

Implementação de um Jogo Utilizando os Recursos do HTML5

Fernando Kenji da Costa Fujimoto¹; Walteno Martins Parreira Júnior²

¹ Analista de Sistemas, Especialista em Análise e Desenvolvimento de Sistemas Aplicados a Gestão Empresarial no Instituto Federal do Triângulo Mineiro, IFTM, Campus Uberlândia Centro.

² Professor do Instituto Federal do Triângulo Mineiro; Campus Uberlândia-Centro; Mestre em Educação, waltenomartins@iftm.edu.br.

Resumo: A versão atual do HTML disponibiliza elementos com recursos necessários para desenvolvimento de um jogo web de qualidade e com muita simplicidade. Com isso, este artigo tem por objetivo evidenciar a potencialidade das tecnologias baseadas na Web demonstrando benefícios, funcionalidades e deficiências perante o estágio atual da versão do HTML5 e, por meio de um estudo de caso, evidenciar as principais características e funcionalidades dos elementos da versão com foco em construção de jogos para Web. O HTML5 dispõe de elementos multimídias que dispensam o uso de tecnologias de terceiros como Flash, SilverLight e JavaFX.

Palavras-chave: Desenvolvimento de jogos. Jogos digitais. Canvas.

Abstract: The current version of HTML has elements with resources needed for development of a game web of quality and with great simplicity. With this, the aim of this article is to demonstrate the potential of Web-based technologies demonstrating benefits, features and deficiencies in the current stage version of HTML5 and, through a case study, highlighting the main features and functions of the elements of the version with a focus on building games for the Web. The HTML5 features multimedia elements that do not require the use of third-party technologies like Flash, Silverlight and JavaFX.

Keywords: Development of games. Games. Digital Canvas.

1 INTRODUÇÃO

Há um grande avanço nos processos relacionados à informação e serviços com a utilização das tecnologias baseadas na Web, pois com a sua imensa disponibilidade nos computadores (também tabletes e smartphones) foi possível ampliar o nível de interação dos usuários, e com maior demanda uma ampliação do desenvolvimento de aplicações, animações e jogos.

Porém, quando do desenvolvimento inicial da tecnologia HTML (HyperTextMarkupLanguage) ela tinha como finalidade apenas ofertar o acesso a textos. Somente com o decorrer do tempo é que as tecnologias baseadas na Web se desenvolveram com pouca documentação e recursos relacionados à interface e à interação com usuários (LEE, 1989).

Com a popularização da Web e com a ampliação do número de usuários é que o HTML começou a ser utilizado para propósitos de apresentação. Os desenvolvedores elegiam tags favoritas para alcançar o efeito visual pretendido, ao invés de utilizar as tags apropriadas de acordo com o significado a elas atribuído desde sua especificação. As tabelas eram utilizadas como ferramenta de layout em vez de uma maneira de exibir dados e, assim, a Web perdeu seu significado semântico e tornou-se um emaranhado de tags (BUDD et al., 2007).

Segundo o W3C (2014) alguns pontos marcantes do desenvolvimento histórico do HTML e da Web podem ser descritas na seguinte cronologia.

Em 1993 foi desenvolvido o HTML+, com algumas definições de aparência, tabelas e formulários. Em 1994 foi desenvolvido o HTML 2.0 e também a versão 3.0, com algumas padronizações das suas principais características (W3C, 2014, tradução nossa).

Em 1995 foi criado o HTML 3.2, com isso a Netscape e a Internet Explorer definiram seus respectivos padrões em relação ao contexto corrente da versão. No mesmo ano houve a primeira versão do JavaScript, criado por Brendan Eich, da Netscape, que permitia a manipulação de objetos de forma dinâmica por alguns métodos e eventos que não tinham no HTML (W3C, 2014, tradução nossa).

Em 1996 foi lançada a primeira folha de estilos chamada Cascading Style Sheets, ou simplesmente CSS, que constituíam formatação para complementação do HTML, que constava de mais de 60 propriedades de parte visual (W3C, 2014, tradução nossa).

