

figurethplop figureFigure

ÚNIOR12

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE  
DO SUL  
CÂMPUS BENTO GONÇALVES-RS

Jean Carlo Machado

# Limitações no desenvolvimento de jogos 2D em HTML5

Bento Gonçalves, Dezembro 2014

Jean Carlo Machado

# Limitações no desenvolvimento de jogos 2D em HTML5

Trabalho de conclusão de curso

Projeto de Pesquisa apresentado junto ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, como requisito parcial ao desenvolvimento do Trabalho de Conclusão de Curso.

Orientador: Me. Rafael Jaques

Bento Gonçalves, Dezembro 2014

## **RESUMO**

Com a introdução dos novos recursos de multimídia oferecidos pelo HTML5 a viabilidade de construção de jogos iterativos com as tecnologias da WEB voltou a ser assunto relevante. Esse trabalho busca avaliar as limitações e possíveis dificuldades à se enfrentar na construção de um jogo 2D utilizando HTML5 e demais tecnologias relacionadas.

Como caráter de avaliação fez-se uma pesquisa de ferramentas e utilitários disponíveis e construiu-se um jogo 2D. Com base na pesquisa efetuada e na experiência desenvolvida, constatamos que...

**Palavras-chave:** HTML5, JOGOS 2D, PROBLEMAS

## **Lista de Figuras**

1	Os cinco jogos mais bem vendidos de todos os tempos segundo à VGChartz, 2013). . . . .	5
2	Funcionalidades implementadas para o projeto Apache Cordova. . . . .	10

## **Lista de Tabelas**

1	Cronograma do projeto. . . . .	14
---	--------------------------------	----

# Conteúdo

<b>1</b>	<b>CONTEXTUALIZAÇÃO</b>	<b>5</b>
1.1	HISTÓRIA E BENEFÍCIOS DOS JOGOS . . . . .	5
1.2	JOGOS NATIVOS VS JOGOS WEB . . . . .	6
1.3	HTML . . . . .	6
<b>2</b>	<b>PROBLEMA</b>	<b>6</b>
<b>3</b>	<b>OBJETIVOS</b>	<b>7</b>
3.1	OBJETIVO GERAL . . . . .	7
3.2	OBJETIVOS ESPECÍFICOS . . . . .	7
<b>4</b>	<b>JUSTIFICATIVA</b>	<b>7</b>
<b>5</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>8</b>
5.1	HTML5 . . . . .	8
5.2	MOTORES DE FÍSICA . . . . .	8
5.3	SOM E VÍDEO . . . . .	9
5.4	ENTRADA DE COMANDOS . . . . .	9
5.5	DETECÇÃO DE RECURSOS . . . . .	9
5.6	TECNOLOGIAS POLYFILL . . . . .	9
5.7	AMBIENTES PARA DESENVOLVIMENTO HTML5 . . . . .	10
5.8	FRAMEWORKS PARA DESENVOLVIMENTO DE JOGOS HTML5 . . . . .	11
5.9	EMPACOTADORES . . . . .	11
5.10	METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE PARA A CONSTRUÇÃO DE GAMES . . . . .	11
5.11	USABILIDADE E TRANSPARÊNCIA . . . . .	12
5.12	TRABALHOS SIMILARES . . . . .	13
<b>6</b>	<b>METODOLOGIA</b>	<b>13</b>

<b>7</b>	<b>RESULTADOS</b>	<b>14</b>
<b>8</b>	<b>CONCLUSÃO</b>	<b>14</b>
<b>9</b>	<b>CRONOGRAMA</b>	<b>14</b>

# 1 CONTEXTUALIZAÇÃO

## 1.1 HISTÓRIA E BENEFÍCIOS DOS JOGOS

Há muito os jogos são utilizados como meio de entretenimento, mas além da diversão propiciada estes podem ser benéficos de diversas outras formas como: na apuração da capacidade lógica e motora, facilidade em tomada de decisões, redução de stress, entre outros (GALLAGHER, 2013).

Na informática, os jogos estão presentes desde meados dos anos 60; no início, apenas protótipos de laboratórios, os jogos eram aplicativos extremamente rudimentares e totalmente dependentes da plataforma onde eram desenvolvidos.

