

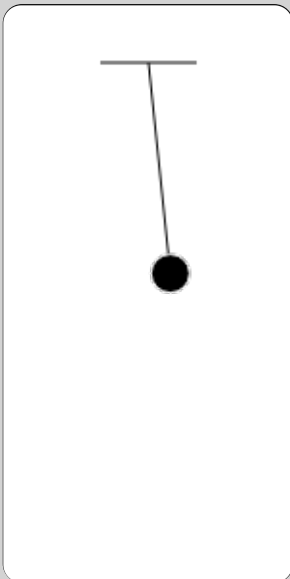
Apêndice C: o pêndulo simples

O exemplo a seguir mostra como produzir uma questão de múltipla escolha para a qual o usuário deve (em princípio!) escolher uma resposta a partir de informações que obtém observando uma animação de um pêndulo simples.

PÊNDULO SIMPLES

Qual o período do pêndulo?

a. 1.6 s
b. 3.2 s
c. 2.1 s



O documento está essencialmente dividido em duas partes. A primeira é uma grande divisão que abarca uma tabela com as opções para o usuário, os controles do cronômetro e o canvas com o pêndulo. A segunda contém o JavaScript que gerencia tudo. No quadro a seguir o conteúdo da tabela e do script foram eliminados para tentar auxiliar na visualização desses dois grandes blocos.

```
<div style="width:500px; margin:auto;  
border:1px solid black; border-radius:10px;  
background-color:lightgray;
```

```

padding-bottom:1em;
margin-bottom:2em">
<p style="width:50%; margin:auto; margin-top:1em;
border:1px solid black; border-radius:10px;
background-color:white; color:black;
text-align:center; padding:10px;
font-size:14pt;font-weight:bold">
PÊNDULO SIMPLES
</p>
<table style="width:80%; margin:auto; margin-top:1em">
...
</table>
</div>
<script>
...
</script>

```

Esse exemplo procura mostrar com um pouco mais de detalhes como as declarações de estilo podem ser utilizadas para obter-se o efeito desejado. Muitas são relativamente óbvias, mas não deixe de pesquisar na internet o significado delas (por exemplo, padding e margin).

A tabela é formada por uma linha com duas células. Na primeira célula está a questão com suas alternativas e os controles do cronômetro. Na segunda célula está o canvas com o pêndulo.

```

<table style="width:80%; margin:auto; margin-top:1em">
<tr>
<td style="text-align:left;vertical-align:top;width:50%">
Qual o período do pêndulo?
<ol style="list-style-type:lower-alpha">
<li>
<input type="radio" name="resp" id="r1" onChange="verifica(1)">
<span id="e1"></span> s
</li>
<li>
<input type="radio" name="resp" id="r2" onChange="verifica(2)">
<span id="e2"></span> s
</li>
<li>
<input type="radio" name="resp" id="r3" onChange="verifica(3)">
<span id="e3"></span> s
</li>
</ol>
<div style="width:100%; margin:auto; margin-top:6em;
text-align:center">
<div id="cronoOut"
style="width:75px; height:20px;
margin:auto; margin-bottom:1em;
text-align:center;
background-color:white;
border:1px solid black; border-radius:5px" >0.0</div>
<input type="button" value="LIGA" style="width:50px"
onClick="ligaPend()">

```

```

<input type="button" value="PARA" style="width:50px"
      onClick="paraPend()">
<input type="button" value="ZERA" style="width:50px"
      onClick="zeraPend()">
<div>
</td>

<td align="center">
<canvas id="cnv" width="150" height="300"
      style="background-color:white;
            border:1px solid black; border-radius:10px">
</canvas>
</td>

</tr>
</table>

```

Para a construção das alternativas foi utilizada uma lista ordenada (`ol`) indexada por caracteres alfabéticos minúsculos (`lower-alpha`). Cada item da lista (`li`) possui um elemento tipo `radio` com seu `id` único e um *container* do tipo `span`, também com seu `id`, que vai receber, dinamicamente, os valores das alternativas sorteadas aleatoriamente. Por conveniência, na mesma célula foram colocados uma divisão que vai receber o tempo marcado pelo cronômetro e os seus controles, que respondem de diferentes maneiras ao clique do mouse. Na célula seguinte está o `canvas`, com seu respectivo `id` e declaração de estilo (tamanho, fundo, borda).