Em 1997 foi lançado o HTML 4.0, que seria uma das versões mais utilizadas. Juntamente com o seu lançamento, os navegadores Netscape 4.0 e IE 4.0 mostraram um conjunto de tecnologias (HTML, JavaScript e CSS) que unidas dispunham de vários recursos, fazendo com que as páginas fossem muito mais dinâmicas; surgiu, assim, o termo DHTML (Dynamic HTML - HTML Dinâmico (W3C, 2014, tradução nossa).

Em 1998 foi lançada a versão CSS2, que possui 70 propriedades a mais que a versão anterior. Em 2000 foi lançado o XHTML 1.0, que consiste na junção de XML com HTML4.01 (W3C, 2014, tradução nossa).

Devido às limitações das versões anteriores, foi lançada em 2011 uma atualização da linguagem denominada de HTML5, o qual contém os elementos de multimídia e de semântica, que podem ser usados até mesmo para se desenvolver um jogo de boa qualidade sem a necessidade de utilizar um software proprietário, tais como o Adobe Flash e JavaFX entre outros, proporcionando uma nova tendência de desenvolvimento de jogos na Web.

Em termos de tecnologias básicas, pode-se dizer que os jogos na Web começaram a se tornar uma realidade com o DHTML. Tecnologias baseadas na Web, por meio de DHTML, conseguem reproduzir pequenas animações frequentemente utilizadas por aplicações e jogos. No entanto estas animações têm um custo de processamento que as limita a simples animações, já que estão intimamente relacionadas aos motores JavaScript de cada navegador e à utilização dos elementos HTML.

Com isso o DHTML não pode ser considerado uma linguagem de programação, é na verdade uma combinação da tecnologia HTML para o conteúdo, CSS para o design e JavaScript para interatividade. Assim, para sua utilização, é necessário apenas um navegador que suporte tais tecnologias, pois elas rodam do lado do cliente e não necessitam de nenhum servidor (LANGRIDGE, 2005).

Por ser a tecnologia base para definição de conteúdo, o HTML é uma linguagem de marcação de texto utilizada para publicação de páginas no WWW, cujos elementos são utilizados por um navegador para apresentação das informações na tela de forma que o usuário as compreenda. Além disso, páginas HTML podem ser criadas e processadas por uma grande quantidade de ferramentas, de um simples editor de texto a sofisticadas ferramentas como o Eclipse (W3C, 2014).

Por outro lado, a tecnologia que auxilia a interação das páginas com o usuário, o JavaScript, “é uma linguagem de programação simples e poderosa usada para adicionar comportamento dinâmico para um website” (LANGRIDGE, 2005, tradução nossa), provê uma melhor interação com o usuário, permitindo a criação de páginas ativas no sentido de dar a ele um retorno das suas ações quando na navegação pelas páginas.

O código em script não precisa ser compilado para sua execução, o processamento é realizado diretamente nos navegadores e roda do lado cliente da aplicação com a vantagem de economizar tempo de processamento e recursos do servidor, que por sua vez se restringe ao processamento de programas que rodam do lado servidor.

No entanto, com o surgimento das CSS, o desenvolvedor ganhou maior controle sobre a aparência da página, o que tornava possível a separação entre o aspecto visual do documento e seu conteúdo, novamente a marcação poderia ser simples, e o significado começava lentamente a voltar aos documentos.

Contudo, para o desenvolvimento de um jogo usando DHTML, necessita-se de um grande esforço por parte dos programadores, pois não existe nenhuma funcionalidade a fim de definir um espaço gráfico a ser trabalhado. Para se ter uma melhor definição nos jogos baseados nas tecnologias Web, necessita-se do uso de outros softwares para se fazer um jogo com qualidade, conforme será abordado na próxima subseção

Para o desenvolvimento de jogos de qualidade na Web sem contar com o recurso Canvas do HTML5, é de suma importância instalar previamente no navegador um software, uma máquina virtual, com recursos adicionais. Para isso, tem-se algumas opções como a utilização do Flash, Java FX (Applets) ou Silverlight.

O Flash é uma das tecnologias principais para desenvolvimento de jogos online na atualidade, no entanto, para seu uso, é necessária a instalação de um plug-in no navegador. No seu desenvolvimento, é utilizado um gráfico vetorial, além disso, suporta imagens e vídeos com alta qualidade, seu principal uso é para animações utilizadas na web. Flash é um produto da Adobe Systems Incorporated, o qual tem o foco voltado para montagem de efeitos visuais de alta qualidade em páginas Web.

