

# Desvendando o HTML5 para Games

Uma Abordagem Prática

2ª Edição, ampliada e revisada.

*Aprenda as técnicas para desenvolvimento  
de games com HTML5 e crie seus próprios  
jogos que rodam em qualquer navegador Web  
e dispositivo Mobile sem a necessidade de plugins!*

Rodrigo Cesar Vertulo



Rodrigo Cesar Vertulo

# **Desvendando o HTML5 para Games**

Uma Abordagem Prática

2ª Edição, ampliada e revisada.

Rodrigo Cesar Vertulo

**Desvendando o HTML5 para Games – Uma Abordagem Prática.**

**2.a Edição, ampliada e revisada.**

**Copyright(c) 2015 – Rodrigo Cesar Vertulo**

Todos os direitos reservados. Nenhuma parte deste material pode ser reproduzida ou transmitida sob qualquer forma ou por qualquer meio eletrônico ou mecânico, incluindo fotocópias, gravações, ou por qualquer meio de armazenamento de informações ou sistema de recuperação, sem a autorização expressa do autor deste livro.

Todas as marcas, logomarcas e ilustrações que eventualmente aparecem neste trabalho são de propriedade de seus respectivos detentores dos direitos autorais.

As informações apresentadas neste livro são fornecidas “como estão”, sem nenhuma garantia. Embora todas as precauções tenham sido tomadas na preparação deste trabalho o autor não se responsabiliza nem garante sua exatidão nem por perdas ou danos causados direta ou indiretamente a qualquer pessoa ou entidade pelo uso do conteúdo apresentado aqui.

Rodrigo Cesar Vertulo

*Dedico este livro aos meus amados pais, Sonia Regina de Araujo Vertulo e Antonio Carlos Vertulo e à minha amada irmã Renata de Araujo Vertulo, principais responsáveis por tudo o que sou. Amo vocês!*

*Também dedico esta obra à minha amada esposa, Michele Marinho Bonifácio Vertulo, que está sempre ao meu lado em todas as situações, apoiando-me, amando-me e construindo uma família e um futuro ao meu lado. Mi, amo você... somos somente um perante Deus!*

# Sumário

---

Introdução.....	9
Capítulo 1	
Conhecendo o HTML5.....	13
Capítulo 2	
Entendendo o que é um Canvas.....	17
Capítulo 3	
Desenhando círculos, quadrados e linhas.....	21
Capítulo 4	
Movimentando círculos, quadrados e linhas.....	31
Capítulo 5	
Renderizando bitmaps.....	37
Capítulo 6	
Reproduzindo sons.....	41
Capítulo 7	
Detectando eventos de teclado e mouse.....	53
Capítulo 8	
Controlando o movimento de elementos na tela.....	61
Capítulo 9	
Deteção de colisão entre elementos da tela.....	65
Capítulo 10	
Persistência de dados.....	77
Capítulo 11	
Geração de números aleatórios e suas aplicações.....	83
Capítulo 12	
Criação de games mobile em HTML5 .....	89
Capítulo 13	
Aprendendo fazendo.....	95
Capítulo 14	
Orientação a Objetos em Javascript.....	103
Capítulo 15	
Codificando o game 8Colors.....	109
Capítulo 16	
Código fonte completo.....	171

Capítulo 17	
Transformando um Web App em um App Nativo.....	185
Capítulo 18	
Conclusão.....	203
Anexo 1.....	204
Anexo 2.....	205
Anexo 3.....	206
Anexo 4.....	207
Anexo 5.....	209
Anexo 6.....	210
Anexo 7.....	211

# Sobre o Autor

---



**Rodrigo Cesar Vertulo** é graduado em Ciência da Computação pela Universidade Municipal de São Caetano do Sul e especialista em Engenharia de Software pela Universidade de Campinas. Foi pesquisador de Informática em Saúde pela Universidade Federal de São Paulo na linha de processamento de sinais biológicos, atuando principalmente com Avaliações Neuropsicológicas Computadorizadas. Foi professor universitário na

