

IntMu.Lab4

0. Requisitos

A realização deste trabalho exige a utilização dos seguintes componentes:

- Computador com ambiente gráfico:
MS-Windows, MAC-OS ou GNU/Linux
- Visualizador de SVG, preferencialmente Google Chrome.
- Editor de texto
- Acesso HTTP e SSH à Internet e ao servidor ave.dee.isep.ipp.pt.

1. Relógio SVG

Prepare um gráfico SVG (clock.svg) definindo uma zona de desenho quadrada com 600 unidades de medida de lado.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 600 600" >
</svg>
```

1.1. Fundo

No centro da área de desenho, coloque um círculo com raio igual a 240 unidades.

```
<g transform="translate(300,280)">
  <circle cx="0" cy="0" r="240" stroke="black" stroke-width="10" fill="#ffffd0"/>
</g>
```

Experimente a visualização deste gráfico no *browser* **Google Chrome**. Verifique o seu comportamento quando altera a geometria da janela do visualizador.

Sobre este círculo vai ser reproduzido um tradicional relógio analógico de ponteiros. Dentro do círculo, coloque uma legenda identificadora do desenho. Utilize o seu nome e número de aluno para identificar o seu relógio.

```
<g id="labels" text-anchor="middle" >
  <text x="0" y="-130"> IntMu </text>
  <text x="0" y="-110"> 2020 </text>
</g>
```

Afine as propriedades visuais da legenda com um bloco de estilo CSS.

```
<style type="text/css">
  #Numbers {
    font-family: "Sans";
    font-size: 30px;
  }
  #labels { font-family: "Sans"; font-size: 20px; fill: blue; }
  #city { font-size: 28px; }
</style>
```

1.2 Marcações

Para representar as 12 marcações correspondentes às horas, crie a definição de uma marcação na posição das 12 horas.

```
<defs>
  <g id="mark">
    <rect y="-230" x="-4" width="8" height="16" fill="brown" />
  </g>
</defs>
```

Utilize esta definição para criar as 12 marcações das horas.

```
<g id="Marcacoes">
  <use xlink:href="#mark" transform="rotate( 0)" />
  <use xlink:href="#mark" transform="rotate( 30)" />
  <use xlink:href="#mark" transform="rotate( 60)" />
  <use xlink:href="#mark" transform="rotate( 90)" />
  <use xlink:href="#mark" transform="rotate(120)" />
  <use xlink:href="#mark" transform="rotate(150)" />
  <use xlink:href="#mark" transform="rotate(180)" />
  <use xlink:href="#mark" transform="rotate(210)" />
  <use xlink:href="#mark" transform="rotate(240)" />
  <use xlink:href="#mark" transform="rotate(270)" />
  <use xlink:href="#mark" transform="rotate(300)" />
  <use xlink:href="#mark" transform="rotate(330)" />
</g>
```

Para que seja possível referenciar objetos através do atributo **xlink**, deve incluir a declaração do *namespace* respectivo. Para isso, acrescente ao elemento **svg** envolvente o seguinte atributo **xmlns**:

```
xmlns:xlink="http://www.w3.org/1999/xlink"
```

Verifique que as marcações das horas são reproduzidas na visualização.

Acrescente à sua definição da marcação as 4 marcas correspondentes aos minutos até à próxima marca de hora.

```
<defs>
  <circle id="sub_mark" cx="-227" cy="0" r="3" fill="#444444" />
  <g id="mark">
    <rect y="-230" x="-4" width="8" height="16" fill="brown" />
    <use xlink:href="#sub_mark" transform="rotate( 6)" />
    <use xlink:href="#sub_mark" transform="rotate(12)" />
    <use xlink:href="#sub_mark" transform="rotate(18)" />
    <use xlink:href="#sub_mark" transform="rotate(24)" />
  </g>
</defs>
```

Confirme a funcionalidade da alteração.

1.3 Ponteiros

Na secção de definições (DEFS), defina os 3 ponteiros para as horas, os minutos e os segundos.

```

<rect id="ponteiro_h" y="-120" x="-6" width="12" height="160" />
<rect id="ponteiro_m" y="-200" x="-4" width="8" height="240" />
<rect id="ponteiro_s" y="-220" x="-3" width="6" height="280" fill="#d80000" />

```

Utilize estas definições para criar os ponteiros do relógio.

