

1 CONTEXTUALIZAO

1.1 JOGOS

1.1.1 BENEFICIOS

Random citation [1] So variadas as pesquisas que apontam os benefcios dos jogos eletrnicos para as pessoas, coordenao motora, maior concentrao.

Muitos jogos ajudam a desenvolver habilidades prticas, servem como uma forma de exerccio, ou de alguma forma executam um papel educacional, de simulao ou psicolgico. Jogos so uma parte universal da experincia humana, presentes em todas as culturas.

1.1.2 O MERCADO

No incio os jogos nativos dominavam... Estatsticas sobre os jogos... A muito a indstria de jogos superou a cinematogrfica...

Com a ascenso dos dispositivos inteligentes a massiva quantidade de dispositivos se tornou atraente para a indstria e consumidores de jogos.

A maior dificuldade em capturar uma base de usurios que o mercado de dispositivos mveis muito fragmentado e no existe uma nica plataforma popular. (HASAN, 2012)

1.1.3 JOGOS E MULTIPLATAFORMA

laboriosa a tarefa dos produtores de software em um panorama to diversificado como o atual, existem muitas plataformas, muitas verses e hardwares diferenciados.

Uma alternativa para mitigar os problemas oriundos da multiplicidade de plataformas o HTML.

1.1.4 HTML E MULTIPLATAFORMA

Desenvolvedores de jogos web podem rapidamente satisfazer as necessidades de seus jogadores, mantendo-os leais a tecnologia HTML5 (ZHANG, 2012).

¿A maioria dos desenvolvedores demonstra interesse para o HTML5

Entretanto, o HTML em sua especificao e implementaes atuais costa com algumas limitaes que precisam ser compreendidas por aqueles interessados em criar jogos em HTML5.

¿ O tempo de desenvolvimento de uma aplicao em HTML5 67que aplicaes nativas. Isso mostra o custo efetivo de aplicaes baseadas em HTML5. A real vantagem de aplicaes em HTML5 o suporte horizontal entre as plataformas - que a maior razo por trs do custo efetivo. (HASAN et al, 2012)

1.1.5 LIMITAES DE JOGOS MULTIPLATAFORMA COM HTML5

O HTML vem sendo desenvolvido por muitos anos e por pessoas que no conheciam umas as outras, muitas funcionalidades foram construdas de maneiras inconsistentes.

¿ Funcionalidades foram disponibilizadas de diversas fontes e no foram construdas de forma especialmente consistente com as demais. Alm disso, devida a nica caracterstica da Web, erros de implementao se tornam frequentes, e muitas vezes se tornam o padro, pois outras funcionalidades dependem destas primeiras antes que elas estejam estveis. (W3C manual)

Enquanto o HTML desenvolvido muitas das funcionalidades disponibilizadas so testadas em apenas um pequeno conjunto de navegadores para um pequeno conjunto de verses (referncia 2). Isso acarreta em suporte inconsistente. A forma mais segura de garantir suporte testando em todas as verses alvo, todavia essa soluo no prtica. (ref. 2)

Os desenvolvedores de navegadores podem interpretar/implementar as especificaes erroneamente aumentando os problemas de compatibilidade.

Nem todos os recursos disponveis atravs das SDK's nativas esto presentes atravs do HTML5.

1.2 ESTE TRABALHO

Este projeto prope analisar as limitaes do HTML5 quanto relativo a construo de jogos multi-plataforma. Atravs de reviso bibliogrfica e da criao de um prottipo de jogo multiplataforma.

Um tratado completo sobre o assunto requiriria um comparativo entre jogos desenvolvidos nativamente e jogos em HTML5.

No objetivo deste trabalho demonstrar onde o HTML5 se sobressai, apenas suas limitaes. Tambm no objetivo deste trabalho comparar o HTML com outras tecnologias de desenvolvimento de jogos, como Flash Player, Silverlight ou alternativas Desktop.

1.2.1 O JOGO

Para a anlise prtica das limitaes foi escolhido um jogo de matemtica simples. Consistindo na gerao de equaes com um candidato de resposta. Cabe ao usurio informar se o resultado apontado pelo jogo est correto ou no.

Porqu escolhi esse tipo de jogo?

2 PROBLEMA

A carncia de definies concretas sobre a viabilidade da atual verso do HTML5 - quando utilizado no desenvolvimento de jogos e o senso comum, acabam por monopolizar construo de jogos nativos as plataformas alvo.

Os custos introduzidos no ciclo vida de um jogo, para diversas plataformas, muito alto para ser considerado trivial. Cerca de 65% mais altos (segundo trabalho 2)

2.1 OBJETIVOS

Abaixo seguem os objetivos deste trabalho.

2.1.1 OBJETIVO GERAL

Identificar possveis limitaes no processo de desenvolvimento de jogos multiplataforma oriundas do atual estado de definio e implementao do HTML5.

2.1.2 OBJETIVOS ESPECIFICOS

Estudar as limitaes de desenvolvimento de jogos nas plataformas de dispositivos inteligentes Android e navegadores Desktop Google Chrome 42 e Firefox 37. Optamos por Android, e no IOS, pois o primeiro contm a vasta maioria do mercado de dispositivos inteligentes, e por termos maior experincia na j mencionada plataforma.