Fatec de São Caetano do Sul do Centro Paula Souza e atua como Engenheiro de Software na IBM do Brasil desenvolvendo e coordenando projetos de construção de softwares de grande complexidade. Atualmente está cursando Engenharia Eletrônica na Faculdade de Engenharia da Fundação Santo André. Tem experiência na área de Ciência da Computação com ênfase em Engenharia de Software, Análise e Projeto Orientado a Objetos, Processamento de Sinais Biológicos, Metodologias de Desenvolvimento de Sistemas, Business Intelligence, Informática em Saúde e Desenvolvimento de Games. Com formação técnica em Eletro-Eletrônica pela ETE Jorge Street do Centro Paula Souza também possui competência para o Desenvolvimento de Sistemas Embarcados Microprocessados e Microcontrolados, Projeto e Desenvolvimento de Dispositivos Eletrônicos e Mecatrônicos.

Rodrigo Cesar Vertulo



# Introdução

---

**C**ada vez mais o padrão HTML5 vem se popularizando na comunidade de desenvolvedores Web e em especial entre os desenvolvedores de Games. A principal vantagem na utilização deste novo padrão é a não necessidade de instalação de plugins (ex: Flash, Java) nos navegadores para que seja possível a execução de aplicações que fazem uso de recursos de áudio, vídeo e animações avançadas. A maioria dos navegadores Web já provêm um bom suporte a este padrão, de modo que os problemas de incompatibilidade com este ou aquele “browser” tornam-se cada vez menos frequentes. Além disso, a possibilidade de se desenvolver Web Apps para dispositivos móveis torna a utilização desse padrão ainda mais atraente. Este livro tem como objetivo desvendar os principais recursos do HTML5 voltados principalmente para os desenvolvedores de Games tanto para computadores Desktop, quanto para Smartphones e Tablets. Cada capítulo apresenta de forma didática um tópico sobre o tema, de modo que ao final da leitura deste livro será possível utilizar todo o conteúdo apresentado para a criação de um jogo completo e funcional. Durante esta nossa jornada você aprenderá os seguintes tópicos distribuídos nos capítulos destacados abaixo:

- **Capítulo 1:** Apresenta de forma sucinta a tecnologia HTML5 e os principais blocos estruturais de um game digital que servirão como “âncora” para a apresentação e desenvolvimento de todos os conceitos referentes ao HTML5 aplicado ao desenvolvimento de games
- **Capítulo 2:** Apresenta o principal elemento da tecnologia HTML5 que será utilizado durante todo o livro para o desenvolvimento dos exemplos e apresentação dos conceitos, o Canvas.
- **Capítulo 3:** Neste capítulo o leitor começará a criar elementos gráficos simples utilizando os recursos do HTML5 e Javascript que servirão como base para a aplicação dos conceitos mais complexos posteriormente.
- **Capítulo 4:** Introduz o leitor à técnica de animação de elementos gráficos em HTML5.
- **Capítulo 5:** Ensina o leitor a utilizar imagens no lugar de primitivas gráficas simples para criar games com alta qualidade visual.
- **Capítulo 6:** Apresenta as técnicas de reprodução de áudio em HTML5, descrevendo todas as especificidades dos principais navegadores Web relacionadas à reprodução de sons.
- **Capítulo 7:** Ensina o leitor a capturar eventos de teclado e mouse disparados pelos jogadores.

- **Capítulo 8:** Neste capítulo, uma extensão do anterior, o leitor aprenderá a controlar elementos visuais presentes no jogo a partir da captura de eventos de teclado e mouse.
- **Capítulo 9:** Aqui o leitor aprenderá a identificar o momento em que elementos do seu jogo colidem uns com os outros para, a partir da ocorrência da colisão, realizar tarefas específicas.
- **Capítulo 10:** Neste capítulo o leitor aprenderá uma das técnicas possíveis para armazenar dados localmente no computador dos jogadores de seu game.
- **Capítulo 11:** Apresenta alguns conceitos sobre jogabilidade e por que a geração de números aleatórios é importante neste quesito.
- **Capítulo 12:** Mostra uma visão geral sobre o desenvolvimento de games para dispositivos móveis e aborda aspectos relacionados à responsividade.
- **Capítulo 13:** Introduz o game a ser desenvolvido para dispositivos móveis utilizando os conceitos apresentados no livro.
- **Capítulo 14:** Apresenta uma visão geral sobre Orientação a Objetos com Javascript, importante para o desenvolvimento do game de exemplo.