Em 1972, com a criação da Atari, os jogos eletrônicos se popularizaram. Criou-se um mercado competitivo onde grandes corporações internacionais disputavam para criar as melhores plataformas e os melhores jogos. Essas corporações, dentre as maiores à dizer: Nintendo, Sony e Microsoft, desenvolveram novas tecnologias, criaram novos conceitos, a fim de prover maior entretenimento e imersão ao usuário. Exemplos de tecnologias desenvolvidas ou melhoradas para os games são: renderizadores de elementos de 3 dimensões (3D), bibliotecas de física avançada, produções musicais específicas para os jogos, bem como o acréscimo no nível de detalhamento gráfico devido às GPU's *Graphics Processing Unit's* (ou unidades de processamento gráfico) mais poderosas utilizadas.

A indústria dos vídeo games foi rejuvenescida com a introdução da Nintendo ao final nos anos 80. Com melhoras significativas nos gráficos, no realismo, e com a introdução de jogos violentos, a Nintendo conseguiu uma dominância de mercado temporária (BUCHMAN, 1996). Atualmente, devido à disponibilidade de internet de alta velocidade, a sofisticação das técnicas dos jogos, e a grande compatibilidade dos hardwares, aumentou-se o fluxo de entrada de dinheiro, o que está gerando um grande boom na indústria dos jogos (AK, 2012).

O mercado global de jogos cresceu de 70.5 bilhões de dólares em 2011 e espera chegar a 117.9 bilhões em 2015, crescendo em média 13.7 de 2011 a 2015 (AK, 2012).

Pos	Game	Platform	Year	Genre	Publisher	North America	Europe	Japan	Rest of World	Global
1	Wii Sports	Wii	2006	Sports	Nintendo	40.59	24.56	3.77	12.46	81.38
2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
3	Mario Kart Wii	Wii	2008	Racing	Nintendo	14.66	10.28	3.65	5.22	33.82
4	Wii Sports Resort	Wii	2009	Sports	Nintendo	14.98	8.46	3.20	5.11	31.75
5	Pokémon Red / Green / Blue Version	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

Figura 1: Os cinco jogos mais bem vendidos de todos os tempos segundo à VGChartz, 2013) .

Além de obter uma estimativa de vendas dos jogos mais populares, através desta lista, é possível ter-se uma concepção de quais gêneros de jogos atraem mais os usuário. O gênero de maior influência na lista é o de esportes, seguido igualmente por plataforma, corrida e RPG. <sup>1</sup>

<sup>1</sup>A VGChartz é uma bem conceituada empresa na área de pesquisas de mercado especificamente de jogos eletrônicos.



## 1.2 JOGOS NATIVOS VS JOGOS WEB

Há diversas decisões há tomar ao decidir fazer um jogo, uma das críticas é: em que plataformas o sistema será disponibilizado? e quais as tecnologias utilizadas para fazê-lo?. Existem as seguintes possibilidades:

**Jogos nativos:** A possibilidade mais comum é a criação de jogos nativos, um para cada plataforma alvo. Esta estratégia contém os seguintes benefícios:

- Aplicativos nativos propiciam uma sensação de continuidade aos demais aplicativos do sistema operacional;
- Normalmente tem melhor performance do que aplicativos WEB
- Permitem maior controle sobre o dispositivo em que estarão rodando

**Jogos WEB** Jogos WEB são aqueles que utilizam HTML5, JavaScript e CSS como tecnologias. Jogos WEB trazem nos seguintes benefícios:

- Única base de código;
- Relativa facilidade de se encontrar desenvolvedores capazes ou interessados;
- Custo de produção e manutenção reduzido.

## 1.3 HTML

Tendo em mente estas considerações, este trabalho objetiva auxiliar o desenvolvimento de jogos em HTML5 fornecendo um estudo de caso - através de um jogo, outrossim, um resumo dos problemas encontrados, a fim de que estes possam servir como uma espécie de caminho facilitado à todos os interessados no desenvolvimento de jogos para a WEB.

