

figurethplop figureFigure

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

Jean Carlo Machado

Criação de Jogos Digitais com HTML5

Bento Gonçalves, Julho 2013

Jean Carlo Machado

Criação de Jogos Digitais com HTML5

Trabalho de conclusão de curso

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

Orientador: Dr. Rogério Tessari

Bento Gonçalves, Julho 2013

Este trabalho trata-se de como criar um jogo de plataforma utilizando das tecnologias mais populares da Web, vulgo HTML 5, Javascript e CSS3, o qual seja transparente à plataforma de utilização, mesmo fazendo uso de recursos específicos de hardware ou software. Fêz-se um levantamento das tecnologias existentes e foi proposta uma metodologia para a criação do aplicativo, este tem por requisito funcionar em todos os sistemas operacionais populares, tendo como foco principal a utilização dos recursos oferecidos pelas plataformas *mobile* emergentes.

Palavras chave: multiplataforma, sistema operacional mobile, smartphone, comparativo de tecnologias, jogos.

Lista de Figuras

| | | |
|---|--|----|
| 1 | Funcionalidades implementadas para o projeto Apache Cordova. | 16 |
| 2 | Ciclo de vida do software no OpenUP. | 23 |

Lista de Tabelas

| | | |
|---|--------------------------------|----|
| 1 | Cronograma do projeto. | 24 |
|---|--------------------------------|----|

Conteúdo

| | | |
|----------|---|-----------|
| 1 | CONTEXTUALIZAÇÃO | 5 |
| 1.1 | História e benefícios dos jogos | 5 |
| 1.2 | Tendência mobile | 7 |
| 1.3 | O jogo | 9 |
| 2 | PROBLEMA | 11 |
| 3 | OBJETIVOS | 12 |
| 3.1 | Objetivo primário | 12 |
| 3.2 | Objetivos secundários | 12 |
| 4 | JUSTIFICATIVA | 13 |
| 5 | REVISÃO BIBLIOGRÁFICA | 14 |
| 5.1 | HTML5 | 14 |
| 5.2 | Paradigmas de páginas para dispositivos móveis | 15 |
| 5.3 | Frameworks polyfill | 15 |
| 5.4 | Bibliotecas canvas | 17 |
| 5.5 | Frameworks full stack para front-end | 17 |
| 5.6 | Outras tecnologias | 19 |
| 5.7 | Especificidades do canvas no desenvolvimento de jogos | 20 |
| 5.8 | Processos de software | 20 |
| 5.9 | Trabalhos similares | 20 |

| | | |
|----------|---|-----------|
| 6 | METODOLOGIA | 22 |
| 6.1 | Plataforma de desenvolvimento | 22 |
| 6.2 | Processo de desenvolvimento | 22 |
| 7 | CRONOGRAMA | 24 |
| 8 | REFERÊNCIAS | 25 |

1 CONTEXTUALIZAÇÃO

1.1 História e benefícios dos jogos

Os seres humanos sempre demonstraram interesse por jogos, há referências arqueológicas que indicam a prática de jogos há mais de 5000 anos.

Adeptos da teoria Darwiniana afirmam que foi um jogo chamado de Jogo da Evolução, praticado pelos Neanderthal. Consta que era um jogo bem simples e rude, jogado com um grande osso. Marcava-se pontos destruindo a cabeça dos adversários e com isso conseguindo o domínio de territórios (ZICO, 2008).

Atualmente os jogos são utilizados como um meio de entretenimento, todavia, outros fatores colaterais, em grande maioria benéficos são observáveis na prática de jogar, dentre estes constam: melhorias na capacidade lógica, motora, bem como na resolução de problemas de caráter genérico.

Os jogos na computação surgiram em meados dos anos 60, estes eram extremamente simples e totalmente dependente da plataforma onde eram desenvolvidos, sem mencionar o fato de serem protótipos de laboratório os quais nunca saíram destes locais.

Com a criação da Atari em 1972, os jogos se popularizaram criando um mercado competitivo onde grandes corporações internacionais disputavam entre si para criar as melhores plataformas (consoles) e jogos. Essas corporações dentre as maiores ressalta-se: Nintendo, Sony e Microsoft; devido ao alto teor de competitividade, desenvolveram novas tecnologias, aumentando a iteratividade e imersão do usuário.