- **Capítulo 15:** Descreve em detalhes toda a codificação do game de exemplo, direcionando o leitor em cada passo do processo de desenvolvimento.
- **Capítulo 16:** Mostra de forma integrada todo o código fonte do game desenvolvido ao longo do livro.
- **Capítulo 17:** Apresenta um passo a passo para transformar o game HTML5 desenvolvido nos capítulos anteriores em um aplicativo Nativo.
- **Capítulo 18:** Término do livro e considerações finais do autor.

## **Para quem este livro é destinado?**

Este livro é para qualquer pessoa interessada em desenvolver games digitais e aplicações multimídia utilizando a tecnologia HTML5, tanto para computadores Desktop, quanto para dispositivos móveis, e que possua conhecimentos básicos em alguma linguagem de programação e em tecnologias de desenvolvimento Web, em especial HTML, CSS e Javascript.

## **Contactando o Autor**

Será um prazer responder às suas perguntas e receber críticas e sugestões sobre este trabalho. Sinta-se a vontade para entrar em contato comigo, Rodrigo Cesar Vertulo, pelo e-mail [contato@wacawaca.com.br](mailto:contato@wacawaca.com.br) ou acessando meu site <http://www.wacawaca.com.br>

# Capítulo 1

## Conhecendo o HTML5

---

**O** HTML5 é uma linguagem para estruturar e apresentar conteúdo em navegadores Web, tais como Firefox, Chrome, Internet Explorer, entre outros. Ela é a quinta revisão da já famosa HTML (sigla para HyperText Markup Language) criada na década de 1990. Atualmente a especificação do padrão HTML5 encontra-se finalizada e ele apresentando recursos suficientes para o desenvolvimento de aplicações e games profissionais. O principal avanço propiciado pelo HTML5 é o suporte a elementos multimídia (sons, imagens e vídeos) de forma mais simples e compatível com diversos equipamentos (computadores, smartphones, tablets, etc.).

O desenvolvimento de games, seja em HTML5 ou em qualquer outra tecnologia, exige dos desenvolvedores a compreensão de alguns blocos fundamentais sobre os quais todos os jogos são criados. Trataremos a partir de agora sobre os aspectos tecnológicos da criação de um game e nos preocuparemos de forma superficial nos aspectos que tangem a disciplina de Game Design. Em outras palavras, falaremos de “bits e bytes” e assuntos ligados a enredo, roteiro, narrativas, etc. não serão abordados, a não ser quando forem

imprescindíveis para o entendimento dos tópicos que serão apresentados.

Na figura 1 podemos visualizar os quatro blocos principais sobre os quais um game é desenvolvido, são eles: Lógica, Gráficos, Sons e Eventos. Note que na figura aparece a linguagem Javascript se relacionando com cada um dos blocos. Isso ocorre por que o HTML5 sozinho não é capaz de fornecer todos os recursos necessários para o desenvolvimento de uma aplicação, inclusive um game. Vale lembrar que o HTML (tanto em sua quinta versão quanto nas outras) é apenas uma linguagem de marcação para a apresentação de conteúdos na Web e não uma linguagem de programação. Por outro lado, o Javascript é uma linguagem de programação e interage perfeitamente com documentos escritos em qualquer versão do HTML.