## 2 PROBLEMA

A falta de disponibilidade de trabalhos acadêmicos, bem como a carência de definições concretas sobre a viabilidade da atual versão do HTML5 - quando utilizado no desenvolvimento de jogos, acaba por monopolizar a construção de jogos nativos as plataformas alvo. Esta realidade corrobora para uma espécie de consenso à respeito da inviabilidade de construir-se jogos competitivos em HTML5.

Os custos adicionais no ciclo vida de um jogo nativo, para diversas plataformas, é muito alto para ser considerado um problema trivial.

## **3 OBJETIVOS**

### **3.1 OBJETIVO GERAL**

O propósito deste trabalho é investigar o processo desenvolvimento de jogos utilizando-se das tecnologias da WEB, à dizer: HTML5, CSS3 e JavaScript. Busca-se através disso apontar possíveis problemas no processo de desenvolvimento oriundos do atual estado de implementação do HTML5 e demais tecnologias. O estudo de caso poderá servir como indicador de viabilidade de projetos de jogos para WEB, bem como fonte de experiência para aqueles buscando se conhecer às tecnologias.

### **3.2 OBJETIVOS ESPECÍFICOS**

- Estudar o HTML5 especificamente canvas, áudio, vídeo e controle de entrada de comandos;
- Estudar frameworks de desenvolvimento de jogos em HTML5;
- Estudar engines (motores) de física para HTML5;
- Estudar tecnicas de detecção de recursos, de dispositivos;
- Estudar tecnologias para preenchimento de gaps HTML5;
- Desenvolver o jogo utilizando somente tecnologias OpenSource (código aberto);
- Identificar os pontos relevantes, fraquezas e acertos, da implementação atual do HTML5, em diferentes tipos de plataformas;

## **4 JUSTIFICATIVA**

- O trabalho justifica-se primeiramente pelo quesito inovação, devido ao HTML5 ser uma tecnologia não totalmente consolidada, sendo de utilidade à comunidade trabalhos que revisem seus recursos, quanto mais em escopos específicas como o HTML5 para a construção de jogos 2D;
- O método de construção de aplicativo pela WEB permite uma única base de código o que habilita a disponibilidade de aplicativos com maior velocidade e significativas reduções nos custos;
- Muitos desenvolvedores estão familiarizados com as tecnologias da WEB ou apontam interesse na tecnologia;
- A opinião comum tende para soluções nativas em detrimento do desenvolvimento de jogos;

## 5 REVISÃO BIBLIOGRÁFICA

### 5.1 HTML5

O padrão HTTP é conhecido por ser o principal fomentador da WEB e a especificação de texto deste padrão é o conhecido HTML, em sua concepção inicial, Tim Berners-Lee acreditava que seria possível interligar hipertextos em computadores diferentes com uso de links globais também chamados de hiperlinks (SILVA 2011).

Trata-se de uma linguagem de marcação que define a estrutura de elementos que uma página deve ter de modo a fornecer conteúdo iterativo aos usuários. Todavia, a interatividade necessária para a construção de jogos animados em HTML é algo recente, anteriormente só se obtinha com a utilização de ferramentas proprietárias como o Adobe Flash, Microsoft Silverlight e Oracle JavaFX.

No HTML5 esta interatividade é alcançada através da utilização do recurso canvas, que é a tag HTML que permite-se "desenhar" dentro da página.

Atualmente o canvas suporta somente o desenvolvimento 2D, sua implementação 3D está em desenvolvimento e chama-se WebGL. Por consequência do ainda baixo nível de especificação do WebGL, não optamos por o desenvolvimento de um aplicativo 3D.

O HTML5 por fatores como a excelente documentação, grande comunidade de desenvolvedores e usuários, e por último, mas não menos importante, por seu caráter multiplataforma, justifica-se para a construção de jogos. Segundo (KURYANOVICH, 2012) a beleza de desenvolver jogos com o padrão aberto WEB é que este nos delega a escrever uma vez e utilizar em qualquer lugar.

Apesar de a tecnologia ainda não estar completa ela já demonstra grande robustez e os padrões de desenvolvimento invariavelmente estão migrando para a perspectiva HTML5, segundo TABUSCA (2013) desenvolvedores que atualmente trabalham no ramo da Web, já podem visualizar que o novo ramo do desenvolvimento de aplicativos mobile está se aproximando mais e mais à alusiva proposta do HTML5.