Grandes exemplos destas tecnologias são: a criação de renderizadores de elementos de 3 dimensões (3D), bibliotecas de física newtoniana, grandes produções

musicais de tema para 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.

Segundo (BUCHMAN,1996)

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.

Porém juntamente com a evolução das tecnologias aumentou também a dificuldade de se construir jogos de caráter comercial. Quando à priori era necessário uma equipe de 5 a 6 pessoas durante uma média de 6 meses para construir um jogo diga-se popular, hoje faz-se necessário período de 3 a 4 anos com equipes de 500 a 1000 pessoas, como é o caso do GTA 4 que levou 4 anos para ser desenvolvido com uma equipe de aproximadamente 1000 pessoas.

(RENYO, 2006) comenta que:

Nas últimas décadas o desenvolvimento de jogos cresceu exponencialmente em complexidade, testava-se a importância de criar-se métodos para melhorar este processo.

À partir dos anos 80, com a popularização da plataforma PC, decorrente do barateamento do custo de fabricação de circuitos integrados e da criação de grades softwares como sistemas operacionais e aplicativos office, os jogos de console perderam parcela de mercado, todavia não deixaram de existir. Este declínio de mercado ocorreu porque com um PC de aproximadamente 200U\$ obtinha-se uma plataforma que além de jogar possibilitava a realização de inúmeras outras tarefas como estudar e trabalhar, possibilitando assim, um grande benefício em relação aos 150U\$ em média gastos com consoles que serviam exclusivamente

para jogos.

1.2 Tendência mobile

Recentemente, devido barateamento e miniaturização dos componentes, tendência já prevista por Moore em 1965, bem como a massificação da internet, surge um novo mercado, o de aplicativos mobile. Torna-se assim possível construir celulares, palmtops e tablets com a arquitetura muito semelhante às dos PC's, inclusive rodando os mesmos sistemas operacionais, que quando não idênticos sofrem poucas customizações. Isso permite a migração de se não todos, a grande maioria dos aplicativos desenvolvidos para os computadores de mesa para os dispositivos mobile, agregando além de seus benefícios primevos, os trunfos da mobilidade e onipresença que este novo paradigma permite.

Contudo, com a grande popularização deste tipo de arquitetura, cada vez mais novos sistemas operacionais para smartphones e aparelhos semelhantes fazem-se disponíveis, cada qual com particularidades tais como linguagens suportadas, recursos mínimos e aparelhos compatíveis, são exemplos de sistemas operacionais mobile: Android, FirefoxOS, Ubuntu, Windows Phone e o ainda não lançado Taizen. Sob esta perspectiva os desenvolvedores sentem cada vez mais o desafio que é construir aplicativos com abrangência suficiente de plataforma, de modo a alcançar todos ou a grande maioria de seus usuários.

Para tratar deste problema algumas soluções já foram propostas dentre as quais cito:

- utilização de linguagens de programação rodando máquinas virtuais (teoricamente independente de plataformas) como o Java;
- utilização da linguagem amplamente utilizada como padrão para o desen-

volvimento como no caso o HTML em especial sua nova especificação o HTML5;

Sobre a utilização de linguagens interpretadas ou parcialmente compiladas rodando em máquinas virtuais, nota-se problemas, pois estas tecnologias nem sempre estão disponíveis em todas as plataformas e os aplicativos são submetidos à temas e outros detalhes de arquitetura específicos de plataforma para plataforma.

A alternativa da utilização da linguagem HTML5, é bem mais promissora, pois em sua nova especificação o HTML5 não trata mais somente de hipertextos trafegando pela rede, serve também para desenhar *on the fly* (ou dinamicamente), permite também rodar em modo offline, implementa semântica e acessibilidade, entre outros recursos, na criação de aplicativos Web de estilo Desktop.