Pretende-se tambm estudar os seguintes tpicos do desenvolvimento de jogos, relativos ao HTML5:

- Depurao - Diferenas em tamanho de tela - Canvas 1. Troca de tamanhos via Canvas vs DOM 2. Acelerao de GPU 3. API de udio (referncia 2) - Performance - Empacotadores HTML5 - Eventos de entrada - Vibrao - Acelermetro - Armazenamento - Disponibilizao de assets (controle de tamanhos, cache, etc) - Aplicaes offline - CSS media queries

Elaborar uma lista de limitaes e correlacionar os dados de acordo com as plataformas.

¡!- Pensamento: talvez seja interessante concluir se vivel produzir jogos com HTML5 do ponto de vista mercadolgico -¿

3 JUSTIFICATIVA

Tendo em vista que este trabalho busca mapear possíveis problemas do desenvolvimento multiplataforma em HTML ele serve para apoiar e justificar decisões relativas ao desenvolvimento de jogos multiplataforma; Por tratar cientificamente de aspectos importantes do HTML, este trabalho tem potencial apontar os pontos chave que necessitam de melhorias nas plataformas alvo, colateralmente colaborando para a melhoria do próprio HTML. A opinião comum tende para soluções nativas em detrimento do desenvolvimento de jogos, este trabalho pretende desafiar esta concepção. (REFERENCIAR) Muitos desenvolvedores estão familiarizados com as tecnologias da WEB ou apontam interesse na tecnologia. ¡!– referenciar –! Estimular e avançar o estudo da implementação da Open Web;

4 REVISÃO BIBLIOGRÁFICA

4.1 JOGOS

Segundo LEMES (2009, pg. 126) o jogo digital constitui-se em uma atividade lúdica composta por uma série de ações e decisões, limitada por regras e pelo universo do game, que resultam em uma condição final.

Essa característica interativa é a dependência comandos sobre uma interface digital, que faz com que o projeto digital desta natureza não seja um filme ou uma animação, e sim um game.

Quando desenvolvendo qualquer jogo, o desenvolvedor tem que considerar seu usuário. O objetivo é maximizar a satisfação de seu usuário. Jogos em plataformas móveis trazem um novo conjunto de desafios para produtores de jogos. Um destes desafios é fornecer feedback suficiente para o player pois o dispositivo é limitado em proporções, som, tela etc. Os jogos multiplataforma em HTML5 têm a dificuldade adicional de ter que comportar, na mesma base, o feedback adequando para cada plataforma móvel.

A interface tem que ser o mais intuitiva possível. No caso de dispositivos móveis, quanto menos gestos necessários melhor. Tornar previsível causa e efeito é uma boa característica para os jogos. Os desenvolvedores tem que evitar fazer o jogo para eles mesmos. E pela falta de crítica os designs tendem a ser ruins. Afinal o que os jogadores querem? LEMES (2009, pg XX) aponta alguns fatores procurados pelos usuários de jogos: Desafio, socializar, experiência solitária, respeito e fantasia.

Mencionar algum jogo (como WOW) e como ele faz para prender a atenção dos usuários. Candy Crush saga

4.1.1 GÊNEROS

LEMES (2009, p. 43) aponta os seguintes gêneros de jogos.

- Adventure Ao RPG Estratégia Simuladores Esportes Luta Casuais - 'God' Games Educativos Puzzle Online / Massive Multiplayer

4.1.2 MECÂNICA

A mecânica é composta pelas regras do jogo. Quais as ações disponíveis aos usuários, fortemente influenciada pela categoria do jogo em questão.

4.1.3 ARQUITETURA

Existem algumas estratégias relativas às plataformas alvo de como efetuar construção de jogos.

4.1.4 DESENVOLVIMENTO DE JOGOS NATIVOS

Habilita a melhor experiência de usuário pois permite utilizar ao máximo os recursos e funcionalidades dos aparelhos. Porém, devido a cada plataforma conter seu próprio sistema operacional, com seus próprios *SDK's* totalmente incompatíveis, os desenvolvedores são forçados a desenvolver uma versão do jogo para cada plataforma alvo. Além da replicação dos fontes, esta abordagem requer mais pessoas, e maior custo com possivelmente parte do mercado não atendido de qualquer forma.

4.1.5 DESENVOLVIMENTO DE JOGOS WEB

Necessitam de apenas uma base de código e pode rodar em todas as plataformas. Contm a mais vasta gama de desenvolvedores e muitos interessados em aprender-lá. Seus custos também são inferiores, aos do desenvolvimento nativo pois demandam menos trabalhadores/hora devido a inexistência de duplicação da base. No obstante, esta opção devido a incompletude da especificação de padrões carece de alguns recursos e outros não estão completamente implementados. Performance também pode ser um limitador, visto que estas tecnologias são executadas através de um navegador, criando uma camada de abstração superior das APIs nativas que fazem chamadas ao sistema diretamente.

