Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Eduardo de Oliveira Silva
Fernando Maciel da Silva
Gleyson Israel Alves
Gusttavo Nunes Gomes
Ianka Talita Bastos de Assis

HTML5 Canvas e Javascript

Outubro
2017

Sumário

HTML5 Canvas e Javascript

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="canvasPosicao" width="578" height="200"></canvas>
    <!--limite de deteccao-->
    <script>
      function escreverMensagem(canvas, mensagem) {
        var context = canvas.getContext('2d');
        context.clearRect(0, 0, canvas.width, canvas.height);
        context.font = '18pt Calibri';
        context.fillStyle = 'black';
        context.fillText(mensagem, 10, 25);
      }
      function getPosicao(canvas, evt) {
        var rect = canvas.getBoundingClientRect();
        return {
          x: evt.clientX - rect.left,
          y: evt.clientY - rect.top
        };
      }
      var canvas = document.getElementById('canvasPosicao');
      var context = canvas.getContext('2d');

      canvas.addEventListener('mousemove', function(evt) {
        var mousePos = getPosicao(canvas, evt);
        var mensagem = 'Posicao do Mouse : x = ' + mousePos.x + '; y = ' + mousePos.y;
        escreverMensagem(canvas, mensagem);
      }, false);
    </script>
  </body>
</html>
```

recursos/codigos/posicao.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="canvasMove" width="578" height="200"></canvas>
    <script>
      window.requestAnimFrame = (function(callback) {
```

```
15          return window.requestAnimationFrame || window.
    webkitRequestAnimationFrame || window.mozRequestAnimationFrame
    || window.oRequestAnimationFrame || window.
    msRequestAnimationFrame ||
16          function(callback) {
17            window.setTimeout(callback, 1000 / 60);
18          };
19        })();
20
21        function desenhaRetangulo(retangulo, context) {
22          context.beginPath();
23          context.rect(retangulo.x, retangulo.y, retangulo.width,
    retangulo.height);
24          context.fillStyle = '#8ED6FF';
25          context.fill();
26          context.lineWidth = retangulo.borderWidth;
27          context.strokeStyle = 'black';
28          context.stroke();
29        }
30        function animate(retangulo, canvas, context, startTime) {
31          var time = (new Date()).getTime() - startTime;
32          var linearSpeed = 100;
33          var newX = linearSpeed * time / 1000;
34
35          if(newX < canvas.width - retangulo.width - retangulo.
    borderWidth / 2) {
36            retangulo.x = newX;
37          }
38
39          context.clearRect(0, 0, canvas.width, canvas.height);
40
41          desenhaRetangulo(retangulo, context);
42
43          requestAnimFrame(function() {
44            animate(retangulo, canvas, context, startTime);
45          });
46        }
47        var canvas = document.getElementById('canvasMove');
48        var context = canvas.getContext('2d');
49
50        var retangulo = {
51          x: 0,
52          y: 75,
53          width: 100,
54          height: 50,
55          borderWidth: 5
56        };
57
58        desenhaRetangulo(retangulo, context);
59
60        setTimeout(function() {
61          var startTime = (new Date()).getTime();
62          animate(retangulo, canvas, context, startTime);
63        }, 1000);
64      </script>
65    </body>
66  </html>
```

recursos/codigos/move_retangulo.html

```
1   <head>
```

```
    <style>
      body {
```

recursos/codigos/move_retangulo.html

Para criar uma animacao usando HTML5 Canvas, nós usamos o *requestAnimFrame* que habilita o navegador determinar o FPS adequado para cada animacao. Para cada frame de animação, nos atualizamos os elementos do canvas, limpando e redesenhando solicitando nova animação para dar aspecto de movimento.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="canvaClique" width="578" height="200"></canvas>
    <script>
      window.requestAnimFrame = (function(callback) {
        return window.requestAnimationFrame || window.
    webkitRequestAnimationFrame || window.mozRequestAnimationFrame
    || window.oRequestAnimationFrame || window.
    msRequestAnimationFrame ||
        function(callback) {
          window.setTimeout(callback, 1000 / 60);
        };
      })();

      function desenhaRetangulo(retangulo, context) {
        context.beginPath();
        context.rect(retangulo.x, retangulo.y, retangulo.width,
    retangulo.height);
        context.fillStyle = '#8ED6FF';
        context.fill();
        context.lineWidth = retangulo.borderWidth;
        context.strokeStyle = 'black';
        context.stroke();
      }
      function animate(lastTime, retangulo, runAnimation, canvas,
    context) {
        if(runAnimation.value) {
          var time = (new Date()).getTime();
          var timeDiff = time - lastTime;
          var linearSpeed = 100;
          var linearDistEachFrame = linearSpeed * timeDiff / 1000;
          var currentX = retangulo.x;

          if(currentX < canvas.width - retangulo.width - retangulo.
    borderWidth / 2) {
            var newX = currentX + linearDistEachFrame;
            retangulo.x = newX;
          }

          context.clearRect(0, 0, canvas.width, canvas.height);
          desenhaRetangulo(retangulo, context);
          requestAnimFrame(function() {
```

```
46              animate(time, retangulo, runAnimation, canvas, context)
    ;
47            });
48          }
49        }
50        var canvas = document.getElementById('canvaClique');
51        var context = canvas.getContext('2d');
52
53        var retangulo = {
54          x: 0,
55          y: 75,
56          width: 100,
57          height: 50,
58          borderWidth: 5
59        };
60
61
62        var runAnimation = {
63          value: false
64        };
65
66        document.getElementById('canvaClique').addEventListener('
    click', function() {
67          runAnimation.value = !runAnimation.value;
68
69          if(runAnimation.value) {
70            var date = new Date();
71            var time = date.getTime();
72            animate(time, retangulo, runAnimation, canvas, context);
73          }
74        });
75        desenhaRetangulo(retangulo, context);
76
77      </script>
78    </body>
79  </html>
```

recursos/codigos/clique.html

1 Referências Bibliográficas

GEARY,David. **Core HTML5 Canvas: Graphics, Animation, and Game Development**. 2012.