Em contrapartida, JavaFX é uma plataforma lançada pela Sun, hoje Oracle, com a finalidade de desenvolvimento focado na web. O diferencial dessa plataforma é que um aplicativo desenvolvido para um desktop pode ser usado para a web e celular, utilizando-se de simples modificações em seus respectivos códigos. Também conta com uma máquina virtual, no caso a JVM (Java Virtual Machine). O modelo de execução segue o dos clássicos Applets Java.

O Microsoft Silverlight foi desenvolvido para competir com o Flash, propondo suporte de várias linguagens de programação, tais como C# e VB. Além disso, podem ser destacados os codecs de vídeo, que transmitem de forma rápida e com qualidade. SilverLight usa a máquina virtual da plataforma .NET, o CLR (Common Language Runtime).

Com todas as tecnologias citadas nessa seção, podem-se desenvolver aplicações e jogos de qualidade para Web de forma semelhante, utilizando-se de frameworks e máquinas virtuais instaladas nos navegadores ou no computador local, contando com vários recursos para sua implementação. Na seção seguinte, será apresentado o HTML5, com seus principais novos recursos e, juntamente com eles, o canvas, que possibilita a construção e execução de gráficos 2D diretamente no navegador apenas com uso de Javascript como ambiente de execução.

2 DESENVOLVIMENTO

2.1 Considerações iniciais

Considerando que a proposta é o desenvolvimento de um jogo utilizando os recursos do HTML5, será descrita nesta seção as etapas e recursos necessários.

Assim, na subseção Material e Métodos é exposta a linguagem HTML5, indicando os novos elementos e recursos incorporados, desempenhando um papel muito importante nos padrões de desenvolvimento Web, onde essa tecnologia pode substituir os utilizados softwares instalados no navegador.

E na subseção Resultados e discussão será apresentado um estudo de caso que tem como objetivo o desenvolvimento de um jogo utilizando canvas/HTML5, demonstrando as principais características e recursos que essa tecnologia possui, além de apresentar o resultado obtido através de sua utilização.

Com a dependência de agilidade no setor de desenvolvimento Web, será mostrado como é simples o desenvolvimento de um jogo utilizando os recursos do HTML5 em especial o Canvas.

2.2 Material e Métodos

Considerando a necessidade de uma reorganização do HTML, a W3C lançou a primeira versão em testes do HTML5, que substituiu o HTML 4.01, fornecendo novos recursos que são necessários para aplicações web mais robustas, tais como elementos de semântica e de multimídias. Além disso, padroniza muitas características da plataforma web que os desenvolvedores usam há anos, porém nunca foram levantadas ou documentadas. Conforme as versões anteriores, o HTML5 é projetado para ser multiplataforma (PILGRIM, 2010).

Quadro 1 - Relação Browser e funcionalidades HTML5

	Safari	Chrome	Opera	Firefox	IE 8	IE 9
Local Storage	s	s	s	s	s	s
Histórico de Sessão	s	s	s	s	s	s
Aplicações Offline	s	s	n	s	n	n
Novos tipos de campos	s	s	s	n	n	n
Form: Autofocus	s	s	s	n	n	n
Form: Autocomplete	n	n	s	n	n	n
Form: Required	s	s	s	n	n	n
Video, Audio e Canvas Text	s	s	s	s	n	s

Fonte: Borges (2014)

Para a execução do HTML5, não é exigido um determinado sistema operacional, pode ser o Windows, MacOS X, Linux entre outros, o que é necessário é um navegador atualizado. O Quadro 1 mostra os principais navegadores relacionados com as funcionalidades do HTML5 quando do lançamento da versão. Os navegadores web que vêm pré-instalados em iPhones, iPads e celulares com Android apresentam um suporte para HTML5. Assim, a Microsoft apresentou a versão 9 do Internet Explorer apoia algumas funcionalidades HTML5, mas não todas.

As principais características do HTML5 incluem novos elementos com dois principais objetivos: implementam recursos multimídias e elementos de atribuições semânticas. A primeira característica que pode ser destacada é o controle multimídia sem intermédio de plug-ins: através dos elementos áudio e vídeo, é possível exibir

conteúdos de alta qualidade com poucas linhas de códigos. Por outro lado, o elemento canvas disponibiliza recursos para desenhos de forma vetorial.