Assim como as maiores tecnologias concorrentes o Adobe Flash e o Microsoft Silverlight o HTML5 também não está totalmente difundido nas plataformas mobile, todavia suas perspectivas são interessantes, estima-se que para o ano de 2016 85% dos navegadores estarão utilizando esta tecnologia. HTML5 não é somente um objetivo de mercado, outra pesquisa aponta que 2/3 dos desenvolvedores de software manifestam interesse em criar aplicações com a tecnologia, o que aponta a linguagem como uma ferramenta bastante promissora nos anos vindouros.

Contudo, o HTML trata apenas da estruturação e disposição do conteúdo, para a criação de aplicativos completos é necessária a utilização de scripts os quais implementam a lógica da funcionalidade e de alguma tecnologia para a customização da aparência do aplicativo. As soluções mais populares para estas demandas são o Javascript e o CSS (*Cascading Style Sheets*), respectivamente, que serão tratados mais adiante.

Os jogos, não são a exceção em representatividade nas plataformas mobile. Se-

gundo (HERMIDA, 2003) o total de lucros obtidos com a venda de jogos mobile ultrapassou 5,8 bilhões.

Devido à intrínseca complexidade de arquitetura dos jogos, estes tendem a ser um desafio ainda maior no quesito portabilidade. O problema se agrava quando os recursos de hardware ou software específicos de plataforma, como o acelerômetro, câmera, GPS, lista de contatos, etc, cuja implementação de soluções de software é extremamente dependente de arquitetura forem utilizados na construção dos jogos.

Quando tratando destas especificidades de plataforma, muitas vezes não é possível utilizar somente o HTML5, pois sua especificação ainda não está completamente fechada outrossim muitos browsers ainda não implementam estas especificações, demanda-se então a utilização de algum *polyfill* (ou *polyfilter*), que nada mais é que uma API *Application Programming Interface* (ou Interface de Programação de Aplicativos) para a utilização de recursos que normalmente não estão disponíveis para ferramentas Web.

Sob este prisma o trabalho a ser desenvolvido propõe a construção de um jogo, desenvolvido em HTML5, Javascript, CSS3 e alguma ferramenta polyfil (caso haja necessidade), que possibilite a criação bem como a utilização linear em todas as plataformas populares, fazendo uso do acelerômetro como recurso de hardware específico de plataforma.

1.3 O jogo

A criação do jogo não é o foco principal do trabalho, todavia afim de ilustrar os conceitos abordados, propõe-se um jogo de plataforma 2D, no estilo de Mario, onde o jogador terá de passar por uma série de desafios para conseguir sair de cada fase. Sendo composto na sua totalidade de 3 fases com dificuldade incremental.

Caberá ao jogador, conduzir seu personagem através do cenário, de modo que o mesmo sobreviva aos desafios propostos, caso o personagem venha a morrer o usuário deverá recomeçar o jogo. O acelerômetro será utilizado como acessório para alterar os eixos do jogo, sendo que, caso o jogador vire o dispositivo, o jogo deve "virar" também, derrubando os personagens e alterando cenário.

Cabe ressaltar que a funcionalidade relativa ao acelerômetro serve para elucidar um componente específico de plataforma o qual deverá ser suportado.

2 PROBLEMA

A dificuldade de criar jogos, fornecendo uma experiência de alta similaridade de utilização por arquitetura, que suporte as plataformas de maior popularidade, utilizando um único set de tecnologias, fazendo uso de recursos específicos de plataforma.

3 OBJETIVOS

3.1 Objetivo primário

Em suma este trabalho tem por objetivo principal a criação de um jogo de plataforma, de presença ubíqua, ou seja, que rode em todas plataformas mobile de maior popularidade, vulgo Android, IOS, Windows Phone e FirefoxOS, suportando também as funcionalidades independentes de hardware para os sistemas operacionais de computadores de mesa diga-se, Linux, Windows e Mac; e que forneça uma experiência de usuário o mais linear possível. O foco do trabalho reside na prova da factibilidade do conceito e não na construção do jogo em si.