### 5.2 MOTORES DE FÍSICA

Motores de física (**Engines de física**) provêm ao software, através de equações matemáticas, um modelo similar das leis da física. Estes motores são geralmente utilizados na construção de games, simuladores entre outros. As bibliotecas de física segundo (SHANKAR, 2012), "geralmente incluem os seguintes recursos: elasticidade, gravidade, fricção e conservação de **momentum** entre dois ou mais objetos que colidem". As bibliotecas de física, segundo (SHANKAR, 2012), geralmente incluem os seguintes recursos: elasticidade, gravidade, fricção e conservação de momentum entre dois ou mais objetos que colidem.

Dentre as bibliotecas mais populares que implementam física, compatíveis com HTML5 constam:

- box2dweb: é um port de sua versão em C++, desenvolvida exclusivamente para ambientes 2D, rica em opções e razoavelmente fácil de utilizar;
- Ammo.js: baseada na biblioteca bullet para física 3D, tem um grande set de funcionalidades, todavia é razoavelmente mais complicada de utilizar;

- JigLibJS: outro port de C++, todavia escrito à mão, com melhorias para o Javascript. Segundo (PRALL), à customização rendeu à JigLibJS extra performance se comparada ao Ammo.js, todavia esta não é tão rica em opções.

### 5.3 SOM E VÍDEO

Atualmente, a maioria dos arquivos de áudio e vídeo rodam através de plugins (como o Adobe Flash). Não obstante, navegadores diferentes podem ter plugins diferentes. O HTML5 define dois novos elementos que especificam o padrão para imbuir áudio e vídeo em páginas Web: `<audio>` e `<video>` (W3SCHOOLS).

### 5.4 ENTRADA DE COMANDOS

Na construção da grande maioria dos jogos é imprescindível alta flexibilidade na gestão de entrada de dados, seja através de teclado, tela sensível, mouse entre outros. O HTML5 trata todos estes casos abstratamente na forma de eventos. Os eventos básicos são: *keydown* (tecla baixa), *keyup* (tecla solta), e *keypress* (tecla pressionada). Basta ao desenvolvedor testar qual caractere gerou o evento em seu laço principal para identificar o que aconteceu.

### 5.5 DETECÇÃO DE RECURSOS

Para detectar suporte aos mais variados recursos do HTML5 no browser do cliente existem duas possibilidades. Pode-se implementar testes para cada funcionalidade utilizada abordando os detalhes de implementação de cada uma ou então fazer uso de alguma biblioteca especializada neste processo, o Modernizr é uma opção open-source deste tipo de biblioteca, este gera uma lista de booleanos sobre grande variedade dos recursos HTML5, dentre estes, geolocalização, canvas, áudio, vídeo e local storage.

### 5.6 TECNOLOGIAS POLYFILL

O HTML5 por não ser um padrão completamente especificado, deixa lacunas de suporte em plataformas, tanto para a gestão de hardware quanto de software. Acarretando assim, que muitos browsers não implementam algumas funcionalidades, completa ou parcialmente especificadas, daí surge a necessidade dos polyfills (tecnologias de preenchimento de lacunas) para implementar estas camadas.

Algumas tecnologias desta classe são:

- Suporte a SVG - Scalable Vector Graphics (vetor de gráficos escaláveis): svgweb, Raphael, canvg, fabric.js;
- Suporte a vídeo: video.js, SublimeVideo, html5media, LeanBack Player;
- Suporte a Geo-localização: Webshims Lib, geolocation polyfill, GeoLocation-API-Polifill;
- Suporte a Web Storage (armazenamento na web): Amplify.js, storage polyfill, session storage;

Uma das soluções mais promissoras polyfill é o PhoneGap ou Apache Cordova, esta ferramenta é open source e possibilita utilizar de inúmeros recursos de hardware da grande maioria das produtoras de dispositivos móveis.