```

<use xlink:href="#ponteiro_h">          </use>
<use xlink:href="#ponteiro_m">          </use>
<use xlink:href="#ponteiro_s">          </use>

```

1.4 Animação

Aplique uma animação ao ponteiro dos segundos para que este adquira o movimento de rotação normal com uma volta por minuto.

```

<use xlink:href="#ponteiro_s">
  <animateTransform attributeName="transform" begin="0s" dur="60s"
    type="rotate" from="0" to="360" repeatCount="indefinite" />
</use>

```

Aplique também as animações correspondentes aos ponteiros das horas e dos minutos.

1.5 Hora real

Para acertar o relógio SVG em função do relógio do computador, crie um função *javascript* (**update_clock**). Utilize a função **Date()** para obter a hora atual. Aplique o método **setAttribute** para definir o atributo **transform** do ponteiro dos segundos com uma rotação proporcional ao número de segundos da hora atual.

```

<script>
function update_clock() {
  var t = new Date();
  var seg = t.getSeconds();
  var asec = 6 * seg;
  document.getElementById('ponteiro_s').setAttribute('transform','rotate('+asec+')');
}
</script>

```

Adicione um atributo **onload** ao elemento SVG envolvente para que a função **update_clock()** seja chamada sempre que o desenho é carregado.

```
onload="update_clock()"
```

Verifique que o ponteiro dos segundos é ajustado sempre que o ficheiro SVG é recarregado.

Acrescente também à função **update_clock()** as instruções necessárias ao ajuste dos ponteiros dos minutos e das horas.

```

var min = t.getMinutes();
var amin = 6 * min + 0.1 * seg;
document.getElementById('ponteiro_m').setAttribute('transform','rotate('+amin+')');

var hor = t.getHours();
var ahor = 30 * (hor % 12) + 0.5 * min + (0.5/60) * seg;
document.getElementById('ponteiro_h').setAttribute('transform','rotate('+ahor+')');

```

Verifique que todos os ponteiros são devidamente inicializados sempre que o ficheiro SVG é recarregado.

1.6. Atualização permanente

Para que o relógio seja mantido em andamento de forma síncrona com o relógio do computador, basta que a função **update_clock()** seja chamada periodicamente todos os segundos. Acrescente à função **update_clock()** a inicialização de um temporizador para que a função seja novamente chamada ao fim de 1000 ms.

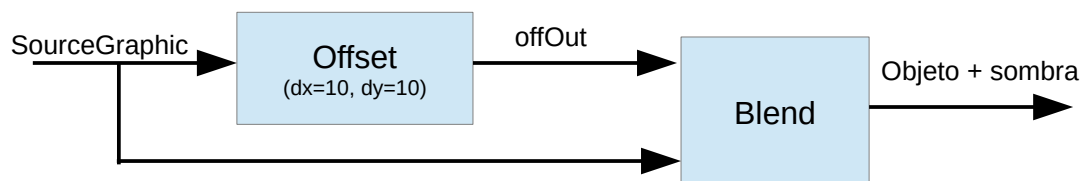
```
var mseg = t.getMilliseconds();  
window.setTimeout(update_clock, 1500-mseg)
```

Elimine as transformações de animação (**animateTransform**).

Verifique cuidadosamente o funcionamento do conjunto.

1.7 Sombras

Para adicionar o efeito de sombra a um objeto, pode ser definido um filtro que crie uma cópia do objeto original (*SourceGraphic*) com um pequeno deslocamento (*Offset*) em *x* e *y* (*dx* e *dy*). O resultado (*offOut*) deste deslocamento é combinado (*Blend*) com o objeto original (*SourceGraphic*) resultando num conjunto objeto / sombra.



```
<filter id="sombra" x="0" y="0" width="400%" height="400%" >  
  <feOffset result="offOut" in="SourceGraphic" dx="10" dy="10" />  
  <feBlend in="SourceGraphic" in2="offOut" mode="normal" />  
</filter>
```

Agrupe os seus ponteiros num grupo e aplique este filtro ao conjunto.