A independência de tecnologias proprietárias é sempre um fator positivo na construção de aplicações baseadas na Web, pois, quanto menor o número de requisitos para uma experiência satisfatória do usuário, maiores são as chances de uma aplicação alcançar os objetivos almejados.

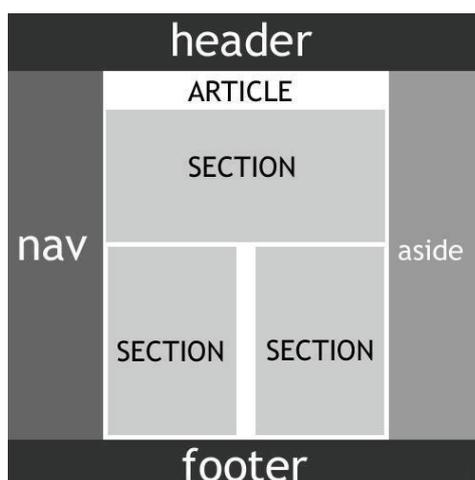
As definições dos elementos com semântica nesta versão são mais específicas tanto para o nível de bloco quanto inline, explicitando o propósito de cada tag na codificação do layout, favorecendo a leitura humana ou mesmo a realizada por mecanismos de buscas assim como por qualquer outro dispositivo. No Quadro 2 os principais elementos de semântica: a) Elementos inline: time, mark, progress, meter e ruby; b) Elementos de bloco: section, nav, article, aside, header e footer.

Quadro 2: Código da estrutura básica de semântica de uma página

```
<html>
<title>Titulo</title>
<body>
<header>
  <h1>teste HEADER</h1>
</header>
<nav>
  <h1>Página Inicial teste NAV</h1>
</nav>
<article>
  <section>
    <h1>Nome da Notícia ARTICLE/SECTION <h1>
  </section>
</article>
```

Conforme o código do Quadro 2, a estrutura da página é definida por seus respectivos elementos, com isso cada elemento se enquadra em uma determinada região da página, como apresentado na figura 1.

Figura 1: Estrutura semântica



Fonte: Carvalho (2010)

Para a execução do HTML5, não é preciso um sistema operacional determinado como Windows, Mac OS X, Linux ou qualquer outro, é necessário apenas um navegador atualizado. Os principais navegadores estendem todas as funcionalidades do HTML5. Os navegadores web que vêm pré-instalados em iPhones, iPads e celulares com Android possuem um excelente suporte para HTML5 (W3C, 2014).

HTML5 inclui novos elementos com dois principais objetivos: implementam recursos multimídias e elementos de atribuições semânticas, proporcionando uma Web muito mais organizada e robusta (PILGRIM, 2010).

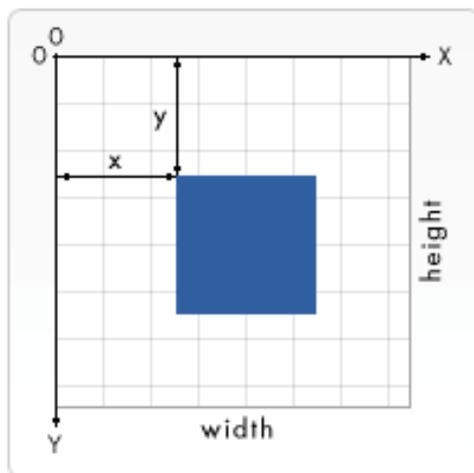
A primeira característica a ser destacada é o controle multimídia sem intermédio de plug-ins: através dos elementos áudio e vídeo, é possível exibir conteúdos de alta qualidade com poucas linhas de códigos. E o elemento canvas disponibiliza recursos para desenhos de forma vetorial (PILGRIM, 2010).

Um fator positivo é a independência de tecnologias proprietárias para a construção de aplicações baseadas na Web, pois, quanto menor o número de requisitos maiores são as chances de uma aplicação alcançar os objetivos almejados no atendimento ao usuário.