Feature	iPhone /iPhone 3G	iPhone 3GS and newer	 Android 1.0 - 4.2	Windows Phone	 BlackBerry BlackBerry 10 and PlayBook OS	 BlackBerry 4.6-4.7	 BlackBerry 5.x-6.0+	 bada Bada	Symbian	webOS	 TIZEN <sup>®</sup> Tizen
Accelerometer	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes
Compass	N/A	Yes	Yes	Yes	Yes	N/A	N/A	Yes	N/A	Yes	Yes
Contacts	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	N/A	Yes
File	Yes	Yes	Yes	Yes	Yes	N/A	Yes	N/A	N/A	N/A	Yes
Geolocation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Media	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A	Yes
Network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (alert)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (sound)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (vibration)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage	Yes	Yes	Yes	Yes	Yes	N/A	Yes	N/A	Yes	Yes	Yes

Figura 2: Funcionalidades implementadas para o projeto Apache Cordova.

Segundo JÚNIOR utilizando as linguagens de desenvolvimento Web HTML, CSS e Javascript. Ele fornece um conjunto de API's para acesso a funções nativas do Sistema Operacional e do hardware do dispositivo, utilizando Javascript. A proposta do PhoneGap é essencial para unir as especificidades de Web com detalhes de sistemas operacionais tanto de hardware como de software.

## 5.7 AMBIENTES PARA DESENVOLVIMENTO HTML5

Após esta revisão das ferramentas e tecnologias disponíveis, fica claro que a utilização de cada uma delas para a criação de um jogo não é um processo trivial, foi pensando nisso que algumas empresas lançaram frameworks "full stack"(ou framework de pilha completa) para a utilização destes tipos de recursos em mobile, interoperáveis. JEFFRIES (2013) define frameworks full stack relacionados ao desenvolvimento front-end como: frameworks que nos auxiliam no completo desenvolvimento de uma aplicação, desde a interface com o usuário aos dados armazenados.

Na pesquisa efetuada sobre estes frameworks full stack foram identificadas as seguintes tecnologias:

- segundo (PRADO, 2012) o GWT é um framework essencialmente para o lado do cliente (cliente-side) e dá suporte à comunicação com o servidor através de RPCs Remote Procedure Calls (ou procedimento de chamadas remotas). Ele não é um framework para aplicações clássicas da web, pois deixa a implementação da aplicação web parecida com implementações em desktop. Este é utilizado em muitos produtos de grande porte como o Google Adwords e Google Wallet. Outra característica interessante é que a plataforma opera sobre a licença Apache versão 2;
- construct 2 - é um editor na nuvem focado para usuários sem conhecimento prévio em programação orientado a comportamento;

- PlayCanvas - é uma plataformas para a construção de jogos 3D na nuvem, desenvolvida com foco em performance. Permite a hospedagem, controle de versão e publicação dos aplicativos nela criados, possibilita também a importação de modelos 3D de softwares populares como: Maya, 3ds Max e Blender;
- o ambiente HTML5 Development Environment (ambiente de desenvolvimento HTML5) da Intel, este fornece uma solução na nuvem, completa para o desenvolvimento em plataforma cruzada, com serviços de empacotamento, serviços para a criação e testes de aplicativos com montagem de interfaces drag and drop (Intex XDK) e bibliotecas para a construção de jogos utilizando aceleração de hardware, o que garante até duas vezes mais performance que aplicativos mobile baseados em Web tradicionais. Esta solução é free, open source e funciona através de um plugin para o Google Chrome, ou seja, o desenvolvimento também é multiplataforma e devido ao fato de os binários ficarem hospedados na nuvem, possibilitou a Intel criar compiladores para cada uma das plataformas disponibilizadas pelo PhoneGap, que é o framework polyfill utilizado na solução.

## **5.8 FRAMEWORKS PARA DESENVOLVIMENTO DE JOGOS HTML5**

Motores de física, áudio e vídeo são apenas algumas facetas do desenvolvimento de jogos, outros aspectos importantes são: gestão de cenas, câmeras, entidades, recursos, elementos de interface, entre outros. Visto esta grande gama de recursos que um jogo pode ter, foram criados frameworks específicos para jogos em HTML5.