4.1.6 DESENVOLVIMENTO DE JOGOS HÍBRIDOS

Jogos híbridos são jogos geralmente desenvolvidos com tecnologias da . web: beneficiando-se da não necessidade de duplicação. Rodam dentro . de um *container* nativo possibilitando o acesso chamadas do . sistema, recursos de hardware, eliminando muitas das dificuldades da . web. Em certo sentido, beneficiam-se do melhor de ambas as metodologias . anteriores. Phone game é uma ferramenta deste tipo. Permite acessar . os dispositivos utilizando sua API JavaScript. Funciona encapsulando . todo o código HTML5. Este tipo de abordagem permite acessar câmera, . acelerômetro, GPS, etc .

4.2 WEB

4.2.1 OPEN WEB

Mais do que um conjunto de tecnologias Open Web, é um conjunto de filosofias as quais a web se baseia.

Neste conjunto inclui-se:

- Descentralização; - Transparência; - Relevância; - Imparcialidade; - Consenso; - Disponibilidade;
 - Manutibilidade;
- !- Ref -;

A OWP (Open Web Platform), uma coleção de tecnologias livres, amplamente utilizadas e padronizadas as quais adotam a postura da Open Web. Quando uma tecnologia se torna amplamente popular, através da adoção de grandes empresas e desenvolvedores ela se torna candidata a adoção pela OWP.

A tecnologia chave que inaugurou e alavancou este processo é o HTML.

4.3 HTML

Hyper Text Markup Language (HTML) é uma linguagem de marcação que define a estrutura semântica do conteúdo das páginas da web, criada por Tim Berners Lee e oficialmente mantida pela W3C. O "Hyper Text" refere-se a links que conectam páginas umas às outras, fazendo a Web como conhecemos hoje (MDN, 2015). !- <https://developer.mozilla.org/en-US/docs/Web/HTML> -
; HTML foi especificado baseado-se em SGML *Standard Generalized Markup Language*, abraçando assim suas premissas:

- Deve ser declarativo, descrevendo estrutura e outros atributos, - Deve ser ao invés de definir o processamento a ser efetuado no - Deve ser documento: rigoroso de modo que as mesmas técnicas de - Deve ser mineração de dados em objetos e bancos de dados possam ser - Deve ser utilizadas;

No obstante, os criadores de navegadores constantemente introduziam elementos de apresentação com o `*blink*`, `*i*` itálico, `*b*` bold, que eventualmente acabavam por serem incluídos na especificação. Foi somente nas últimas versões que elementos de apresentação voltaram a ser proibidos reforçando as propostas chave HTML como uma linguagem de conteúdo semântico, abrindo espaço para a expansão de outras tecnologias como o CSS para responder as demandas de apresentação.

A atual versão do HTML o HTML5, desenvolvido como um trabalho em conjunto entre a WHATWG e a W3C, seu rascunho foi proposto em 2008 e apenas ratificado em 2014. O HTML5 introduziu elementos interativos que viabilizaram a construção de jogos para a plataforma como: Canvas, áudio, vídeo.

Cada elemento HTML informa algo ao navegador sobre a informação que reside entre o abrir e fechar da tag. `!- ducket pg 20 -`

Um elemento o abrir fechar de uma tag e todo o conteúdo que dentro dele reside.`!- ducket pg 24 -`

O HTML5 muitas vezes interpretado como um conceito guarda chuva para designar as tecnologias da web HTML, JavaScript e CSS3.

4.4 CSS

uma linguagem de folhas de estilo, criada por Håkon Wium Lie em 1994, com intuito de definir a apresentação de páginas HTML.

`! Possibilitam ligação tardia *late biding*`. Essa característica atrativa para os publicadores por dois motivos. Primeiramente pois permite o mesmo estilo em várias publicações, segundo pois os publicadores podem focar-se no conteúdo. Lie

O CSS dividido em módulos, contendo aproximadamente 50 deles, cada qual evoluindo separadamente.

Sua última versão, o CSS3, introduziu várias funcionalidades relevantes para jogos, como `*media-queries*`: possibilitam regras para tamanhos de tela, transformações 3D e animações.

Os navegadores interpretam CSS através da tag `<style>`.

`!- Cascading Style Sheets, PhD thesis, by Håkon Wium Lie provides an authoritative historical reference of CSS pg. 23 -`

`!- Muitos navegadores também suportam aceleração de GPU (Unidade de processamento gráfico) para elementos que tenham transformações 3D. -`

Flow de documento, ordem e posição em que os elementos tem que aparecer na página. Modelo de caixa o que encapsula o conteúdo em um elemento. `-`

`!- falar do suporte a variáveis do CSS -`

4.5 JAVASCRIPT

ECMAScript, melhor conhecido como JavaScript, criada por Brendan Eich em 1992, a linguagem da Web. Devido a tremenda popularidade entre comunidade de desenvolvedores a linguagem foi abraçada pela W3C e atualmente um dos componentes da `*Open Web Platform*`.

As definições da linguagem são descritas na especificação ECMA-262. Esta possibilitou o desenvolvimento de outras implementações além da original - `*SpiderMonkey*` - como o Rhino, V8 e TraceMonkey; bem como outras linguagens similares como JScript da Microsoft e o ActionScript da Adobe.