Canvas é um dos elementos adicionados no HTML5 que permite definir uma área como se fosse uma tela e o seu acesso ocorre através do JavaScript, permitindo um completo domínio sobre os elementos gráficos da área do navegador sem a utilização de plug-ins (MOZILLA, 2015). Segundo Sheridan (2013), “[canvas é] uma tela de bitmap

dependente de resolução que pode ser usada para a renderização de elementos gráficos ou outras imagens visuais rapidamente”.

Figura 2: Coordenadas do Canvas



Fonte: Mozilla (2015)

Para a utilização dos métodos de canvas é importante que as passagens de parâmetros referentes às coordenadas de localização em seu gráfico sejam bem definidas, determinando o local exato no gráfico de canvas, como observadas na figura 2.

O HTML5 contribui para o desenvolvimento de jogos de uma forma simples, que não exige grande esforço na aprendizagem como acontece quando da utilização de frameworks complexos. Por outro lado, a tecnologia que auxilia a interação das páginas com o usuário, o JavaScript, “é uma linguagem de programação simples e poderosa usada para adicionar comportamento dinâmico para um website” (STEFANOV, 2008), que provê uma melhor interação com o usuário, permitindo a criação de páginas ativas dando ao usuário um retorno das suas ações quando na navegação pelas páginas.

O código quando desenvolvido em script não precisa ser compilado para ser executado pois o processamento é realizado no navegador e roda do lado cliente da aplicação e desta forma apresenta a vantagem de economizar tempo de processamento e de recursos do servidor, que por sua vez se restringe ao processamento de programas que rodam do lado servidor (STEFANOV, 2008).

O JavaScript e o HTML5, com os recursos acrescidos nesta versão e, também, o canvas, que possibilita a construção e execução de gráficos 2D diretamente no

navegador apenas com uso de Javascript como ambiente de execução são uma boa opção.

O Quadro 3 apresenta os principais métodos para a construção de desenhos utilizando a API de canvas.

Quadro 3: Métodos para desenho da API de canvas

`fillRect(x, y, largura, altura)` - Desenha um retângulo preenchido.

`fillStyle()` - Define a cor do desenho.

`strokeRect(x, y, largura, altura)` - Desenha as linhas de um retângulo sem preenchimento.

`clearRect(x, y, largura, altura)` - Limpa algum desenho da tela.

`drawImage()` - Mostra uma imagem na tela.

`beginPath()` - Inicia o desenho de um caminho.

`moveTo(x, y)` - Move a posição de um desenho.

`lineTo(x, y)` - Desenha uma linha.

`arc(x, y, raio, InicioAngulo, FimAngulo, Sentido)` - Desenha um círculo.

`quadraticCurveTo(cp1x, cp1y, x, y)` - Desenha uma curva arredondada.

`bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)` - Desenha uma curva de bezier.

`rect(x, y, largura, altura)` - Desenha um retângulo no path.

`closePath()` - Faz a linha que liga ao início do path.

2.3 Resultados e Discussão

Neste estudo de caso sobre o desenvolvimento de um jogo com a utilização da tecnologia baseada na versão do HTML5 com a apresentação das principais características e funcionalidades do recurso denominado canvas. O game desenvolvido é um modelo tradicional de quebra de tijolos que pode ser observado na Figura 3.

Inicialmente para o desenvolvimento do game deve ser criada uma página HTML simples e então, no corpo dela, deve ser adicionado o elemento canvas e então devem ser definidos os atributos relacionados à largura e altura. Após a definição do canvas na página principal é então necessário criar um arquivo JavaScript que vai conter a regra do jogo e utilizar os métodos de canvas. A primeira função que deve ser

executada é a função `inicio()`, que ativa o contexto de canvas em 2D e define o espaço a ser trabalhado (Quadro 4).

Figura 3: A tela do jogo



Fonte: Elaborado pelo autor (2015)

Quadro 4: Função que Contextualiza o Canvas

```
function inicio() {
var canvas = document.getElementById("canvas");
ctx = canvas.getContext("2d");
largura = canvas.width;
altura = canvas.height;
inicio_pegador();
inicio_tijolos();
return setInterval(desenha, 10);
}
```

Fonte: Elaborado pelo autor (2015)

Conforme o Quadro 4, nesta função é chamada uma função do JavaScript denominada `setInterval()`, que tem como objetivo acionar a função `desenha()` em um determinado intervalo de tempo,.