A seguir será apresentada uma breve descrição sobre cada um dos blocos constituintes de um game que serão detalhados ao longo deste livro. Cada um desses blocos pode ser visualizado na figura 1:

Anatomia de um Game

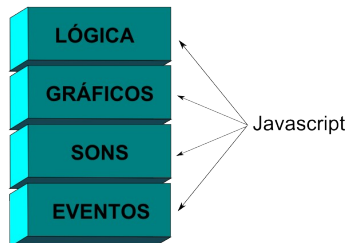


Figura 1 – Blocos principais de um projeto de game.

**Lógica:** Este bloco refere-se à codificação do comportamento do game, da forma como seus elementos (personagens, cenário, sons, gráficos) irão interagir. Por exemplo, é no bloco de lógica que podemos determinar que no momento que o tiro de um personagem atinge um outro personagem, este último deverá ser destruído e a pontuação do jogador deverá ser incrementada. Em outras palavras, é no bloco de lógica que definimos as regras e a mecânica do nosso game.

**Gráficos:** É perfeitamente possível desenvolver games que não se utilizam de gráficos. No passado, quando os computadores não possuíam tantos recursos como os atuais, muitos jogos totalmente baseados em textos foram desenvolvidos, especialmente os do gênero “Adventure”. Contudo, atualmente os jogadores consideram inaceitável um game que não forneça elementos gráficos, sejam em 2D ou 3D, fazendo com que a utilização dos mesmos seja obrigatória.

**Sons:** A mesma regra apresentada acima para os gráficos também aplica-se aos sons. Apesar de ser possível criarmos games sem o uso de recursos de áudio, atualmente ele é um item obrigatório. O som é um elemento muito importante em um game, pois ele, juntamente com os gráficos, é um dos grandes responsáveis pela imersão do jogador no “mundo virtual” do jogo.

**Eventos:** O bloco de eventos é o responsável por identificar as ações do jogador no “mundo real” transpondo-as para o “mundo virtual”. Em outras palavras, o bloco de eventos identifica os comandos do jogador, seja pelo mouse, joystick,

teclado, webcam , etc. e atualiza o “mundo virtual” de acordo com as regras e mecânica do game.

Neste capítulo foi dada uma breve introdução sobre o HTML5 sobre a anatomia de um game. Nos próximos capítulos trabalharemos em cada um dos blocos apresentados utilizando o HTML5 em conjunto com o Javascript para desvendarmos o funcionamento de cada um.



# Capítulo 2

## Entendendo o que é um Canvas

---

**A**prender a programar um computador não é fácil, e fica ainda pior quando não conseguimos transpor os conceitos que envolvem esta atividade para coisas e fatos de nosso dia a dia. Em outras palavras, se não utilizarmos metáforas durante o aprendizado de programação poucas pessoas serão capazes de compreender toda a abstração envolvida nesta atividade. Pois bem, pensando nisso e procurando facilitar a vida dos leitores deste livro a partir de agora este e todos os outros capítulos estarão repletos de metáforas, sempre buscando facilitar o aprendizado de cada um.

Vamos iniciar nossa jornada falando sobre o Canvas! Mas afinal de contas o que é isso?!?! Muito bem, não se assuste, pois o conceito é muito mais simples de ser compreendido do que a palavra. Vamos começar a explicação com um exercício mental. Imagine que você seja um pintor. Como todo pintor você precisa de uma superfície para fazer sua arte, ou seja, você precisa de uma tela, um painel, um pedaço de papel ou qualquer local onde você possa rabiscar, dar pinceladas, escrever, desenhar, etc. No mundo do desenvolvimento de games em HTML5 você é como se fosse o pintor e o Canvas é como se fosse a sua tela, ou seja, é o local onde você colocará em prática

a sua “arte”. É no Canvas onde os elementos visuais de seu game aparecerão e você, da mesma forma que o pintor, será o responsável por desenhar nesta tela (o Canvas) tudo aquilo que você espera que apareça no monitor do computador das pessoas que jogarão seu game.

