

Specielle effekter med musen

Musen er en god måde at få interaktion med Flash på, da man kan lave mange gode ting med musen ved at bruge Actionscript.

Opgave 16a

Tegn en flot baggrund – landskab.

Tegn et sigtekorn til musen, lav det til et Movie Clip og giv det et navn.

Placer det uden for scenen.

Lav et Actionscript lag.

Skriv følgende kode ind i Action:

```
stage.addEventListener(MouseEvent.MOUSE_MOVE , flytCursor);

function flytCursor(event:MouseEvent):void {
    sigteKorn.x = mouseX;
    sigteKorn.y = mouseY;
    if ((mouseX < 10) || (mouseX > 540) || (mouseY < 10) || (mouseY > 390)) {
        Mouse.show();
        sigteKorn.visible = false;
    } else {
        Mouse.hide();
        sigteKorn.visible = true;
    }
    event.updateAfterEvent();
}
```

Opgave 16b

Tegn et skudhul og konverter det til et symbol af typen Movie Clip.

I biblioteket sættes Property egenskaberne til at eksportere til Actionscript med navnet skudHul.

Tilføj så følgende kode i actionscript-laget:

```
stage.addEventListener(MouseEvent.MOUSE_DOWN , skud);

function skud(event:MouseEvent):void {
    var hul:MovieClip = new skudHul();
    hul.x = mouseX;
    hul.y = mouseY;
    stage.addChild(hul);
}
```

Når man skyder, så sætter skudhullet sig et forkert sted i forhold til sigtekornet.

Prøv at korrigere i koden eller i tegningen, så skudhullet rammer rigtigt.

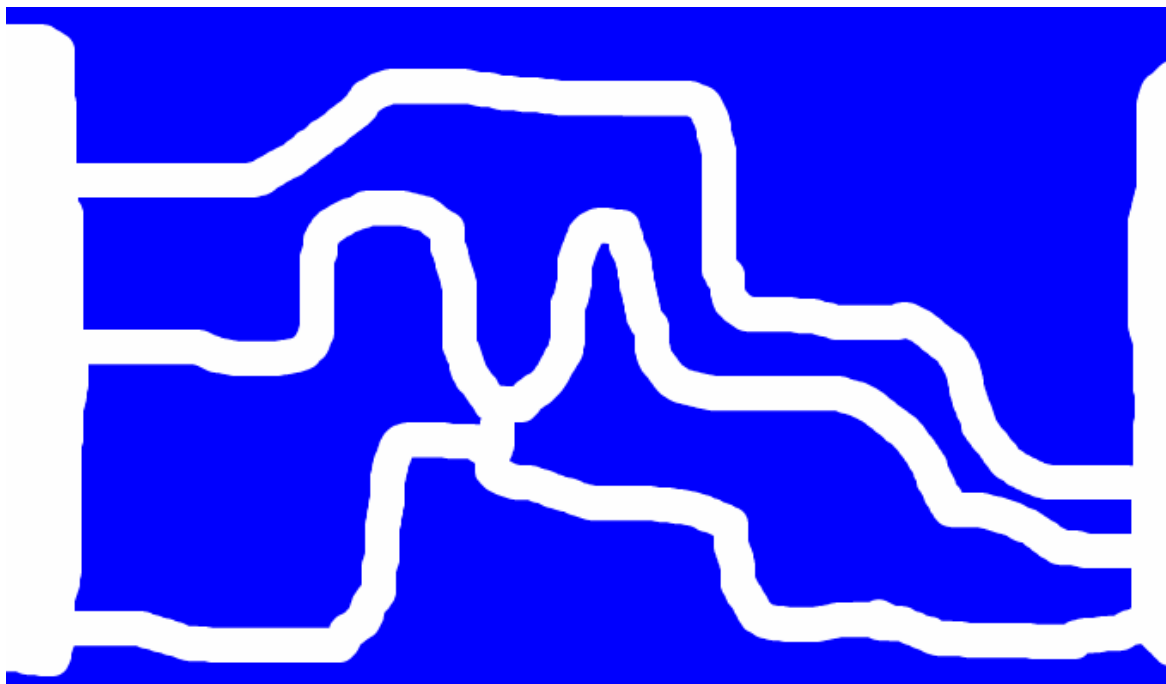
Opgave 17a

En ting der er ret populært at lave i Flash er små spil. Jeg vil her demonstrere udviklingen af et lille spil, der måske ikke er særligt ophidsende for unge i gymnasiet, men illustrer nogle principper i Actionscript, og det kan faktisk være et ganske godt spil til yngre målgrupper.