3.2 Objetivos secundários

Dentre os objetivos de caráter secundários constam:

- Desenvolver o aplicativo utilizando somente de tecnologias open source;
- Deseja-se também identificar os pontos relevantes (fraquezas e acertos) que a implementação atual do HTML5, nos diferentes tipos de plataforma, dessa forma sendo possível obter uma visão panorâmica de o quanto falta para o HTML5 estar plenamente estabelecido;
- Objetiva-se também demonstrar os conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas, abordando às áreas de programação, desenvolvimento de sistemas, interação humano computador, engenharia de software entre outros;

4 JUSTIFICATIVA

O trabalho se justifica, por fornecer um material de *proof-of-concept* (ou prova de conceito) sobre a factibilidade da construção de jogos de experiência linear em múltiplas plataformas utilizando as tecnologias emergentes, cujo conteúdo é até então, relativamente pouco explorado em trabalhos acadêmicos.

Servirá também para demonstrar o estado da arte do HTML5 e os contornos necessários, para se desenvolver um jogo completo, multiplataforma fazendo-lhe uso.

Considerando à popularidade destas tecnologias e suas perspectivas o trabalho também poderá auxiliar desenvolvedores, a selecionar um conjunto de tecnologias os quais poderão trabalhar com maior facilidade em meio ao ecossistema de soluções existentes e não cair nos erros que este tipo de tecnologia pioneira ainda pode pregar.

Vários trabalhos tratam da utilização de HTML5 na criação de jogos, todavia, este trabalho se diferencia pela abordagem da criação de jogos, na perspectiva das diferenças de cada plataforma e na procura de uma solução que permita tanto o desenvolvimento quanto uma utilização horizontalizada.

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), todavia a iteratividade necessária para a construção de jogos animados em HTML é algo recente, anteriormente só se obtinha com a utilização de ferramentas de terceiros como o Adobe Flash.

No HTML esta iteratividade é alcançada através da utilização do recurso canvas, que é a tag HTML que permite-se "desenhar" dentro da página. Para testar se um navegador possui suporte ou não, basta renderizar uma página HTML utilizando este recurso, caso o navegador ainda não suporte a tecnologia nada será mostrado, caso contrário a tag aparecerá, e poderá ser manipulada através de Javascript.

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.

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 elusiva proposta do HTML5.

5.2 Paradigmas de páginas para dispositivos móveis

Até o momento, existem duas formas de desenvolver aplicativos mobile com HTML5, são elas:

- **Páginas convencionais:** são páginas HTML com otimizações para esta categoria de dispositivos, como poucas ou nenhuma imagem, possivelmente com resolução ajustável, e foco em simplicidade.
- **Páginas com aparência nativa:** São aquelas páginas que são construídas aparentando serem interfaces gráficas desktop da própria plataforma móvel. Ou seja, as páginas "aparentam ser aplicações nativas, utilizando todos os componentes específicos e a forma de usabilidade totalmente voltada para os monitores otimizados" (FRANZINI, 2013).

Percebe-se que o paradigma das páginas de aparência nativa agregam muito mais valor ao aplicativo, devido a fatores como usabilidade pois a aplicação mantém uma experiência homogênea de navegação, outro fator relevante é o preconceito de se pensar que páginas Web não podem fornecer tanta interatividade quanto aplicativos nativos.

Para criar páginas Web de aparência nativa existem alguns frameworks como o RichFaces Mobile, PrimeFaces Mobile, JQuery Mobile e Dojo Mobile. Estas bibliotecas, nada mais são que páginas HTML puras as quais cabe ao CSS e JavaScript, simular as funcionalidades do dispositivo.

5.3 Frameworks polyfill

Como dito anteriormente, o HTML5 por não ser um padrão completamente especificado, deixa lacunas de suporte em plataformas, para recursos específicos tanto para a gestão de hardware quanto de software, acarretando que muitos browsers não implementam algumas funcionalidades, completa ou parcialmente

especificadas, daí surge a necessidade dos polyfills para implementar estas camadas.

Algumas tecnologias desta classe são:

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

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 10 and PlayBook OS |  BlackBerry 4.6-4.7 |  BlackBerry 5.x-6.0+ |  bada Bada |  Symbian |  webOS |  TIZEN 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 1: Funcionalidades implementadas para o projeto Apache Cordova.

Segundo (JÚNIOR) utilizando as linguagens de desenvolvimento Web HTML, CSS e Javascript. Ele fornece um conjunto de APIs 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.