Considerando o objeto definido em 2D anteriormente é que permite a utilização dos métodos e propriedades da representação gráfica, e então é exibida na área delimitada pelo elemento canvas. Permitindo que objetos de forma simples sejam desenhados e que serão manipulados conforme as regras definidas no jogo.

Para a criação de animações na área do canvas deve-se considerar que vai ocorrer o redesenho da imagem a cada alteração sempre obedecendo a regra do jogo e a frequência de atualização definida. Foi implementada um conjunto de regras de movimentação definida pela função desenha(). Esta função desenha todos os objetos na tela usando as respectivas funções considerando o intervalo de 10 milissegundos, atribuindo aos retângulos os valores que definem se estão invisíveis ou não.

Para a criação do game, primeiramente deve ser criada uma página HTML simples e, no corpo dela, deve ser adicionado o elemento canvas onde estão definidos os atributos relacionados à largura e altura, conforme pode ser observado no Quadro 5.

Quadro 5: Definindo o canvas do Jogo

```
<canvas id="canvas" width="500" height="500">  
Seu Navegador Não suporta HTML5 </canvas>
```

Fonte: Elaborado pelo autor (2015)

Após a definição do canvas na página principal, é necessário criar um arquivo JavaScript com a finalidade de definir a regra do game e utilizar os métodos de canvas.

A primeira função que deve ser acionada é a função inicio(), que concebe o contexto de canvas em 2D e define o espaço a ser trabalhado. Nesta função é chamada uma função do JavaScript cujo nome é setInterval(), que tem como objetivo acionar a função desenha() em um determinado intervalo de tempo, conforme o Quadro 6.

Quadro 6: Função que contextualiza canva

```
function inicio() {  
  toca_audio();  
  var canvas = document.getElementById("canvas");  
  ctx = canvas.getContext("2d");  
  largura = canvas.width;  
  altura = canvas.height;  
  inicio_pegador();  
  inicio_tijolos();  
  return setInterval(desenha, 10);  
}
```

Fonte: Elaborado pelo autor (2015)

A partir do objeto contextualizado em 2D, é possível utilizar de métodos e propriedades da representação gráfica que é exibida na área delimitada pelo elemento canvas no documento. Com isso desenham-se objetos de forma simples, os quais serão manipulados de acordo com a regra definida do jogo, conforme pode ser verificado no código apresentado no Quadro 7.

Para o desenho dos retângulos que aparecem no jogo foi criada uma função cujo objetivo é instanciar um vetor de tijolos definindo sua respectiva posição no gráfico de canvas, e ele deve ser desenhado conforme pode ser observado no Quadro 8.

Quadro 7: Funções que utilizam métodos de canvas

```
function circulo(x, y, r) {
  ctx.beginPath();
  ctx.arc(x, y, r, 0, Math.PI * 2, true);
  ctx.closePath();
  ctx.fillStyle = 'red';
  ctx.fill();
}

function retangulo(x, y, w, h) {
  ctx.fillStyle = 'blue';
  ctx.fillRect(x, y, w, h);
}
```

Fonte: Elaborado pelo autor (2015)

Quadro 8: Função que desenha os retângulos

```
function inicio_tijolos() {
  linha = 6;
  colun = 6;
  tijolo_lar = (largura / colun) - 1;
  tijolo_alt = 15;
  aux = 3;
  tijolo = new Array(linha);

  for (i = 0; i < linha; i++) {
    tijolo[i] = new Array(colun);

    for (j = 0; j < colun; j++) {
      tijolo[i][j] = 1;
    }
  }
}
```

Fonte: Elaborado pelo autor (2015)

Conforme o Quadro 9 é possível a utilização de imagens na tela do jogo. Para isso, é necessário a criação de uma variável do tipo image e definir qual imagem será exibida pela propriedade src, com isso o método drawImage() do contexto de canvas é utilizado afim de definir o local e a medida exata em que será colocada a imagem na tela.