Os mais populares são:

- enchant.js – dentre suas funcionalidades constam: orientação à objetos, orientado à eventos, contém um motor de animação, suporta WebGL e Canvas, etc;
- three.js - considerada leve, renderiza WebGL e Canvas, arquitetura procedural;

## **5.9 EMPACOTADORES**

## **5.10 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE PARA A CONSTRUÇÃO DE GAMES**

Como o jogo é um software complexo demanda-se a utilização de metodologias de engenharia de software, dentre os processos de software mais conhecidos academicamente destacamos:

- OpenUP: este é bem detalhado e de característica iterativa e incremental. Gerando assim, um levantamento mais apurado dos riscos, requisitos e outros detalhes do sistema e a criação incremental do sistema, com requisitos maleáveis;
- Cascata: processo antigo, caracteriza-se por ser pouco maleável aos requisitos mapeados posteriormente ao processo de análise;

- Processo ágil - SCRUM: sua utilização é flexível e sendo um método ágil especifica pouca documentação, ou como dizem, somente a documentação necessária, este processo é bem conhecido e aceito na comunidade de desenvolvimento de software. Suas principais características são: divisão do processo de desenvolvimento através uma série de iterações chamadas sprints. Cada sprint consiste tipicamente em duas a quatro semanas. É bem aplicado a projetos que mudam constantemente e que demandam rápidas adaptações;
- Processo ágil – XP: tem muitas características similares ao SCRUM por este também ser um processo ágil. Dentre suas especificidades destaca-se: versões frequentes, pequenos ciclos de desenvolvimento que buscam aumentar a produtividade, introduzem checkpoints onde os clientes podem agregar novas funcionalidades;

## 5.11 USABILIDADE E TRANSPARÊNCIA

Conceitos de usabilidade, quando relativos ao desenvolvimento multiplataforma, devem levar em questão inúmeros fatores ou “dimensões” como:

- Atores: ex – seus números, interesses, as atividades de cada um;
- Plataformas: ex – tipos de dispositivos, número de dispositivos, seus recursos: tamanho da tela, capacidade de som;
- Ambientes: ex – localização, propriedades de localização: barulho do ambiente, iluminação;
- Recursos do sistema: ex – disponibilidade de acesso à internet, latência de banda, CPU, memória.

Estes contextos variam em todas as dimensões. Por exemplo, um PDA quando segurado em uma mão perto de um dispositivo eletrônico, pode se tornar um controle universal daquele aparelho (PATERNÒ, 2003).

O conceito predominante relativo ao desenvolvimento de aplicativos multiplataforma que leva em consideração a usabilidade é a técnica de desenvolvimento model-based (baseado em modelos), consiste em fornecer suporte a recursos dinâmicos para uma aplicação. Segundo (BERTI, 2003), arquiteturas model-based, podem representar uma solução eficaz para problemas de usabilidade multi-plataforma. Vários sistemas de construção de interfaces a implementam, todavia esta também pode ser utilizada como conceito arquitetural para os mais variados projetos de software. Alguns produtos que utilizam construção baseada em modelos são: Fuse, Humanoid, Mastermind, Mobi-D, Tadeus, Dellach, Teresa e Trident (LUYTEN, 2004).

Em suma a arquitetura é baseada em múltiplos níveis de abstração onde cabe aos desenvolvedores detalharem os requisitos de entrada e saída dos dispositivos, os usuários a definirem suas preferências e então o sistema deve escolher a forma de interação mais apropriada levando em consideração os fatores anteriores.

O HTML5, não provê técnicas para o desenvolvimento baseado em modelos, nativamente, ficando a par do desenvolvedor desenvolver este recurso se assim desejar.

## 5.12 TRABALHOS SIMILARES

O browserquest da Mozilla é um jogo de RPG 8-bits focado em demonstrar na prática a utilização de muitos recursos do HTML5, este trabalho se assemelha razoavelmente ao browserquest; todavia, o browserquest não tem versão estável para a maioria dos dispositivos mobile e também distingue-se por não guardar algumas informações relativas ao estado como o posicionamento, impossibilitando assim, a experiência transparente proposta neste trabalho.