Infelizmente nem tudo na vida é perfeito como gostaríamos que fosse, e com o HTML5 não é diferente. Um dos grandes problemas (que tende a desaparecer com o tempo) deste padrão é o fato de que nem todos os navegadores Web são compatíveis com esta tecnologia, ainda mais quando levamos em consideração aqueles mais antigos. Com isso, muitos dos recursos especificados por esse padrão não estão disponíveis em todos os “browsers”. Muitas vezes um ou outro recurso pode estar presente em um navegador específico e ausente em outro. Outras vezes o recurso está presente parcialmente, ou ele está presente mas apresenta defeitos. Deste modo, é importante apresentarmos quais navegadores dão suporte aos recursos do HTML5 que serão utilizados neste livro. A seguir são listados os navegadores Web que dão suporte ao recurso Canvas do HTML5:

- Mozilla Firefox 3.5 ou superior
- Opera 9 ou superior
- Internet Explorer 9 ou superior
- Chrome 4 ou superior
- Safari 4 ou superior
- iOS Safari 3.2 ou superior
- Opera Mobile 10 ou superior
- Android Chrome 2.1 (Eclair) ou superior

Portanto, se você pretende desenvolver games que utilizam o Canvas para serem executados em navegadores diferentes dos apresentados acima, não é possível garantir que eles funcionarão perfeitamente.

Para utilizar o Canvas precisamos iniciar criando um novo documento html para ser exibido no navegador Web. Vamos chamar este documento de “exemplo1.html”. Após criar o documento no seu editor de códigos favorito, digite o conteúdo abaixo (obs: não digite os números das linhas):

```
1: <html>
2:   <head><title>exemplo1</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela">Sem suporte ao HTML5</canvas>
5:   </body>
6: </html>
```

Na linha 1 iniciamos a criação do nosso documento HTML e definimos um título para o mesmo na linha 2. Em seguida, na linha 3, definimos que nossa página HTML terá o fundo na cor preta. (obs: suponho que você conheça um pouco de CSS se estiver lendo este livro). Finalmente, na linha 4, nosso Canvas é criado. Note que somos obrigados a identificá-lo com um “id”, que neste caso foi definido como “tela”. Perceba também que entre as “tags” de abertura e fechamento do Canvas foi colocada uma frase que será exibida somente se o navegador Web do usuário não tiver suporte ao uso do Canvas. Nas linhas 5 e 6 encerramos nosso documento HTML.

Se você gravar o arquivo “exemplo1.html” com o conteúdo do código apresentado e tentar carregá-lo em seu navegador, você notará que aparecerá apenas uma página preta semelhante à da figura 2. Contudo, apesar de não parecer, o Canvas já está presente, mas sem nada desenhado nele, ou seja, é como se fosse sua tela de pintura vazia.

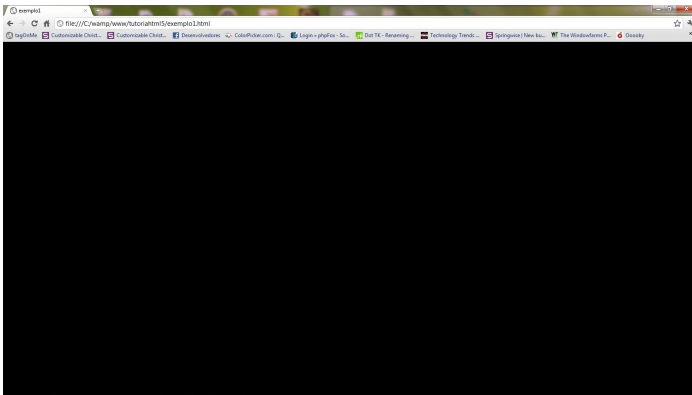


Figura 2 – Resultado gerado pelo código do exemplo 1.

Por enquanto é isso, contudo, não fique ansioso, pois no próximo capítulo começaremos a desenhar no Canvas e tudo começará a ficar mais interessante.