JavaScript uma linguagem de script. Segundo a Ecma Internacional 2012:

`! "Uma linguagem de script uma linguagem de programação que usada para manipular e automatizar os recursos presentes em um dado sistema. Nesses sistemas funcionalidades j esto`

disponíveis através de uma interface de usuário, uma linguagem de script e um mecanismo para expor essas funcionalidades para um programa protocolado.”

A intenção original era utilizar o JavaScript para dar suporte aos já bem estabelecidos recursos do HTML, como para validação, alteração de estado de elementos, etc. Em outras palavras, a utilização do JavaScript era opcional e as páginas da web deveriam continuar operantes sem a presença da linguagem.

Entretanto, com a construção de projetos Web cada vez mais complexos, as responsabilidades delegadas ao JavaScript aumentaram a ponto que a grande maioria dos sistemas web não funcionarem sem ele. No obstante, JavaScript não evoluiu ao passo da demanda e muitas vezes carece de definições expressivas, completude teórica, e outras características de linguagens de programação mais bem estabelecidas, como o C++ ou Java (Barnett, 2013). A nova versão do JavaScript, o JavaScript 6, um esforço nessa direção. JavaScript 6 ou *EMAScript Harmonia*, contempla vários conceitos de orientação a objetos como classes, interfaces, herança, tipos, etc.

Estes esforços de padronização muitas vezes não são rápidos o suficiente para produtores de software web, demora-se muito para obter-se um consenso sobre quais as funcionalidades desejadas em determinada versão e seus detalhes de implementação. Outrossim, uma vez definidas as especificações, necessário que os distribuidores do JavaScript implementem o especificado.

Alternativamente, existe uma vasta gama de conversores de código - *transpilers* - para JavaScript; possibilitando programar em linguagens formais e posteriormente gerar código JavaScript. No obstante, essa alternativa tem seus pontos fracos, necessita-se de mais tempo de depuração, visto que o JavaScript gerado não é conhecido pelo desenvolvedor, e provavelmente o código gerado não será tão otimizado, nem utilizar os recursos mais recentes do JavaScript.

Mesmo com suas fraquezas amplamente conhecidas, JavaScript está presente em praticamente todo navegador atual. Sendo uma espécie de denominador comum entre as plataformas. Essa onipresença torna-o integrante vital no processo de desenvolvimento de jogos multiplataforma em HTML5. Vários títulos renomeados já foram produzidos que fazem extensivo uso de JavaScript, so exemplos: Candy Crush Saga, Angry Birds, Dune II, etc.

Jogos Web são escritos na arquitetura cliente servidor, JavaScript pode rodar em ambos estes contextos, para tanto, sua especificação não define recursos de plataforma. Distribuidores do JavaScript complementam a o JavaScript com recursos específicos para suas plataformas alvo. Por exemplo, para servidores, define-se objetos de terminal, acesso a arquivos e dispositivos, etc. No contexto de cliente, são definidos objetos como janelas, frames, DOM, etc.

Para o navegador o código JavaScript geralmente é disposto no elemento “script” dentro de arquivos HTML. Quando os navegadores encontram esse elemento eles fazem a requisição para o servidor e injetam o código retornado no documento, e a não ser que especificado de outra forma, iniciam sua execução.

4.5.1 ASM.JS

Asm.js é um subconjunto da sintaxe do JavaScript a qual permite grandes aumentos de performance quando em comparação com JavaScript normal. No contexto dos jogos performance usualmente um recurso estimável, asm.js consegue-o supra utilizando recursos que permitam otimizações antes do tempo *ahead of time optimizations*. Entretanto, não é trivial escrever código em asm.js e geralmente a geração de código asm.js é feita através da transpilação de outras linguagens como C.

¿ Muita da performance adicional em relação ao JavaScript devido a consistência de tipo e a não existência de um coletor de lixo (memória gerenciada manualmente através de um grande vetor). Esse modelo simples desprovido de comportamento dinâmico, sem alocação e desalocação de memória, apenas um bem definido conjunto de operações de inteiros e flutuantes possibilita grande performance e abre espaço para otimizações.

4.5.2 AJAX

4.5.3 ALTERNATIVAS AO JAVASCRIPT

Abaixo seguem algumas tecnologias que servem de alternativa ao JavaScript.

4.5.4 TYPESCRIPT

Conhecido como uma versão estendida do JavaScript que compila para JavaScript normal.

4.5.5 DART

Google. DartVM é uma máquina virtual que está embutida no Google Chrome. Significantes melhorias em performance quando comparado ao JavaScript. Existe o dart2js que compila código em Dart para JavaScript.

4.6 DOCUMENT OBJECT MODEL (DOM)

É uma plataforma e interface agnóstica à linguagem que permite aos programas e scripts dinamicamente acessar e atualizar o conteúdo, estrutura e estilo de documentos. Pode ser novamente processado e o resultado aparecer na tela. O navegador cria um DOM quando ele processa os elementos e tags encontrados em um documento HTML. Gmail é uma aplicação de única página (single-page) que se baseia fortemente no DOM para gerar conteúdo dinâmico e interativo oferecido pelo DOM.

4.6.1 CANVAS

A nova tag `<canvas>` define um layer gráfico em documentos HTML que pode ser desenhado através de JavaScript.

