).

Gå tilbage i Properties.

Marker knappen, og giv den et Instance Name minKnap.

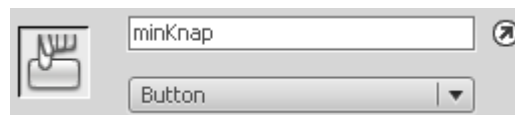
Oppe i Actionlaget tilføjes følgende (vær præcis med store og små bogstaver):

```
minKnap.addEventListener(MouseEvent.CLICK, knapClick);
```

```
function knapClick(event:MouseEvent):void {
    gotoAndPlay(2);
}
```

Test hvordan det virker nu.

Den skulle gerne gennemføre tingene, og komme tilbage til start.



## Opgave 12d

Nu vil vi gerne ændre lidt i funktionen, så vi ikke automatisk kommer tilbage og kan starte igen.

Det gør vi ved at skrive:

```
stop();
```

Det skal stå i Frame 30 i Actionscript-laget.

Test det.

Resultatet skulle gerne være, at filmen går i stå, og man kan ikke komme videre.

Lav en ny knap i Layer 1 Frame 30 (der skal lige oprettes keyFrames til at det kan lade sig gøre).

Giv knappen passende egenskaber, og giv den Instance Name minKnap2.

I Actionscript-laget programmeres nu følgende ind:

```
minKnap2.addEventListener(MouseEvent.CLICK, knapClick2 );  
  
function knapClick2(event:MouseEvent):void {  
    gotoAndStop(1);  
}
```

Test det.

Funktionen skulle nu gerne være som følger:

I starten har vi kun den første knap vist – ikke andet.

Den har fornuftig knapopførsel.

Når vi klikker på den, så forsvinder knappen, og bevægelsen udføres.

Når bevægelsen er slut, så stopper det der har bevæget sig, og der kommer en ny knap til syne.

Denne knap har også fornuftig knapopførsel.

Klikker vi på den knap, så kommer vi tilbage til udgangspunktet, med den første knap, og vi kan nu starte forfra.

## Opgave 12e

Prøv nu at foretage følgende rettelse på egen hånd.

Vi er fuldt tilfreds med tingen der kommer frem og bevæger sig – men:

når den anden knap kommer frem, så vil vi ikke have andet på scenen.

Når vi trykker på den anden knap skal den igen springe til den første knap, så vi kan starte forfra.

## Knap ud af Flash

Man kan tit få brug for at man i Flash ønsker at få en knap til at lave noget uden for Flash – typisk er det til at gå til et andet sted på Internettet, hvis man vil programmere en menu.

## Opgave 13a

Her vil vi ikke programmere en hel menu, men blot få en knap til at starte en side op.

Start en ny Flash-fil op (husk Actionscript 3).

Tegn en knap.

Konverter den til et symbol af typen Button.

Giv den fornuftig knap-reaktion.

Giv knappen instance-navnet googleKnap.

Lav et lag der hedder Actionscript.

Skriv følgende kode i Actions i Actionscript-laget:

```
googleKnap.addEventListener(MouseEvent.CLICK, internetKnap );  
  
function internetKnap(event:MouseEvent):void {  
    trace("Henvendelse til google !!");  
    var googleURL:URLRequest = new URLRequest("http://www.google.dk");  
    navigateToURL( googleURL );  
}
```

Test det.

Der skal gerne ske to ting, når man trykker på knappen:

Trace skriver i Output-vinduet – Henvendelse til google !!

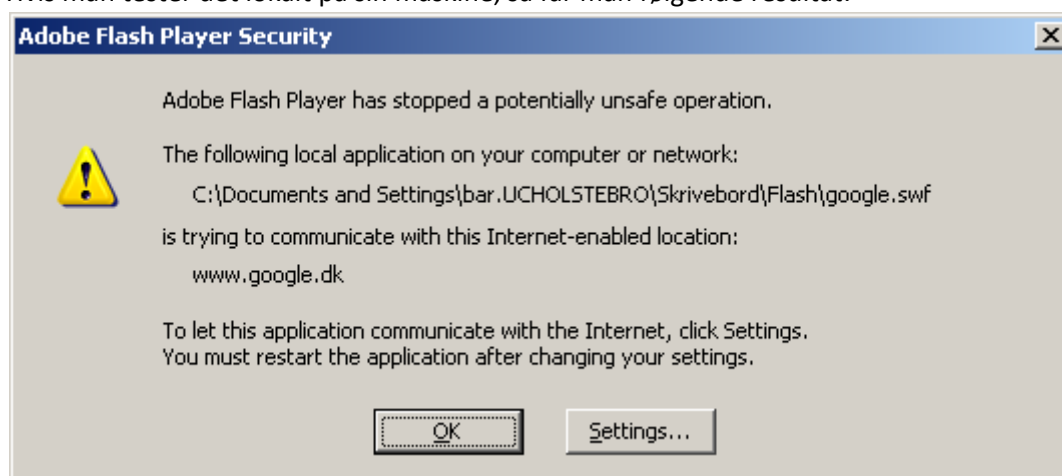
Der åbnes en browser med [www.google.dk](http://www.google.dk).

### Opgave 13b

Flash-filer er lavet til at fungere på Internettet, og derfor er det let at lave en side, hvor Flashresultatet ligger integreret. Husk at gemme flashfilen et sted hvor du kan finde den. Den funktion man skal bruge er Publish i File.

Den laver dels en .swf fil, en også en .html fil, der kan lægges direkte på nettet.

Hvis man tester det lokalt på sin maskine, så får man følgende resultat:



Grunden er at man har en sikkerhedsmæssig risiko, når man afvikler Flash lokalt på sin maskine.

Det skal afvikles på en internet-server.

Prøv at placere tingene på en server, og teste det via en browser.

På Holstebro HTX kan I placere det på jeres adgang til HTX-serveren, hvor I kan lægge det så det kun kan ses lokalt, eller I kan vælge at lægge det så det kan ses offentligt på Internettet – Se beskrivelsen af HTX-serveren – ligger kun lokalt på skolen.

Der skal 2 filer med:

Jeres .swf-fil (det er den oversatte .fla fil). Den .html fil som Adobe Flash har lavet.

Test at det virker nu.

### Opgave 13c

Som koden er i opgave 13b vil den åbne google i et nyt vindue.

Hvis man vil ændre det, så skal der foretages en lille rettelser:

```
navigateToURL( googleURL, "_self" );
```

Test det (igen på en server).