# Capítulo 3

## Desenhando círculos, quadrados e linhas

---

No capítulo anterior falamos um pouco sobre o que é um Canvas e aprendemos a criar o código necessário para que ele se torne disponível para nossa aplicação. Recordando um pouco sobre o tema, utilizei anteriormente a metáfora do pintor com sua tela para explicar o conceito, dizendo que um Canvas, para o desenvolvedor de games, é como a tela de um pintor onde os elementos do game serão desenhados. Seguindo a didática das metáforas apresentarei para você o equivalente aos pincéis de um pintor que lhe permitirá efetivamente “desenhar” na tela do computador.

Antes de falarmos sobre os “pincéis”, precisamos definir o tamanho de nossa tela, ou seja, quais serão as dimensões do Canvas estabelecendo uma área útil que poderá ser utilizada para trabalharmos. Veja abaixo o código apresentado no capítulo anterior para a criação de um Canvas. Note que agora a “tag” de criação de um Canvas possui dois atributos, “width” e “height”, representando respectivamente a largura e altura do Canvas. Neste exemplo o Canvas possui 640 pixels de largura e 480

pixels de altura. Crie um arquivo chamado exemplo2.html com o conteúdo apresentado abaixo:

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>
5:   </body>
6: </html>
```

Como você já sabe, o HTML5 é apenas uma linguagem de marcação para a apresentação de conteúdos em navegadores Web, ou seja, ela não fornece os recursos de uma linguagem de programação. Por esse motivo precisamos fazer uso do Javascript para manipular os recursos fornecidos pelo HTML5. Deste modo, partirei do pressuposto que você já conhece um pouco sobre esta linguagem para continuar nossa jornada. Para sermos capazes de desenharmos círculos, quadrados e linhas em nosso Canvas precisamos ter em mãos o equivalente aos pincéis de um pintor. No HTML5 estes “pincéis” são representados por algo que é chamado de **Contexto gráfico do Canvas**, portanto, para podermos desenhar na tela precisamos obter este “contexto”. Note no código abaixo que logo após a criação do Canvas inseri o código Javascript necessário para obter o seu contexto gráfico. Perceba que inicialmente precisei obter uma referência ao Canvas, utilizando para isso o “id” do mesmo e a instrução “getElementById” do Javascript. Em seguida, de posse desta referência, obtive o contexto gráfico do Canvas que foi gravado em uma variável que chamei de “context”. É importante que você perceba que ao obter o contexto gráfico do Canvas eu informei que desejava obter um “contexto 2D”

especificando que desenharemos elementos gráficos em duas dimensões (2D) em nossa tela:

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>
5:
6:     <script>
7:       var canvas = document.getElementById("tela");
8:       var context = canvas.getContext("2d");
9:     </script>
10:  </body>
11:</html>
```

A partir deste ponto já temos nossa “tela de pintura”, que é o Canvas, e nosso “pincel”, o contexto gráfico do Canvas representado pela variável “context”. Agora podemos iniciar a “pintura”. Vamos começar definindo a “cor da tinta” que será utilizada em nosso “pincel”. Para conseguirmos isso utilizaremos a variável do contexto gráfico (context) em conjunto com a instrução “fillStyle” que nos permite definir a cor desejada. Note que informamos a cor passando a intensidade das cores vermelho (red), verde (green) e azul (blue), ou seja, definimos os valores “rgb” de nossa cor para obtermos as variações de cores desejadas. Para cada componente da cor (red, green ou blue) definimos um valor que varia de 0 a 255, sendo o 0 a ausência daquela cor e o 255 a cor em sua máxima intensidade. Veja o código a seguir, em que definimos a “cor do pincel” como sendo vermelha:

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>

5:     <script>
6:       var canvas = document.getElementById("tela");
7:       var context = canvas.getContext("2d");