Permite desenhar diagramas, gráficos e animações [7]. Baseado em bitmap.

O suporte ainda é escasso.

Muitas vezes lento. Algumas soluções tentam arrumar isso através da utilização de GPU. Apache Cordova utiliza o FastCanvas.

CocoonJS é um aplicativo híbrido que preenche a fraca implementação de OpenGL nos dispositivos móveis possibilitando-se desenvolver em WebGL.

4.7 WEBGL

Baseado no OpenGL.

WebGL não foi utilizada no trabalho apesar de ser de grande relevância no processo de jogos pois ainda não está completamente especificada e a dificuldade e escopo do projeto aumentariam muito se tivessem de incluir um jogo 3D. Versão da especificação atual?

4.8 VIDEO

4.9 AUDIO

Áudio é um componente vital para oferecer grande satisfação aos usuários de jogos. Prov feedback e imerge o usuário. Efeitos de som e música podem servir como mecanismo. Jogadores têm baixa tolerância a volume, deve ser utilizado com cautela.

4.9.1 TAG AUDIO

A tag `<audio>` define um som dentro de um documento HTML. Quando o elemento é renderizado pelos navegadores, ele carrega o conteúdo que pode ser reproduzido pelo player de áudio do navegador. Existem muitas discrepâncias entre os formatos aceitos pelos navegadores. Um tanto limitada quanto comparada ao áudio de múltiplos canais disponibilizados por SDKs nativas.

4.9.2 API DE ÁUDIO

uma interface de áudio experimental para JavaScript. Prov maior flexibilidade na manipulação de áudio. Essa tecnologia muito mais nova do que a tag áudio.

FORMATOS DE ÁUDIO

4.10 CÂMERA

4.11 ENTRADA DE COMANDOS

Na construção da grande maioria dos jogos muitas vezes imprescindível alta flexibilidade na gestão de entrada de dados. Este fator muito se amplia na criação de jogos multiplataforma, seja através de teclado, tela sensível ou sensor de movimentos, o importante oferecer a melhor experiência possível por plataforma. O HTML5 trata todos estes casos abstratamente na forma de eventos, os quais podem ser escutados através de listeners. Os eventos básicos são: `keydown` (tecla baixa), `keyup` (tecla solta) e `keypress` (tecla pressionada).

Para detectar suporte aos mais variados recursos do HTML5 no navegador 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. 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.

4.12 CACHE

Aplicações offline.

Algumas tecnologias desta classe são:

4.12.1 OFFLINE E ARMAZENAMENTO

Uma das grandes limitações do HTML era a ausência de capacidade de armazenamento de dados. Armazenamento no lado do cliente um requerimento básico para qualquer aplicação moderna. Essa era uma das áreas onde as aplicações nativas detinham grande vantagem sobre as aplicações web. O HTML5 solucionou este problema introduzindo várias formas de armazenamento de dados. (HASAN et al, 2012)

4.12.2 LOCAL STORAGE

Também conhecido como WebStorage na especificação do HTML5. Prov uma forma de armazenar os dados como chave-valor dentro do navegador. Os dados são persistidos mesmo que o navegador seja fechado.

4.12.3 WEB SQL

Simplesmente um banco de dados SQLite embutido no navegador. Permite tabelas relacionais. O tamanho padrão do banco de dados é 5 megabytes e pode ser estendido pelo usuário.

4.12.4 RECURSOS NATIVOS ATUALMENTE INDISPONVEIS PARA O HTML5

- Suporte camera; - Suporte calendario;

4.12.5 DEBUG

4.12.6 WEINRE

5.21 TECNOLOGIAS POLYFILL Acarretando assim, que muitos navegadores no implementam algumas funcionalidades, completa ou parcialmente especificadas, da surge a necessidade dos polyfills (tecnologias de preenchimento de lacunas) para implementar estas camadas.

Uma das solues mais promissoras polyfill o PhoneGap ou Apache Cordova, esta ferramenta Open-source e possibilita utilizar de inmeros recursos de hardware da grande maioria das produtoras de dispositivos mveis.

4.12.7 FERRAMENTAS

4.12.8 NODEJS

Permite rodar JavaScript fora do navegador. Utiliza um modelo dirigido eventos sem bloqueio, tornando-o rpido e eficiente.

4.12.9 SISTEMAS DE BUILDING

Aquivos JavaScript so requisitados do servidor assincronamente. Isso pode levar a tempos de requisio pouco desejveis. Uma sada seria escrever o cdigo em apenas um arquivo mais isso leva a gerncia de cdigo bagunada. A sada mais comum entre desenvolvedores utiliz ruma ferramenta que junta todos os arquivos e disponibiliza apenas um para o usurio.

4.12.10 GRUNT

Aplica as modificaes separadamente em cada arquivo.

4.12.11 GULP

Utiliza o conceito de streams para aplicar todas as modificaes sobre um arquivo de uma vez s.

4.12.12 SOURCE MAPS

Para encontrar os arquivos minificados a fim de ajudar o desenvolvedor a debugar a aplicao.

4.12.13 Minificar

Remover caracteres desnecessrios do JavaScript como espaos vazios, diminuindo o tamanho dos nomes, fazendo o tempo de loading diminuir.