Quadro 9: Função da imagem de fundo

```
function imagemfundo() {
  var img = new Image();
  img.src = 'tijolos_cinza.jpg';
  ctx.drawImage(img, 0, 0);
}
```

Fonte: Elaborado pelo autor (2015)

A estrutura do código do Quadro 10 mostra a movimentação da paleta (barra controlada pelo jogador), que utiliza uma função do JavaScript que determina se a tecla está pressionada ou não.

Quadro 10: Código para movimentação da paleta

```
function tecla_1(evt) {  
    if (evt.keyCode == 39)  
        paradireita = true;  
    else if (evt.keyCode == 37)  
        paraesquerda = true;  
    }  
    function tecla_2(evt) {  
        if (evt.keyCode == 39)  
            paradireita = false;  
        else if (evt.keyCode == 37)  
            paraesquerda = false;  
        }  
    document.onkeydown = function(event) {  
        tecla_1(event)  
    };  
    document.onkeyup = function(event) {  
        tecla_2(event)  
    };  
};
```

Fonte: Elaborado pelo autor (2015)

A criação de animações na área do canvas implica no redesenho de toda a imagem apresentada a cada modificação necessária, obedecendo a regra do jogo e a frequência de atualização. Para este desenho contínuo, foi implementada toda a regra de movimentação com função `desenha()`, esta regra desenha todos os objetos na tela por suas respectivas funções nos intervalos de 10 milissegundos, atribuindo aos retângulos os valores que filtram se eles estarão visíveis ou não, conforme o fragmento de código apresentado no Quadro 11.

Quadro 11: Função que desenha

```

function desenha() {
  limpa();
  imagemfundo();
  circulo(x, y, 10);
  // desenha para direita
  if (paradireita) {
    if ((pegador_t + 70) < largura)
      pegador_t += 5;
  }
  // desenha para esquerda
  else if (paraesquerda) {
    if ((pegador_t) >= 0)
      pegador_t -= 5;
  }
  retangulo_pa(pegador_t, altura - pegador_a, pegador_l, pegador_a);
  // desenha retangulos 'tijolos'
  for (i = 0; i < linha; i++) {
    for (j = 0; j < colun; j++) {
      if (tijolo[i][j] == 1) {
        retangulo((j * (tijolo_lar + aux)) + aux, (i * (tijolo_alt + aux)) + aux, tijolo_lar, tijolo_alt);
      }
    }
  }
  rowheight = tijolo_alt + aux;
  colwidth = tijolo_lar + aux;
  row = Math.floor(y / rowheight);
  col = Math.floor(x / colwidth);
  if (y < linha * rowheight && row >= 0 && col >= 0 && tijolo[row][col] == 1) {
    dy = -dy;
    tijolo[row][col] = 0;
  }
  if (x + dx > largura || x + dx < 0)
    dx = -dx;
  if (y + dy < 0)
    dy = -dy;
  else if (y + dy > altura - 12) {
    if (x > pegador_t && x < pegador_t + pegador_l) {
      dy = -dy;
    } else {
      teste = 10;
      clearInterval(intervallId);
    }
  }
  x += dx;
  y += dy;
}

```

Fonte: Elaborado pelo autor (2015)

Após o desenvolvimento do jogo é possível escrever que os elementos multimídias, em especial canvas de HTML5, permitem o controle total sobre o ambiente, permitindo o acesso aos métodos e propriedades do contexto 2D para a

criação de desenho com alta qualidade e performance. E assim, há a possibilidade de efetuar o desenvolvimento de um jogo sem demandar grande esforço do desenvolvedor. E com a utilização dessa tecnologia não há a necessidade de utilizar software de terceiros, os denominados plug-ins.

3 CONCLUSÃO

Com o desenvolvimento do jogo apresentado, foi possível mostrar que é possível utilizar os recursos do HTML5 para desenvolver atividades multimídia. Assim, pode-se escrever que o HTML5 apresenta inovações que preenchem uma lacuna nas recomendações da linguagem HTML4.01. Logo, o HTML5 apresenta uma oportunidade para o desenvolvimento de aplicativos e jogos com os novos elementos de semântica e multimídia.

Assim, as novas especificações da linguagem trazem recursos que facilitam o processo de desenvolvimento e trazem independência tecnológica de fornecedores e softwares proprietários.