8:       context.fillStyle = "rgb(255,0,0)";
9:     </script>
10:   </body>
11:</html>
```

Agora que temos nossa “tela”, o “pincel” e a “cor da tinta” podemos começar a “fazer arte”! Começaremos desenhando um círculo e para isso precisaremos utilizar o “pincel” representado pela variável “context”. Antes de desenharmos um círculo precisamos saber do que ele é composto. Imagine que você está prestes a desenhar um círculo no papel utilizando uma caneta. Para iniciar você apoia a ponta da caneta no papel para fazer o desenho. Se o papel fosse o Canvas, o ponto onde você apoiou a caneta seria representado por suas coordenadas “x” (horizontal) e “y” (vertical). É importante destacar que no Canvas a coordenada x=0 e y=0 está localizada no canto superior esquerdo do mesmo, de modo que o valor de “x” aumenta para a direita e o de “y” aumenta para baixo. Pois bem, já sabemos que nosso círculo precisa ter um ponto inicial de onde ele começará a ser desenhado, representado pelos valores “x” e “y” deste ponto, além disso, se estamos desenhando um círculo, precisamos definir o raio do mesmo e representaremos este valor com a letra “r”. Precisamos



também informar se ao desenharmos o círculo o faremos completo ou parcial, ou seja, necessitamos definir, a partir de um ângulo inicial (por exemplo 0 graus), até qual ângulo final desenharemos (por exemplo 360 graus para um círculo completo). Para deixar a explicação mais simples informei estes ângulos em graus, porém, no HTML5, estes valores **devem ser informados em radianos**. Chamaremos o ângulo inicial de desenho de “ai” e o final de “af”. Finalmente, precisamos informar se o desenho será feito no sentido horário ou anti horário. Se o desenho for feito no sentido horário informaremos o valor “true”, caso contrário, o valor “false”. De posse de todas estas informações utilizaremos a instrução “arc” do nosso contexto gráfico para fazer o desenho. Veja abaixo a definição desta instrução:

```
arc(x, y, r, ai, af, true)
```

onde:

x = coordenada x do início do círculo

y = coordenada y do início do círculo

r = raio do círculo

ai = ângulo inicial, em radianos, do círculo

af = ângulo final, em radianos, do círculo

true = desenhar no sentido horário

Ainda não terminamos, pois precisamos informar ao HTML5 que ao iniciarmos o desenho do nosso círculo estamos iniciando um “traçado”. Fazemos isso com a instrução “beginPath”. Ao final precisamos informar que terminamos o

traçado com a instrução “closePath”. Finalmente dizemos ao HTML5 para preencher o círculo com a cor definida para nosso “pincel”. Ufa! parece complicado né? Mas não se assuste, pois na prática é muito mais simples que na teoria. Veja o código abaixo que desenha um círculo vermelho nas coordenadas  $x=320$  e  $y=240$  com raio de 30 pixels no sentido horário.

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>

5:     <script>
6:       var canvas = document.getElementById("tela");
7:       var context = canvas.getContext("2d");

8:       context.fillStyle = "rgb(255,0,0)";

9:       context.beginPath(); //início do traçado

10:      //Math.PI é a constante matemática PI, ou seja
        2*PI = 360 graus
11:      context.arc(320, 240, 30, 0, Math.PI*2, true);
12:      context.closePath(); //fim do traçado

13:      context.fill(); //preenchimento do círculo com a
        cor vermelha
14:    </script>
15:  </body>
16:</html>
```

Se você sobreviveu até aqui parabéns, pois o desenho de um círculo é o mais complicado. Desenhar um quadrado/retângulo é bem mais fácil, veja a seguir a

complementação do código anterior para desenhar um quadrado. Note que estamos utilizando a instrução “fillRect” com o seguinte formato:

```
fillRect(x, y, l, a)
```

onde:

x = coordenada x do canto superior esquerdo

y = coordenada y do canto superior esquerdo

l = largura do quadrado/retângulo

a = altura do quadrado/retângulo

OBS: O desenho será feito com a cor atual definida para o contexto gráfico.