4.12.14 GERENCIADORES DE PACOTES

4.12.15 BOWER

4.12.16 NPM

4.12.17 DISPONIBILIZAO DA APLICAO

Links com manifestos

4.13 INSTALAO

Este mtodo benfico pois possibilita ao usurio a mesma experincia ao adquirir uma aplicao normal. Este tipo de aplicao comumente referido como "hbrido".

4.14 CROSSWALK

Crosswalk empacota os fontes juntamente com uma verso do Chromium, a verso Open-source do Google Chrome. Isso faz com que o software se comporte da mesma forma para todas as verses de dispositivos Android.

4.15 PHONEGAP

4.16 PHONEGAP CLOUD

Este servio possibilita que se faa upload de um arquivo compactado contendo os fontes ou apontando para um repositrio no GitHub que no tempo desta pesquisa no estava funcionando; e se gere o APK para o Android nativamente.

4.16.1 O JOGO

Devido ao fato deste trabalho explorar as limitaes dos jogos em HTML5, optei por evitar a utilizao de plugins e ferramentas de terceiros que pudessem ocultar alguma limitao.

Escolhi a simplicidade para no precisar ficar muito tempo aprendendo as coisas em detrimento do refinamento da pesquisa.

4.17 MECNICA

O jogo consiste em simplesmente em uma tela que apresenta equaes e um possvel resultado. Cabe ao jogador decidir se o resultado est certo ou errado. O tempo um fator levado em considerao, quo mais rpido o jogador acertar se a afirmao est correta ou no, mais pontos ele receber.

Argumentos favor da escolha do game: Tem profundidade, permite a adio de novos recursos no futuro; facilmente traduzvel em tamanhos de telas diferentes e tipos de entrada de dados diferentes;

4.18 IMPLEMENTAO

No tenho grande experincia com o desenvolvimento de jogos nem com o desenvolvimento em HTML5. Tambm para no interferir na pesquisa busquei no me distanciar do que considerado padro em ferramentas e mtodos. Comecei escrevendo o aplicativo para o Navegador do desktop pois era o que estava mais acessvel no momento. Mais tarde descobri que de fato assim que de desenvolve.

4.19 TRABALHOS SIMILARES

(Referncia 2) Faz uma reviso de aspectos do HTML5 atravs da construo de um jogo. O autor foca muito nos aspectos de criao de jogos e feedback do desenvolvimento. Troca de tecnologias e no especificamente nas limitaes conforme o meu trabalho. Em outras palavras seu escopo mais genrico e no to preciso quanto este

4.20 ANDROID

um sistema operacional *open-source* desenvolvido pela Google. Utiliza o kernel Linux. Softwares para Android são geralmente escritos em Java e executados através da máquina virtual Dalvik.

similar à máquina virtual Java, mas roda um . formato de arquivos diferenciado (dex), otimizados para consumir pouca . memória, que são agrupados em um pacote Android Package (apk) Android. permite a renderização de documentos HTML através de sua própria . API WEBVIEW. Ou através do navegador disponibilizado por padrão, ou . outros de terceiros como o Google Chrome, Firefox, Opera, etc .

No quesito jogos para dispositivos móveis prefere disponibilizar os jogos através da interface nativa pois dá a sensação de continuidade para com os demais aplicativos instalados no dispositivo.

4.21 NAVEGADORES

Aplicações do lado do cliente geralmente se comunicam com um servidor através de documentos em HTTP. Quando o navegador recebe um destes pacotes em HTML ele começa o processo de renderização. A renderização pode requisitar outros arquivos a fim de completar a experiência desenvolvida para o endereço em questão.

Nos navegadores os usuários necessitam localizar a página que desejam, sabendo o endereço, ou pesquisando em buscadores. Isso é um processo ruim para as plataformas móveis pois necessitam maior interação do usuário e não são naturais se comparado ao modo normal de consumir aplicativos nestas mesmas plataformas simplesmente adquirindo o aplicativo na loja e abrindo-o no sistema operacional. Alguns contornos para este problema serão descritos nas tecnologias offline.

Para transformar as instruções retornadas pelo servidor em algo útil para o usuário final os navegadores geralmente fazem uso de bibliotecas externas capazes de interpretar HTML5 e gerar o conteúdo iterativo.

4.21.1 BIBLIOTECAS WEB

O Google Chrome utiliza o Webkit para renderizar seu conteúdo HTML5. O webkit foi criado pela Apple baseando-se no motor de renderização do Konqueror do projeto KDE. Safari e Opera também fazem uso do Webkit. V8 para JavaScript.

O motor de renderização do HTML5 do Firefox é o XXX. O motor de JavaScript é o.

5 METODOLOGIA

Este trabalho procura investigar a estratégia web, sendo assim não será explorado em detalhes as alternativas nativas.

Também foi dada primazia ao HTML puro, pois ao utilizar frameworks muitos dos problemas da web podem já serem resolvidos por essas ferramentas e as limitações ficarão escondidas.

O primeiro passo consiste em definir as plataformas alvo do trabalho; devem ser plataformas mercadologicamente relevantes ao desenvolvimento de jogos, que possibilitem a criação de aplicativos em HTML e que acentuem o antagonismo de características.