Ideen er at vi kan komme til at samle 3 ting op, og føre dem gennem labyrinten.

Opbyg en bane ved at sætte en kasse ind og få den til at have nogenlunde dette udseende:

Det er vigtigt at det hvide (eller hvilken baggrund I har) laves med viskelæderet.



Lav banen til et Movie Clip, og giv den Instance Name bane.

Tegn et lille kvadrat på 10x10 pixels, placeret i start-feltet.

Lav det til et Movie Clip ved navn plet.

Giv den aktuelle plet Instance Name plet1.

Lav et lag der hedder Actionscript.

Skriv følgende kode ind i laget:

```
stop();
var flyt1:Boolean = false; // Angiver om plet1 er ved at flytte
// Tilføj det der skal ske med musen
addEventListener(MouseEvent.CLICK, samleOp);
addEventListener(MouseEvent.CLICK, slip);
addEventListener(MouseEvent.CLICK, flyt);

function samleOp(event:MouseEvent):void {
    if (plet1.hitTestPoint(mouseX, mouseY)) {
        flyt1 = true;
        Mouse.hide();
        plet1.startDrag();
    }
}

function slip(event:MouseEvent):void {
    if (flyt1) {
        flyt1 = false;
        Mouse.show();
        plet1.stopDrag();
    }
}

function flyt(event:MouseEvent):void {
}
```

Test at man kan samle pletten op, og flytte rundt på den, indtil nu helt frit.

Opgave 17b

Lav en ny keyFrame i Frame 2, både i Layer 1 og laget Actionsript.

I layer 1 laves der en afslutnings-skærm.

I laget Actionsript skrives følgende ind i Action:

```
stop();
```

I Frame 1 skrives der følgende ind i laget Actionsript i funktionen flyt:

```
if (flyt1) {
    if (testBane(plet1)) {
        slut();
        gotoAndPlay(2);
    }
}
```

Der tilføjes også to funktioner som følger:

```
function testBane(objekt:MovieClip) {
    if (bane.hitTestPoint(objekt.x, objekt.y, true)) {
        return true;
    }
    return false;
}

function slut():void {
    Mouse.show();
    if (flyt1) {
        plet1.stopDrag();
    }
    flyt1 = false;
    removeEventListener(MouseEvent.MOUSE_DOWN, samleOp);
    removeEventListener(MouseEvent.MOUSE_UP, slip);
    removeEventListener(MouseEvent.MOUSE_MOVE, flyt);
}
```

Test at det virker

Der er stadig den lille skønhedsfejl, at man kan godt trække pletten et stykke ud over banen i den ene retning, før man slutter.

Man er også nødt til at lukke ned, inden man kan komme til at spille igen.

Opgave 17c

Opret i Frame 2 et symbol af typen button, og giv den fornuftig knap-funktion.

Giv den et Instance Name knap.

I Frame 2 indsættes følgende kode

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

function knapClick(event:MouseEvent):void {
    knap.removeEventListener(MouseEvent.CLICK, knapClick );
    gotoAndPlay(1);
}
```

I Frame 1 indsættes følgende kode ude efter flyt1:

```
var pletX:Number = 10; // Angiver størrelsen af pletten
var pletY:Number = 10;
```

Inde i funktionen testBane indsættes følgende kode inden return;

```
if (bane.hitTestPoint(objekt.x + pletX, objekt.y, true)) {
    return true;
}
if (bane.hitTestPoint(objekt.x, objekt.y + pletY, true)) {
    return true;
}
if (bane.hitTestPoint(objekt.x + pletX, objekt.y + pletY, true)) {
    return true;
}
```

Opgave 17d

Vi vil nu have en plet mere ind på scenen, så vi kan trække begge pletter gennem banen.

Tag symbolet fra Library og placer det på Scenen.

Giv det Instance Name plet2.

I Actionscript-laget angives en ny flyt2 variabel:

```
var flyt2:Boolean = false; // Angiver om plet2 er ved at flytte
```

Inde i funktionen samleOp() tilføjes

```
if (plet2.hitTestPoint(mouseX, mouseY)) {
    flyt2 = true;
    Mouse.hide();
    plet2.startDrag();
}
```

Prøv selv at regne ud hvad der skal tilføjes i funktionen slip().

I funktionen slut() tilføjes der:

```
if (flyt2) {
    plet2.stopDrag();
}
flyt2 = false;
```

Endelig tilføjes der i funktionen flyt() følgende:

```
if (flyt2) {
    if (testBane(plet2)) {
        slut();
        gotoAndPlay(2);
    }
}
```

Test nu koden, og se at begge pletter kan lave fejl.

Tilføj på samme måde en plet3.

Test igen.

Opgave 17e

Ret farven på de to af pletterne – brug Color effect – Tint ellers ændres alle på en gang, og vælg den farve de skal have – husk at skrue op for farven.

Tegn en mål-kurv på den modsatte side af banen.

Konverter til Movie Clip og giv den Instance Name kurven.

Inde i funktionen slip() tilføjes følgende:

```
if (plet1.hitTestObject(kurven)) {
    this.removeChild(plet1);
}
```

Således at der står følgende if-sætning i slip():

```
if (flyt1) {
    flyt1 = false;
    Mouse.show();
    plet1.stopDrag();
    if (plet1.hitTestObject(kurven)) {
        this.removeChild(plet1);
    }
}
```

Der programmeres det samme for plet 2 og plet 3

Test at man kan putte alle 3 pletter i kurven, og at de forsvinder.

Opgave 17f

Vi vil nu gøre så spillet kan slutte med success. Det gør vi ved at tælle hvor mange pletter vi har puttet i kurven.

Vi opretter en variabel til at tælle i:

```
var hits:Number = 0; // Antallet af pletter det er kommet i kurven
```

Når vi så får en plet i kurven, så registreres det inde i funktionen slip() ved at tælle hits frem:

```
if (plet1.hitTestObject(kurven)) {  
    hits += 1;  
}
```

Det skal selvfølgelig gøres på alle 3 pletter.

Inden funktionen slip() afsluttes tjekkes der om vi har vundet:

```
if (hits == 3) {  
    slut();  
    gotoAndPlay(3);  
}
```

Som det kan ses i tjekket, så sender vi visningen til Frame 3, så der skal der laves en tillykkeskærm, og der skal opbygges en knap med en funktion, så man kan starte igen. Det gøres på samme måde som med slutskærmen, bare lidt mere festligt.

Opgave 17g

For at sikre at brugeren ikke bare vælger den letteste ved igennem, men at hver plet kommer gennem den rigtige bane, så laver vi lidt snyde-tjek inde i banen.

Opret en cirkel og konverter den til et symbol tjek af typen Movie Clip. Det skal være samme farve som baggrunden, så det kan gemmes inde i banen. Lav 3 objekter af symbolet og giv dem instans navnene tjek1, tjek2 og tjek3, og placer dem i hver sin bane.

Opret 3 variabler som følger:

```
var tjekket1:Boolean = false; // Angiver om plet1 er kommet forbi tjek1  
var tjekket2:Boolean = false; // Angiver om plet2 er kommet forbi tjek2  
var tjekket3:Boolean = false; // Angiver om plet3 er kommet forbi tjek3
```

Inde i funktionen flyt() tilføjes:

```
if (plet1.hitTestObject(tjek1)) {  
    tjekket1 = true;  
}
```

Og tilsvarende for plet2 og plet3.

Der rettes inde i funktionen slip(), der hvor der testes for om pletten slippes i kurven, så det kommer til at se ud som følger:

```
if (plet1.hitTestObject(kurven)) {  
    if (! tjekket1) {  
        return;  
    }  
    hits += 1;  
    this.removeChild(plet1);  
}
```

Der rettes tilsvarende til plet2 og plet3.

Test det.

Som en sidste ting lægger vi en tid på spillet, fra den første plet er taget, til den sidste er kommet i kurven.

Placer en Dynamic Text et sted på scenen, hvor den kan udvide sig, og skriv 0 i den.

Giv den et instance name tid.

Lav 2 nye variabler som følger:

```
var startet:Boolean = false; // Om tiden er begyndt at tælle  
var ticks:Number = 0; // Tiden der er gået i antal Frames
```

Inde i funktionen samleOp() sættes startet = true; når vi samler en af pletterne op (den skal ind 3 gange).

```
startet = true;
```

Endelig tilføjes funktionen der får tiden til at gå:

```
addEventListener(Event.ENTER_FRAME, tidsTick);  
  
function tidsTick(event:Event):void {  
    if (! startet) return;  
    ticks += 1;  
    if ((ticks % 24) != 0) return;  
    tid.text = (ticks / 24).toString();  
}
```

Test at det virker.

Den laver en fejl, fordi den ikke kan se eventListeneren til tid, når den er stoppet.

Find ud af at fjerne den igen ned i slut().

Opgave 18a

Lav et puslespil med 9 brikker, der kan løses ved at trække brikkerne på plads.

Tjaa – den er endnu kun en ide – den kan I selv tumle med.