```
<g id="ponteiros" filter="url(#sombra)" >  
  ...  
</g>
```

Verifique o efeito produzido.

Para que o efeito seja mais realista pode criar a sombra a partir apenas da informação de opacidade do objeto original (*SourceAlpha*). Desta forma será produzida uma sombra negra, independentemente da cor do objeto. Este método também permite evitar as sombras de objetos transparentes.

```
<feOffset result="offOut" in="SourceAlpha" dx="10" dy="10" />
```

Verifique novamente o efeito produzido.

Para simular sombras com penumbra, é possível aplicar um filtro de desfocagem (*Blur*) ao resultado do deslocamento (*offOut*). A operação final de combinação (*Blend*) deve agora utilizar como entrada a saída deste filtro (*blurOut*).

```

<filter id="sombra" x="0" y="0" width="400%" height="400%" >
  <feOffset result="offOut" in="SourceAlpha" dx="10" dy="10" />
  <feGaussianBlur result="blurOut" in="offOut" stdDeviation="3" />
  <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
</filter>

```

Verifique o resultado. Confirme que a sombra é sempre produzida corretamente na direção da luz incidente.

Crie outro filtro semelhante mas com um vetor de deslocamento mais reduzido, para produzir um efeito de relevo nas marcações e na legenda.

```

<filter id="relevo" x="0" y="0" width="200%" height="200%" >
  <feOffset result="offOut" in="SourceAlpha" dx="0.7" dy="0.7" />
  <feGaussianBlur result="blurOut" in="offOut" stdDeviation="1" />
  <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
</filter>

```

Aplique este novo filtro às marcações e ao texto de legenda.

```

<g id="Marcacoes" filter="url(#relevo)">

```

Verifique o resultado.

1.7 Numeração

Adicione os números de 1 a 12 correspondentes às horas.

```

<g id="Numbers" text-anchor="middle" transform="translate( 0,12)">
  <text transform="rotate( 0) translate(0,-195) rotate( -0)">12</text>
  <text transform="rotate( 30) translate(0,-195) rotate( -30)"> 1</text>
  ...
</g>

```

Verifique o resultado.

2. Fusos Horários

2.1. Declare uma variável *javascript* para armazenar o fuso horário ativo. Inicialize esta variável com 0;

```

var tz = 0;

```

2.2. Acrescente uma etiqueta de texto para indicar o nome do fuso horário utilizado.

```

<text id="label" x="0" y="290"> London </text>

```

2.3. Na atualização do relógio, adicione à hora atual (t) a diferença devida ao fuso horário pretendido, em milissegundos:

```

var t = new Date();
t.setTime( t.getTime() + (tz*60*60*1000) ); // uma hora por fuso

```

2.4. Aproveite também a atualização do relógio para definir o fuso horário a partir de atributos que possam ser especificados na inclusão do relógio.

```

var nn = window.frameElement;
if ( nn ) {

```

```

        if ( nn.hasAttribute('city') )
            document.getElementById('label').innerHTML = nn.getAttribute('city');
        if ( nn.hasAttribute('tz') )
            tz = nn.getAttribute('tz');
    }

```

3. SVG em HTML

~~3.1. Instale o seu relógio SVG num diretório **intmu/lab4** na sua área Web do servidor **ave**. Verifique que o relógio ficou acessível através do URL:~~

~~—— http://ave.dee.isep.ipp.pt/~nr_aluno/intmu/lab4/clock.svg ——~~

3.2. No mesmo diretório, crie uma página HTML (wclock.html) que inclua o seu relógio SVG.

```

<!DOCTYPE html>
<html>
  <head>
    <title>IntMu Clock</title>
  </head>
  <body>
    <h1>IntMu Clock</h1>
    <div id='main' >
      <object data="clock.svg" type="image/svg+xml"> </object>
    </div>
    <div id='footer' >done by Lab_svg@IntMu</div>
  </body>
</html>

```

Verifique a funcionalidade da página com o *browser*.

3.3. Acrescente à sua página uma folha de estilos CSS.

```

<style>
  #map { float: right; width: 400px; }
  .center, #main, h1, #footer {
    margin:auto;
    width: 820px;
    text-align: center;
    overflow: auto;
  }
  #clock0 { width: 400px; float: left; }
  #clock1, #clock2, #clock3, #clock4 { width: 200px; }
</style>

```

3.4. Verifique que pode manipular o fuso horário através da especificação dos atributos **tz** e **city** do elemento **object** que inclui o relógio SVG.