Segue-se com a construção de uma lista com os recursos relevantes aos jogos que, empiricamente, sofrem ou são comumente ligados limitações multiplataforma. Segue-se uma pesquisa para aprofundar teoricamente cada um dos recursos, possivelmente elegendo novos.

Com um baseamento teórico substancial, o próximo passo é a criação do protótipo de um jogo multiplataforma que utilize recursos potencialmente limitados. Para ser considerado pronto, o protótipo deve ser testado, e estar funcional, com adaptações ou não, em cada uma das plataformas alvo definidas.

Com o protótipo concebido, o passo que segue a enumeração, e descrição das limitações detectadas no processo de desenvolvimento e testes do jogo. Este detalhamento deve responder as seguintes perguntas:

- Quais as limitações foram encontradas no jogo? - Em quais plataformas? - Sob quais circunstâncias? - As limitações puderam ser contornadas? - Algum efeito colateral das limitações no jogo? - Qual a categoria do problema: usabilidade, funcionalidade, manutibilidade, portabilidade ou performance? (segundo ISO)

6 RESULTADOS

Abaixo constam as limitações encontradas durante a pesquisa e concepção do jogo

Durante a construção do jogo utilizei a estratégia de declarar todos os objetos relativos ao window e limitar o escopo. Isso se demonstrou uma boa forma de separar as responsabilidades.

6.0.2 8.1 LIMITAÇÕES

1. Apesar da grande maioria dos recursos dos dispositivos estar presente em HTML5 ainda existem muitas funcionalidades faltando para este tipo de aplicação. Por exemplo, não podemos mudar a imagem de fundo do dispositivo, ou adicionar toques etc. Similarmente, existem muitas APIs de nuvem como os serviços de impressão do iCloud ou Google cloud que estão disponíveis para aplicações nativas mas não para HTML5. Outros serviços utilitários como o C2DM do Google que está disponível para desenvolvedores Android para utilizar serviços de push também não estão disponíveis para o HTML5. (HASAN, 2012)

2. VERSES A grande maioria dos dispositivos atualmente no mercado utilizam obsoletos de seus softwares. Isso dificulta o desenvolvimento. Se a tecnologia de tradução para o navegador utilizar a classe Webview do Android - como o Apache Cordova faz - as versões mais antigas podem ser penalizadas com problemas de performance ou falta de recursos.

2. OFFLINE

Refresh duplo para ver assets cacheados. Ver: <http://buildnewgames.com/game-asset-management/>

3. AUDIO API de som quebra quando executado diversas vezes. Os navegadores variam na disponibilização de formatos aceitáveis. Somente um áudio pode ser tocado no Navegador do Android. Não é possível trocar o volume no iOS. Alguns navegadores favorecem formatos ogg (vorbis) e outros, como o Safari, favorecem o MP3.

4. O maior problema com as APIs de áudio e de vídeo do HTML5 é a disputa entre os codecs dos navegadores. Por exemplo, Mozilla e Opera suportam Theora, já o Safari suporta H.264 que também é suportado pelo IE9. Ambos, iPhone e Android suportam H.264 em seus navegadores. A W3C recomenda OggVorbis e OggTheora para áudio e vídeo respectivamente. (HASAN et al, 2012)

3. VIDEO

Codecs

4. ASSETS

Trafegar muitos assets deixa o sistema lento.

Contorno Utilizando páginas de carregamento e/ou cache;

5. UI

É muito custoso desenvolver uma interface que pareça nativa para cada dispositivo sem a utilização de plugins e ferramentas especializadas. Em termos gerais, trabalhar com propores positivo. No entanto há casos, como o dos botões de certo e errado que as propores ficam exageradas, nesses casos a utilização de max-width é uma solução conveniente.

6. PERFORMANCE

De acordo com uma pesquisa, para um usuário uma tarefa instantânea se ele leva até 0.1 segundos para ser executada. Se a tarefa toma aproximadamente um segundo então a demora será

notada mas o usuario no se incomodar com ela. Entretanto, se a tarefa leva aproximadamente 10 segundos para terminar o usuario ento comea a ficar aborrecido e esse o limite que algum feedback deve ser dado para um usuario.

ACELERAO DE GPU

7. Acelermetro

8. IMPLEMENTAO INCONSISTENTE DE APIs

9. TAMANHO DE TELA Em alguns casos o tamanho das telas pode ser um fator limitante como no caso de jogos de estrategia. Jogadores com telas menores podem sair em desvantagem.

9. CMERA

10 . JavaScript Ciclo de vida demorado pois necessita que todos os consumidores da especificao entrem em consenso e implementem a.

Desktop/Firefox Desktop/Google Chrome Smartphone/Android

7 CONCLUSES

No pude testar todos os mtodos e ferramentas e verses disposio, um trabalho completo demandaria esforos conjuntos de muitos individuos ou um periodo de tempo bem mais extenso. Se uma empresa deseja produzir jogos nativos elas precisaro de vrios desenvolvedores. Eu sozinho fui capaz de produzir um jogo em tempo razovel trabalhando apenas com a plataforma web.