JSWars é um jogo clássico de tiro, escrito especialmente para demonstrar o poder do HTML5 nos navegadores modernos (WAGNER, 2013). O trabalho aborda conceitos interessantes como física, áudio, vídeo e entrada de comandos por teclado, todavia o aplicativo não é interoperável entre dispositivos, nem fornece a perspectiva de continuidade que este trabalho propõe. Para a construção o autor utilizou javascript puro, sem auxílio de nenhuma biblioteca adicional.

Angry Birds é um título de bastante renome, este jogo foi originalmente desenvolvido para IOS, todavia, este foi portado para HTML5. O jogo faz bastante utilização de física, e como motor gráfico, utiliza o WebGL em 2D. Assim como o trabalho anterior, este não aborda o conceito de continuidade de dispositivos, havendo simplesmente uma versão para navegadores, sem utilizar recursos específicos de plataforma (HAWKES, 2013).

(SILVA,2010), demonstra a utilização de HTML5 para a criação de jogos simples, todavia seu trabalho não se foca nas diferenças entre uma plataforma e outra.

## 6 METODOLOGIA

Primeiramente há de ser feita uma pesquisa explanatória - e respectiva documentação - relativo às técnicas, ferramentas e conceitos importantes na construção de jogos 2D. Este tipo de pesquisa foi selecionado pois proporciona uma aproximação do pesquisador com o tema, visando melhor familiaridade com o fenômeno ou assunto (LEMÕES et al., 2012). Neste passo, pretende-se englobar o estudo das tecnologias mencionadas nos objetivos específicos e fazer a seleção das mais adequadas para o desenvolvimento do jogo selecionado.

O segundo passo reside na análise e construção do software em si.

Através de um processo de software iterativo, há de se efetuar a análise e geração de artefatos que melhor mapeiem os requisitos e aspectos do jogo. Depois do período de análise, iniciar-se-á o processo de desenvolvimento, este consistirá em duas etapas subdivididas em iterações. Primeiramente há de se fazer a elaboração de um protótipo, utilizando de imagens e áudio de características livre na internet; após o protótipo criado, pretende-se então construir os modelos reais e integrá-los ao jogo. Com o software desenvolvido, far-se-á a exportação dos binários para as plataformas que suportem aplicativos nativos em HTML5 e disponibilização da aplicação online.

Ao final do desenvolvimento será feito um levantamento textual à respeito dos problemas e acertos da implementação do HTML5 encontrados durante as etapas de construção e pesquisa do projeto. Também há de ser desenvolvido um texto descritivo abordando as diferenças de implementação do controle de movimento e da interface de usuário afim de demonstrar como estes requisitos foram atendidos e quais as customizações necessárias para atendê-los nas diferentes plataformas.

Por fim, há de ser criada uma tabela comparativa a qual exponha as funcionalidades com características dependen-



tes de plataforma e informe se o atendimento da funcionalidade obteve êxito e como se chegou a este resultado, bem como quais outras soluções seriam possíveis. Tendo como fundamento este artefato, se fará uma análise qualitativa dos trunfos e limitações deste tipo de arquitetura para o desenvolvimento de jogos transparentes de plataforma, que oferecem experiências contínuas.

## 7 RESULTADOS

## 8 CONCLUSÃO

## 9 CRONOGRAMA

O cronograma foi especificado de acordo com o detalhado na metodologia, suas datas estão especificadas de acordo com dias úteis disponíveis no calendário.

Identificador	Tarefa	Duração	Início	Término
1	Concepção	5 dias	1 agosto	7 agosto
2	Elaboração	15 dias	8 agosto	29 agosto
3	Construção	15 dias	30 agosto	19 setembro
4	Construção	10 dias	31 agosto	3 outubro
	Total	45 dias	1 agosto	3 outubro

Tabela 1: Cronograma do projeto.