```

tz='-4'  city='New York'

```

~~3.5. Acrescente à sua página versões reduzidas dos relógios de alguns dos seus colegas~~

```
<div class='center'>  
  <object id="clock1" data="/-aaaa/intmu/lab4/clock.svg" type="image/svg+xml"> </object>  
  <object id="clock2" data="/-bbbb/intmu/lab4/clock.svg" type="image/svg+xml"> </object>  
  <object id="clock1" data="/-cccc/intmu/lab4/clock.svg" type="image/svg+xml"> </object>  
  <object id="clock2" data="/-dddd/intmu/lab4/clock.svg" type="image/svg+xml"> </object>  
</div>
```

~~Configure estas versões reduzidas para diferentes fusos horários.~~

4. Mapa interativo

Importe o mapa SVG da Europa disponível em ave.dee.isep.ipp.pt/~jml/intmu/lab4.

```
wget http://ave.dee.isep.ipp.pt/~jml/intmu/lab4/map.svg
```

4.1 Inclua este mapa no bloco principal (**main**) da sua página HTML:

```
<object id="map" data="map.svg" type="image/svg+xml"> </object>
```

4.2. Inclua no mapa em formato SVG, uma folha de estilos CSS para configurar o aspeto gráfico dos elementos incluídos.

```
<style type="text/css">  
  <![CDATA[  
    .europe {  
      transition: fill .1s ease-out;  
      fill-opacity:1;  
      fill:navy;  
      stroke:#ffffff;  
      stroke-width:0.8;  
      display:inline;  
    }  
    .europe:hover {  
      fill: cyan;  
      stroke: yellow;  
      stroke-width:1.8;  
    }  
  ]]>  
</style>
```

4.3. Crie uma tabela com a especificação dos fusos horários de cada uma das regiões do mapa.

```
var tz_table = new Object();  
tz_table.pt = [0, "Lisbon"];  
tz_table.es = [1, "Madrid"];  
tz_table.fr = [1, "Paris"];  
tz_table.gb = [0, "London"];  
tz_table.it = [1, "Rome"];  
tz_table.de = [1, "Berlin"];  
tz_table.fi = [2, "Helsinki"];  
tz_table.gr = [2, "Athens"];  
tz_table.ru = [3, "Moscow"];  
tz_table.tr = [3, "Istanbul"];  
tz_table.ka = [2, "Kaliningrad"];
```

4.3. Adicione uma função para controlar o fuso horário utilizado pelo relógio principal (**clock0**) da sua página:

```
function mapClick(ev) {
    var clock = parent.document.getElementById('clock0');
    if ( clock ) {
        var tzele = tz_table[ev.target.id];
        if ( tzele ) {
            clock.setAttribute('tz',tzele[0]);
            clock.setAttribute('city',tzele[1]);
        }
    }
}
```

4.4. Registe esta função para atendimento do evento de *click* dos diversos países do mapa.

```
function installHandlers() {
    var cmap = document.getElementById('countries').children;
    var i;
    for ( i=0 ; i<cmap.length ; i++) {
        var id = cmap[i].id;
        if ( id.length == 2 ) {
            cmap[i].className += " active";
            cmap[i]['onclick'] = mapClick;
        }
    }
}
```

4.5. Execute esta função de instalação durante a inicialização do mapa, através do atributo *onload* do elemento SVG.

```
onload = "installHandlers();"
```

Verifique a funcionalidade da sua página com o *browser*.

5. Entrega

5.1.

~~Confirme que os seus ficheiros **map.svg**, **clock.svg** e **map.html** se encontram disponíveis na sua área Web e acessíveis no diretório~~

~~—— http://ave.dee.isep.ipp.pt/~nr_aluno/intmu/lab4/~~

~~Mantenha estes ficheiros acessíveis até ao fim do mês de junho. A data da alteração mais recente destes ficheiros será utilizada como data de entrega.~~

Envie os ficheiros **map.svg**, **clock.svg** e **map.html** por mail para jml@isep.ipp.pt, utilizando o assunto **INTMU:2021:LAB4:nr_aluno**

5.2.

Responda e submeta o questionário Lab4 disponível na área INTMU do Moodle.