Todavia, um fator limitante do PhoneGap é que o mesmo compila somente para a plataforma que está rodando o aplicativo, limitando assim a geração dos binários para as outras plataformas.

5.4 Bibliotecas canvas

Em desenvolvimento de jogos para Web, outro fator que agrega complexidade é que a utilização de canvas puro para a criação de jogos é um tanto complexa, e demanda muito tempo devido ao grande número variáveis que devem ser contabilizadas, nestes casos, é recomendável a utilização de bibliotecas que implementem recursos adicionais, como motores de física com detecção de colisões e encontro de forças, grafos de cenas, animações, etc.

São diversas as soluções que propõem soluções para este tipo de problema, dentre elas cita-se: Paper.js, Processing.js, jCanvasScript, FabricJs, etc.

5.5 Frameworks full stack para front-end

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, 2012) 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.
- o ambiente *HTML5 Development Environment* (ou 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 multi-plataforma 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.

Em uma análise das tecnologias "full stack" disponíveis a fim de viabilizar a construção de jogos mobile fica claro que a melhor opção encontrada é o ambiente de desenvolvimento HTML5 oferecido pela Intel. Este se encaixa no quesito principal, que é uma solução para auxiliar na criação de jogos mobile multiplataforma, também compreende imediatamente um dos objetivos secundários por sua característica livre, sem mencionar a possibilidade de compilar os programas, bem como debugá-los na nuvem, onde são passíveis de serem exporados em sua

versão final.

O desenvolvimento com GWT, não oferece bibliotecas específicas para jogos, bem como não oferece uma API de desenvolvimento nativa em HTML5 sendo estes seus maiores fatores limitacionais neste projeto.

5.6 Outras tecnologias

Nesta seção são abordadas outras tecnologias necessárias para a concretização do projeto proposto.

Javascript

O Javascript é uma linguagem de script, criada pela Netscape, tendo sua primeira aparição em 1995, e atualmente uma das linguagens de propósito genérico mais populares do mundo, interpretada em praticamente todos os navegadores atuais, este há de ser utilizado para a criação da lógica iterativa do jogo.

CSS

Já o CSS, principalmente em sua versão 3 permite a estilização de páginas de marcação como o HTML e XML em todos os aspectos de desing e layout, seu principal benefício é permitir a separação entre o formato e o conteúdo do documento. Este recurso há de ser utilizado para implementar a estilização do jogo.

As tecnologias acima mencionadas (HTML5, Javascript e CSS) são especificadas por uma organização a W3C *World Wide Web Consortium*, a qual consiste em um consórcio internacional de 400 membros dos mais diversos tipos de entidades, que tem por objetivo estabelecer os padrões de desenvolvimento e tecnologias utilizadas para a construção de conteúdo para a Web.

5.7 Especificidades do canvas no desenvolvimento de jogos

Um ponto bastante mencionado na bibliografia é que para a criação de jogos fluentes usualmente recomenda-se a utilização da técnica de *double buffering*, segundo (SILVA, 2010) ela consiste em criar um canvas secundário, desenhar nele e depois copiar as variáveis para o canvas principal. Evitando assim que o jogador perceba a tela sendo desenhada em cada frame.

5.8 Processos de software

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.

5.9 Trabalhos similares

(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.

Este trabalho pr em a utiliza  o do intel XDK para o desenvolvimento de jogos mobile multiplataforma que utilizam HTML5, possibilitando assim uma revis  o sobre esta tecnologia para a cria  o de aplica   es, em espec  fico jogos. Esta pesquisa n o conseguiu identificar trabalhos que englobem todos estes detalhes acima dispostos.

6 METODOLOGIA

Primeiramente há de ser criado uma versão demo utilizando de imagens e audio de características livre na internet, após o demo criado, pretende-se então construir os modelos reais, bem como selecionar as faixas musicais finais.

6.1 Plataforma de desenvolvimento

Como ambiente de desenvolvimento escolhido utilizar-se-á o XDK da intel, que é um, este ambiente proporciona o desenvolvimento na nuvem e internamente contém um compilador para cada plataforma necessária o que facilita na geração e distribuição de binários. O SDK é atraente também por conter otimizações de software para aumentar o desempenho em jogos.