## Referências

- 1 AK, Sheela *Global Gaming Market Is Expected to Reach USD 117.9 Billion by 2015: Transparency Market Research*. Disponível em: <http://www.prnewswire.com/news-releases/global-gaming-market-is-expected-to-reach-usd-1179-billion-by-2015-transparency-market-research-169284526.html> Acesso em: Jul 2012.
- 2 BUCHMAN, Debra D; FUNK, Jeanne B. *VIDEO AND COMPUTER GAMES IN THE "90S: CHILDREN'S TIME COMMITMENT GAME PREFERENCE*. Health and Human Services Department (HHS), 2013.
- 3 KURYANOVICH, Egor; SHALOM Shy, et all. *The State of Open Web Games*. Addison Wesley, Massachusetts, pg. 12, ISBN: 978-1-4302-3978-9, 2012.
- 4 SILVA, Jucimar Maria Júnior; FIRMINO, Emiliano Carlos M. *Desenvolvimento de jogos em HTML5*. Coordenação da engenharia da Computação, Univerisdade Federal do Amazonas, Amazonas, 2010.
- 5 SHANKAR, Aditya Ravi . *Pro HTML5 Games*. ISBN: 978-1-4302-4710-4, p. 39-64, 2012.
- 6 TABUSCA, Alexandru *THE NEW "UNIVERSAL TRUTH" OF THE WORLD WIDE WEB*. American University, School of Computer Science for Business Management, Bucharest, 2013
- 7 PRALL, Chandler *JavaScript Physics Engines Comparison*. Disponível em: <http://www.htmlgoodies.com/html5/client/javascript-physics-engines-comparison.htmlfbid=AAITVDXjb40> Acesso em: Jul 2013.
- 8 VGChartz *Global sales (in millions of units) per game*. Disponível em: <http://www.vgchartz.com/gamedb/> Acesso em: Jul 2012.
- 9 SILVA, Maurício Samy *HTML5 A linguagem EM DE MARCAÇÃO QUE REVOLUCIONOU A WEB*. Editora novatec, p. 15, 2011.
- 10 FRANZINI, Fernando *Nova tendência de aplicativos móveis web*. Disponível em: [\[http://www.infobase.com.br/nova-tendencia-de-aplicativos-moveis-web/\]](http://www.infobase.com.br/nova-tendencia-de-aplicativos-moveis-web/). Acesso em: jun, 2013.
- 11 JEFFRIES, Ron *Full-Stack frameworks vs. Non Full-Stack frameworks*. Disponível em [\[http://codingarchitect.wordpress.com/2012/10/22/full-stack-frameworks-vs-non-full-stack-frameworks/\]](http://codingarchitect.wordpress.com/2012/10/22/full-stack-frameworks-vs-non-full-stack-frameworks/), Acesso em: jun, 2013.
- 12 JÚNIOR, Gesmar de Paula Santos; OLIVEIRA, Luciene Chagas; CARDOSO, Alexandre; LAMOUNIER, Edgard Afonso. *Aplicação Multiplataforma da Realidade Aumentada Móvel para Geolocalização utilizando o PhoneGap*. Programa de Pós Graduação em Engenharia Elétrica Universidade Federal de Uberlândia, rograma de Pós Graduação em Engenharia Elétrica Universidade Federal de Uberlândia, Uberlândia, Minas Gerais.

- 13 PRADO, Ely Fernando *Introdução ao Desenvolvimento de Games com GWT e HTML5*. Departamento de Computação, Universidade Federal de São Carlos (UFSCar) São Carlos, SP, 2012.
- 14 RENYO Emanuel Montero *MODEL-DRIVEN GAME DEVELOPMENT: 2D PLATFORM GAME PROTOTYPING*. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia, España, 2006.
- 15 HERMIDA, Alfred Japan leads mobile game craze. BBC News, 2003. Acesso em: jun 2013.
- 16 *A incrível evolução de videogames de console*. Disponível em: [<http://www.failwars.blog.br/nerd-feelings/incrvel-evolucao-dos-video-games-de-console-de-1967-2012/>] Acesso em: jun, 2013.
- 17 WEINTRAUB, Seth *Industry first: Smartphones pass PCs in sales*. Disponível em: <http://tech.fortune.cnn.com/2011/02/07/idc-smartphone-shipment-numbers-passed-pc-in-q4-2010/> Acesso em: Jul 2012.
- 18 ZICO, Mário Lucio *A História dos Jogos*. Disponível em [<http://www.jogos.antigos.nom.br/artigos.asp>], Acesso em: jun 2013.