O mercado de jogos exige cada vez mais velocidade e qualidade, e com a adoção do HTML5 é possível verificar o atendimento a estas exigências, pois é uma plataforma de fácil entendimento e possibilitando o desenvolvimento de jogos com qualidade e bom desempenho.

REFERÊNCIAS

BORGES, Kelvis Antonio R. **Compatibilidade entre os navegadores com o HTML5**. 2014. Disponível em: <<http://fabrica.ms.senac.br/2014/10/compatibilidade-entre-os-navegadores-com-o-html5/>>. Acesso em: 20 set. 2015.

BUDD, Andy et al. **CSS mastery: advanced web standards solutions**. Rio de Janeiro: Prentice Hall, 2007.

CARVALHO, Raviel. **HTML5: semântica, SEO e organização**. 2010. Disponível em: <<http://imasters.com.br/artigo/16598/web-standards/html5-semantica-seo-e-organizacao?trace=1519021197&source=single>>. Acesso em: 5 set. 2016.

LANGRIDGE, Stuart. **DHTML utopia: modern web design using JavaScript & DOM**. Australia: Sitepoint, 2005.

LEE, Tim-Berners. **Information management**: a proposal. 1989. Disponível em: <<http://www.w3.org/History/1989/proposal.html>> . Acesso em: 20 set. 2015.

MOZILLA Developer. **Drawing shapes with canvas**. 2015. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Drawing_shapes> Acesso em: 25 set. 2015.

PILGRIM, Mark. **HTML5: up and running**. United States of America: Reilly Media. 2010.

SHERIDAN, Melcolm. **HTML5: guia do desenvolvedor para o HTML5 Canvas**. 2013. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dn151487.aspx> > . Acesso em: 5 set. 2016.

STEFANOV, Stoyan. **Object oriented JavaScript**: create scalable, reusable high-quality JavaScript applications and libraries. Birmingham, UK: Packt, 2008.

W3C - WORLD WIDE WEB CONSORTIUM - **HTML5**: a vocabulary and associated APIs for HTML and XHTML. 2014. Disponível em: <<https://www.w3.org/TR/2014/REC-html5-20141028/>>. Acesso em: 25 set. 2015.

Periódico de Pesquisas e
Trabalhos de conclusão de curso
IFTM - Campus Uberlândia Centro

2016



ISSN: 2526-2041

Organizador:

Prof. Dr. Ricardo Soares Bôaventura

PERIÓDICOS



INSTITUTO
FEDERAL
Triângulo Mineiro

Campus
Uberlândia Centro

Copyright 2016

IFTM – Campus Uberlândia Centro
Todos os direitos reservados

Este trabalho está sujeito a direitos de autor. Todos os direitos são reservados, no todo ou em parte, mais especificamente os direitos de tradução, reimpressão, reutilização de ilustrações, re-citação, emissão, reprodução em microfilme ou de qualquer outra forma, e armazenamento em bases de dados. A permissão para utilização deverá ser sempre obtida do IFTM Campus Uberlândia Centro. Por favor, entrar em contato com pesquisa.udicentro@iftm.edu.br.

Organizado por:

Ricardo Soares Bôaventura

Coordenador da temática de pesquisa:

Poliana Cristina de Oliveira Cristo Diniz

Coordenador da temática de TCC de graduação:

Walteno Martins Parreira Júnior

Coordenador da temática de TCC de pós-graduação:

Luciana Araújo Valle de Resende

Bibliotecária:

Márcia Aparecida Bellotti

Comitê Científico

Márcio Bonesso

Rodrigo Gomes Santana

Karina Estela Costa

Cristiano Borges dos Santos

Dickson Duarte Pires

Sirley Cristina Oliveira

Gyzely Suely Lima

Marcelo Dias de Almeida

Priscila Santos de Araújo

Luiza Helena Araújo de Oliveira

Leonice M. R. Souza Pereira

Poliana Cristina de Oliveira Cristo

Walteno Martins Parreira Júnior

Robson Humberto Rosa

Luciana Araújo Valle de Resende

Nilton Spindola Júnior

Maria Fernanda Soares de Almeida

Ricardo Soares Bôaventura

Márcia Aparecida Bellotti

Capa

Alexandre Miranda Machado

Alvaro Tavares Latado

Arthur Augusto Bastos Bucioli

Vinicius Carvalho Cazarotti