Como dito anteriormente o trata-se SDK de um plugin do Google Chrome, então com a liberdade de plataforma proporcionada por esta tecnologia, optou-se pelo sistema operacional Linux para realizar o desenvolvimento do software.

6.2 Processo de desenvolvimento

Como processo de desenvolvimento foi escolhido o OpenUp, justamente por este ser incremental e possuir uma alta gama de artefatos, características aprováveis para o desenvolvimento de algo complexo como um jogo.

O Open Up define 4 estágios para o desenvolvimento de software, são elas: **concepção, elaboração, construção e transição**, abaixo estas estão descritas de acordo com o escopo do projeto.

- Concepção: Nesta fase será feita uma análise nos riscos e requisitos do trabalho proposto bem como dar-se-á início ao GDD *Game Document Design* (ou documento de design do jogo);

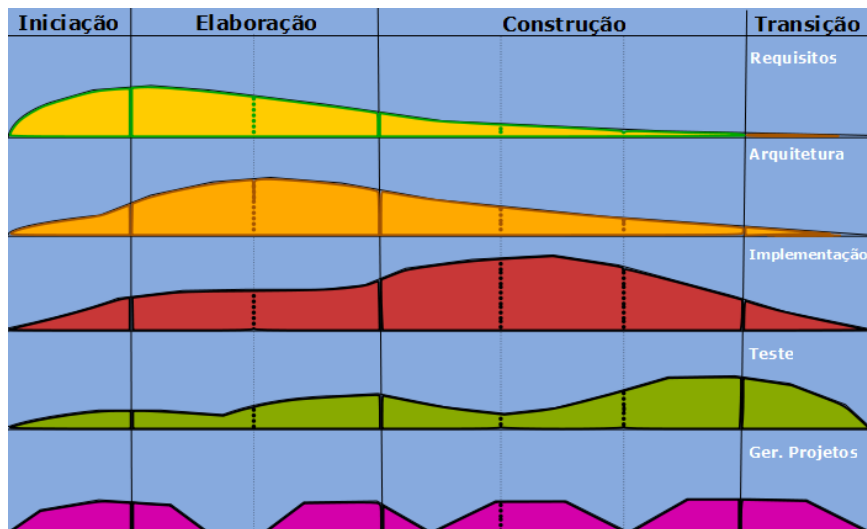


Figura 2: Ciclo de vida do software no OpenUP.

- Elaboração: Esta fase detalhará o funcionamento arquitetural do aplicativo, e de ser concluída a construção do GDD.
- Contrução: É neste estágio que a criação do aplicativo acontecerá, em um primeiro momento da forma de um demo, e posteriormente como uma aplicação completa.
- Transição: Baterias de testes e a documentação dos resultados será desenvolvido neste estágio.

7 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.

8 REFERÊNCIAS

SILVA, Jucimar Maria Júnior; FIRMINO, Emiliano Carlos M. Desenvolvimento de jogos em HTML5. Coordenação da engenharia da Computação, Univerisdade Federal do Amazonas, Amazonas, 2010.

SILVA, Maurício Samy. HTML5 A linguagemEM DE MARCAÇÃO QUE REVOLUCIONOU A WEB. Editora novatec, p. 15, 2011.

FRANZINI, Fernando .Nova tendência de aplicativos móveis web . Disponível em: [<http://www.infobase.com.br/nova-tendencia-de-aplicativos-moveis-web/>]. Acesso em: jun, 2013.

TABUSCA, Alexandru THE NEW “UNIVERSAL TRUTH” OF THE WORLD WIDE WEB. American University, School of Computer Science for Business Management, Bucharest, 2013

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.

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.

RENYO Emanuel Montero MODEL-DRIVEN GAME DEVELOPMENT: 2D PLATFORM GAME PROTOTYPING. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia, España, 2006.

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.

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/>], Acesso em: jun, 2013.

ZICO, Mário Lucio A História dos Jogos. Disponível em [<http://www.jogos.antigos.nom.br/artigos.asp>], Acesso em: jun 2013.

HERMIDA, Alfred, Japan leads mobile game craze. BBC News, 2003. Acesso em: jun 2013.

A incrível evolução de vídeo games 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.