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>
5:
6:     <script>
7:       var canvas = document.getElementById("tela");
8:       var context = canvas.getContext("2d");
9:
10:      context.fillStyle = "rgb(255,0,0)";
11:
12:      //Desenho do círculo
13:      context.beginPath(); //início do traçado
14:      //Math.PI é a constante matemática PI, ou seja
15:      2*PI = 360 graus
16:      context.arc(320, 240, 30, 0, Math.PI*2, true);
```

## Rodrigo Cesar Vertulo

```
13:         context.closePath(); //fim do traçado

14:         //preenchimento do círculo com a cor vermelha
15:         context.fill();

16:         //Desenho do quadrado
17:         context.fillRect(290, 270, 60, 60);
18:     </script>
19: </body>
20:</html>
```

Desenhar linhas também é simples. Veja o código abaixo e **leia os comentários** das instruções para o desenho de uma linha no próprio código.

```
1: <html>
2:   <head><title>exemplo2</title></head>
3:   <body style="background:#000000;">
4:     <canvas id="tela" width="640" height="480">Sem
       suporte ao HTML5</canvas>

5:     <script>
6:       var canvas = document.getElementById("tela");
7:       var context = canvas.getContext("2d");

8:       context.fillStyle = "rgb(255,0,0)";

9:       //Desenho do círculo
10:      context.beginPath(); //início do traçado
11:      //Math.PI é a constante matemática PI, ou seja
        2*PI = 360 graus
12:      context.arc(320, 240, 30, 0, Math.PI*2, true);
13:      context.closePath(); //fim do traçado

14:      context.fill(); //preenchimento do círculo com a
        cor vermelha
```

## Desvendando o HTML5 para Games – 2ª Edição

```
15:          //Desenho do quadrado
16:          context.fillRect(290, 270, 60, 60);

17:          //Desenho de uma linha
18:          context.strokeStyle = "rgb(255,0,0)"; //define a
          cor da linha
19:          context.beginPath(); //início do traçado da
          linha
20:          context.moveTo(270, 210); //ponto inicial da
          linha
21:          context.lineTo(370, 210); //ponto final da linha
22:          context.closePath(); //fim do traçado

23:          context.stroke(); //desenha efetivamente a linha

24:          </script>
25:          </body>
26:</html>
```

Chegamos ao final de mais um capítulo, porém, lembre-se que não basta apenas ler o que é apresentado aqui! É necessário que você exercite os conceitos aprendidos de modo que, a partir dos exemplos mostrados, você deverá criar seus próprios desenhos para se familiarizar com os comandos e forma de trabalho.

Rodrigo Cesar Vertulo

# Capítulo 4

## Movimentando círculos, quadrados e linhas

---

No capítulo 3 aprendemos a desenhar algumas formas geométricas simples em nosso Canvas. Agora chegou o momento de aplicar um pouco de movimento a estas formas. Veremos neste capítulo como movimentar elementos gráficos pelo Canvas fazendo algumas animações simples.

Antes de iniciarmos é preciso que você compreenda o conceito de “game loop”. No desenvolvimento de games um “game loop” é um ciclo que se repete durante todo o tempo em que o jogo está ativo, sendo responsável por obter os comandos do jogador, atualizar os elementos gráficos do jogo e processar a lógica do game. A cada vez que o “game loop” é executado todas essas atividades se repetem. Para movimentar nossos elementos gráficos precisaremos criar este ciclo que, inicialmente, apenas fará a animação de nossos desenhos.

Você deve ter percebido no capítulo anterior que ao desenharmos os círculos, quadrados e linhas no Canvas fomos obrigados a informar as coordenadas “x” e “y” de onde os desenhos começariam a ser feitos. De acordo com os valores definidos para essas coordenadas os desenhos eram feitos em

Rodrigo Cesar Vertulo

# ATENÇÃO

Você está lendo a versão de demonstração do EBook  
“Desvendando o HTML5 para Games: Uma Abordagem  
Prática”.

**Caso você deseje continuar a leitura do EBook, por favor  
instale o aplicativo completo disponibilizado no link abaixo.**

[https://play.google.com/store/apps/details?  
id=wacawaca.ebook.htmlgames.AOVBFGCKTATZEHVG](https://play.google.com/store/apps/details?id=wacawaca.ebook.htmlgames.AOVBFGCKTATZEHVG)