O bloco com o script contém uma longa lista de declarações de variáveis com suas respectivas inicializações e as declarações das funções que vão responder ao pressionar de um botão e ao temporizador.

```

var xmin = -0.75, xmax = +0.75, ymin = 0, ymax = 3;
var cnv = document.getElementById("cnv");
var ctx = cnv.getContext("2d");
var cnvW = cnv.width; var cnvH = cnv.height;
var scx = cnvW/(xmax-xmin);
var scy = cnvH/(ymax-ymin);
var dx = cnvW/2;
var dy = cnvH;
ctx.setTransform(scx,0,0,-scy,dx,dy);
ctx.lineWidth = 1/scx;

var H = ymin + (ymax-ymin)*0.9; // altura do pto de fixacao
var lmin = 0.2; // tamanho minimo do pendulo
var lmax = 2.5; // tamanho maximo do pendulo
var L = lmin + (lmax-lmin)*Math.random(); // tamanho do pendulo
var g = 9.8; // aceleracao da gravidade
var T = 2*Math.PI*Math.sqrt(L/g); // periodo do pendulo

var alt = 1 + Math.floor(Math.random()*3); // alternativa correta
document.getElementById("e" + alt).innerHTML = T.toFixed(1);
for (var i=1;i<=3;i++) {
  if (i!=alt) {
    var x = T * 0.8*Math.random() + i;
    document.getElementById("e" + i).innerHTML = x.toFixed(1);
  }
}
var radios = document.querySelectorAll('input[type=radio]');
for (var i = 0; i < radios.length; i++) {
  radios[i].checked = false;
}

```

```

var inicioPend;
var cronoPend;
var t = 0, dt = 0.1;
var timerPend = setInterval("movePend()",100);

```

O código acima está dividido essencialmente em quatro blocos. No primeiro estão as instruções que viabilizam a transformação de coordenadas. No segundo, as definições das variáveis associadas ao movimento do pêndulo propriamente dito. O terceiro, bastante trabalhoso de interpretar, a definição de qual alternativa será a correta e a distribuição dos valores para as demais (e, adicionalmente, um conjunto de instruções que garante que nenhuma alternativa estará marcada quando a página for recarregada em alguns navegadores). O quarto (e final) bloco define as variáveis associadas ao tempo e inicializa o temporizador que movimenta o pêndulo.

```

function verifica(opt) {
    if (opt==alt) alert("Parabéns!");
    else alert("Tente novamente!");
}

function ligaPend() {
    if (cronoPend) clearInterval(cronoPend);
    inicioPend = new Date();
    cronoPend = setInterval("cronometro()",100);
}

function paraPend() {
    clearInterval(cronoPend)
}

function zeraPend() {
    document.getElementById("cronoOut").innerHTML = "0.0";
}

function cronometro() {
    var agoraPend = new Date();
    var tempoPend =
        (agoraPend.getTime() - inicioPend.getTime())/1000;
    document.getElementById("cronoOut").innerHTML =
        tempoPend.toFixed(1);
}

function movePend() {
    var theta = Math.PI/20 * Math.cos(2*Math.PI/T*t);
    var x = L * Math.sin(theta);
    var y = H - L * Math.cos(theta);
    t = t + dt;

    // limpa a tela
    ctx.clearRect(-0.75,0,1.5,3);

    // desenha o teto
    ctx.beginPath();
    ctx.moveTo(-0.25,H);
    ctx.lineTo(+0.25,H);
    ctx.strokeStyle = "black";
    ctx.stroke();

    // desenha o fio
    ctx.beginPath();
    ctx.moveTo(0,H);
    ctx.lineTo(x,y);
    ctx.strokeStyle = "black";

```

```
ctx.stroke();

// desenha a esfera
ctx.beginPath();
ctx.arc(x,y,0.1,0,2*Math.PI,0);
ctx.fillStyle = "black";
ctx.fill();
ctx.strokeStyle = "white";
ctx.stroke();
}
</script>
```

Fica ao leitor o desafio de estudar, pesquisar e entender os detalhes e juntar esses fragmentos em um documento que efetivamente funcione.

