SVG

Denne artikel skal vise noget af det uforløste potentiale der ligger i teknologien SVG. Indtil for nylig var IE ikke interesseret i SVG og derfor skulle man veksle mellem IE's VML og SVG som resten (de øvrige browser agenter) har forpligtet sig til.
Men

Skrevet den **15. Nov 2010** af **mike1963** I kategorien **Design / Generelt** | ★★☆☆☆

Jeg vil tage udgangspunkt i en klassisk urskive med 12 time markeringer og minut markeringer mellem disse. Det giver 60 punkter og jeg anvender cos() og sin() der er kalibreret til 360 grader.
Kalibreringen af canvasen er 2000 x 2000 punkter, med en width på 100%. På denne måde vil uret's reelle størrelse variere alt efter det absolute mål.
Jeg sætter skivens centrum i 1000,1000 da vi har brug et Cartetisk koordinat system, hvor cos() og sin() udgår fra centrum. Istedet for 0,0 bruger vi altså 1000,1000, så vi bare skal lave funktionen til: 1000-cos(radian)*viserlængde

Skabelon
Jeg bruger et stylesheet (XSLt) til at ligge objekterne ud på canvas'en:

line (Visere og Markeringer)
circle (Urskive og centrum)
marker (selve spyddet på viserarmen)
text (tekst)

Når prototypen er på plads, er fase I på plads og fase II kan testes.
[code]
msxsl degrees.xml clockSetup.xsl -o clock.XML
[/code]

Som det ses bruger jeg msxsl, Andre parsere kan også benyttes, da det handler om standarder fra W3C. clockSetup.xml er en driver kilde, der skal få XSLt til at gøre det tunge beregningsarbejde. degrees.xml indeholder lidt noder til at kalibrere urskiven med. Her er en kvantificering af markeringerne og en underinddeling som blot er nogle dummy noder.
clock.xml er selve resultatet.

Animering
Animeringsdelen - altså den mekanisme, der får urviserne til at bevæge sig - kaldes animering i det følgende.
Ved hjælp af event'en onload kan vi bringe animeringslogikken i spil i det øjeblik at dokumentet er load'et
Jeg monterer ID's på de tre visere, så jeg kan tilgå dem via DOM API'et:
[code]
secHand = evt.target.ownerDocument.getElementById("secHand");
[/code]

Basis
Ved at bruge setInterval i javascript, kan jeg få genereret en tidsevent, der uendeligt trigger moveHands() funktionen. Jeg har valgt at bruge 1000 millisekunder som "heartbeat" interval.
[code]

setInterval("moveHands()", 1000);
[/code]
Det eneste jeg så skal er at flytte koordinaterne x og y, hver gang jeg får kontrol. Sekund værdien mellem 0 - 59 benyttes direkte til at beregne x,y
[code]
function moveHands() {
var now = new Date();
var secHandPosition = (now.getSeconds() * 6) + 90;
...
[/code]
Den endelige beregning ser sådan ud:
[code]
secHand.setAttribute("x1", 1000 - Math.cos(secHandPosition * Math.PI/180) * 380);
[/code]

De 380 er længden på viseren. Der monteres en marker på dette punkt så længden øges tilsvarende Minut-viseren trækkes med for hver gang sekund = 0. Samme metode. Derimod er timeviseren lidt anderledes i adfærd. Den skal i praksis bevæge sig 1/60 del af en time som jo er en 1/12 del af urskiven - altså 1/720 punkt hver gang sekund = 0
[code]
if (secHandPosition == 90) {
  var minHandPosition = (now.getMinutes() * 6) + 90;
  var hrHandPosition = (now.getHours()   * 30) + (now.getMinutes() / 2) + 90 ;
[/code]
De 90 skal tillægges, da 0 punktet skal forskydes en kvart omgang så nulpunktet er øverst i urskiven.

Endelig logik
[code]
var factor = 1000, secHand = '', minHand = '', hrHand = '';
function beginMotion(evt) {
  secHand = evt.target.ownerDocument.getElementById("secHand");
  minHand = evt.target.ownerDocument.getElementById("minHand");
  hrHand = evt.target.ownerDocument.getElementById("hrHand");
  setInterval("moveHands()", 1000);
}
function moveHands() {
  var now = new Date();
  var secHandPosition = (now.getSeconds() * 6) + 90;

  secHand.setAttribute("x1", factor - Math.cos(secHandPosition * Math.PI/180) * 380);
  secHand.setAttribute("y1", factor - Math.sin(secHandPosition * Math.PI/180) * 380);

  if (secHandPosition == 90) {
    var minHandPosition = (now.getMinutes() * 6) + 90;
    var hrHandPosition = (now.getHours()   * 30) + (now.getMinutes() / 2) + 90 ;
    minHand.setAttribute("x1", factor - Math.cos(minHandPosition * Math.PI/180) * 360);
    minHand.setAttribute("y1", factor - Math.sin(minHandPosition * Math.PI/180) * 360);
    hrHand.setAttribute("x1", factor - Math.cos(hrHandPosition * Math.PI/180) * 300);
    hrHand.setAttribute("y1", factor - Math.sin(hrHandPosition * Math.PI/180) * 300);
  }
}
[/code]
kilde: http://xmlsoap.dk/clockSVG_byMikeMainFrame1963.zip
uret: http://xmlsoap.dk/clock.xml
artikel: http://xmlsoap.dk/docClock.xml