Por no utilizar frameworks e bibliotecas estou me distanciando dos casos da vida real. S poderemos considerar o HTML como uma especificao pronta quando for possvel fazer tudo o que se faz nativamente com os dispositivos atravs de uma API web padronizada.

Conforme JavaScript vai ganhando importancia rpido progresso feito por diferentes empresas a fim de prover boas ferramentas de debug e inspecionamento para JavaScript.

Baixa frico quer dizer que voc pode ir de um site para outro sem ter que instalar, o servio est em demanda, voc no obrigado a t-lo em sua tela inicial.

7.0.3 TRABALHOS FUTUROS

EMACSCRIPT 7

8 ANEXOS

8.0.4 CONVERSORES PARA HTML5

Alm da possibilidade de escrever em HTML, pode-se optar pela alternativa de utilizar-se um conversor de linguagens.

8.0.5 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE PARA A CONSTRUO DE GAMES

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

- OpenUP: este bem detalhado e de caracterstica iterativa e incremental. Gerando assim, um levantamento mais apurado dos riscos, requisitos e outros detalhes do sistema e a criao incremental do sistema, com requisitos maleveis;

- Cascata: processo antigo, caracteriza-se por ser pouco malevel aos requisitos mapeados posteriormente ao processo de anlise;

- Processo gil - SCRUM: sua utilizao flexvel e sendo um mtodo gil especifica pouca documentao, ou como dizem, somente a documentao necessaria, este processo bem conhecido e aceito na comunidade de desenvolvimento de software. Suas principais caractersticas so: diviso do processo de desenvolvimento atravs uma serie de iteraes chamadas sprints. Cada sprint consiste

tipicamente em duas a quatro semanas. bem aplicado a projetos que mudam constantemente e que demandam rápidas adaptaes;

- Processo gil XP: tem muitas caractersticas similares ao SCRUM por este tambm ser um processo gil. Dentre suas especificidades destaca-se: verses frequentes, pequenos ciclos de desenvolvimento que buscam aumentar a produtividade, introduzem checkpoints onde os clientes podem agregar novas funcionalidades;

8.0.6 AMBIENTES PARA DESENVOLVIMENTO HTML5

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

- segundo (PRADO, 2012) o GWT um framework essencialmente para o lado do cliente (client side) e d suporte comunicao com o servidor atravs de RPCs Remote Procedure Calls (ou procedimento de chamadas remotas). Ele no um framework para aplicaes clssicas da web, pois deixa a implementao da aplicao web parecida com implementaes em desktop. Este utilizado em muitos produtos de grande porte como o Google Adwords e Google Wallet. Outra caracterstica interessante que a plataforma opera sobre a licena Apache verso 2;

- construct 2 - um editor na nuvem focado para usurios sem conhecimento prvio em programao orientado a comportamento; - PlayCanvas - uma plataformas para a construo de jogos 3D na nuvem, desenvolvida com foco em performance. Permite a hospedagem, controle de verso e publicao dos aplicativos nela criados, possibilita tambm a importao de modelos 3D de softwares populares como: Maya, 3ds Max e Blender;

- o ambiente HTML5 da Intel, este fornece uma soluo na nuvem, completa para o desenvolvimento em plataforma cruzada, com servios de empacotamento, servios para a criao e testes de aplicativos com montagem de interfaces puxa e arrasta (Intel XDK) e bibliotecas para a construo de jogos utilizando acelerao de hardware, o que garante at duas vezes mais performance que aplicativos mobile baseados em Web tradicionais. Esta soluo gratuita, open-source e funciona atravs de um plugin para o Google Chrome, ou seja, o desenvolvimento tambm multi-plataforma e devido ao fato de os binrios ficarem hospedados na nuvem, possibilitou a Intel criar compiladores para cada uma das plataformas disponibilizadas pelo PhoneGap, que o framework polyfill utilizado na soluo.

8.0.7 HTTP

8.0.8 FRAMEWORKS DE DESENVOLVIMENTO DE JOGOS EM HTML5;

8.0.9 FRAMEWORKS PARA DESENVOLVIMENTO DE JOGOS HTML5

Com o intuito de simplificar o processo para os desenvolvedores, auxiliando-os a focarem-se apenas nas solues que esto desenvolvendo, foram criados os frameworks para desenvolvimento de jogos. No obstante, o intuito deste trabalho desenvolver um jogo sem auxilio de frameworks pois estes muitas vezes escondem possveis limitaes, desenvolvendo solues prprios.

- enchant.js: dentre suas funcionalidades constam: orientao ; - objetos, orientado eventos, contm um motor de animao ; - suporta WebGL e Canvas, etc three.js: considerada leve, renderiza ; - WebGL e Canvas, arquitetura procedural ; - quintus: bom para plataformas 2D - limeJs: bom para 2d

8.0.10 INTERFACE E ESCOLHAS DE DESIGN

8.0.11 PROGRESSO CONTNUA

8.0.12 JAVASCRIPT NO OBSTRUTIVO

8.0.13 Arquitetura Cliente Servidor

References

- [1] K. J. Arrow, L. Hurwicz, and H. Uzawa, “Constraint qualifications in maximization problems,” *Naval Research Logistics Quarterly*, vol. 8, pp. 175–191, 1961.