

1 JOGOS

1.1 BENEFÍCIOS

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.

1.2 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)

1.3 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.

1.4 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 67

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.

2 ESTE TRABALHO

Este projeto propõe analisar as limitações do HTML5 quanto relativo à construção de jogos multiplataforma. Através de revisão bibliográfica e da criação de um protótipo 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 limitações. Também no objetivo deste trabalho comparar o HTML com outras tecnologias de desenvolvimento de jogos, como Flash Player, Silverlight ou alternativas Desktop.

2.1 O JOGO

Para a análise prática das limitações foi escolhido um jogo de matemática simples. Consistindo na geração de equações com um candidato de resposta. Cabe ao usuário informar se o resultado apontado pelo jogo está correto ou não.

Porque escolhi esse tipo de jogo?

PROBLEMA A carência de definições concretas sobre a viabilidade da atual versão do HTML5 - quando utilizado no desenvolvimento de jogos e o senso comum, acabam por monopolizar a construção de jogos nativos às plataformas alvo.

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

3 OBJETIVOS

Abaixo seguem os objetivos deste trabalho.

3.1 OBJETIVO GERAL

Identificar possíveis limitações no processo de desenvolvimento de jogos multiplataforma oriundas do atual estado de definição e implementação do HTML5.

3.2 OBJETIVOS ESPECÍFICOS

Estudar as limitações 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 contém a vasta maioria do mercado de dispositivos inteligentes, e por termos maior experiência na mencionada plataforma.

Pretende-se também estudar os seguintes tópicos do desenvolvimento de jogos, relativos ao HTML5: - Depuração - Diferenças em tamanho de tela - Canvas 1. Troca de tamanhos via Canvas vs DOM 2. Aceleração de GPU 3. API de áudio (referência 2) - Performance - Empacotadores HTML5 - Eventos de entrada - Vibração - Acelerômetro - Armazenamento - Disponibilização de assets (controle de tamanhos, cache, etc) - Aplicações offline - CSS media queries

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

!- Pensamento: talvez seja interessante concluir se é viável produzir jogos com HTML5 do ponto de vista mercadológico -;

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

concepo. (REFERENCIAR) Muitos desenvolvedores esto familiarizados com as tecnologias da WEB ou apontam interesse na tecnologia. ¡!– referenciar –¡ Estimular e avanar o estudo da implementao da Open Web;

REVISO BIBLIOGRFICA

4 JOGOS

Segundo LEMES (2009, pg. 126) ¡ jogo digital constitui-se em uma atividade ldica composta por uma srie de aes e decises, limitada por regras e pelo universo do game, que resultam em uma condio final.

Essa caracterstica interativa a dependncia comandos sobre uma interface digital, que faz com que o projeto digital desta natureza no seja um filme ou uma animao, e sim um game.

Quando desenvolvendo qualquer jogo, o desenvolvedor tem que considerar seu usurio. O objetivo maximizar a satisfao de seu usurio. Jogos em plataformas mveis trazem um novo conjunto de desafios para produtores de jogos. Um destes desafios fornecer feedback suficiente para o player pois o dispositivo limitado em propores, som, tela etc. J jogos multiplataforma em HTML5 tem a dificuldade adicional de ter que comportar, na mesma base, o feedback adequando para cada plataforma mvel.

A interface tem que ser o mais intuitiva o possvel. No caso de dispositivos mveis, quanto menos gestos necessrios melhor Tornar previsvel causa e efeito uma boa caracterstica para os jogos Os desenvolvedores tem que evitar fazer o jogo para eles mesmos. E pela falta de crtica os designs tendem a ser ruins. Afinal o que os jogadores querem? LEMES (2009, pg XX) aponta alguns fatores procurados pelos usurios de jogos: Desafio, socializar, experincia solitria,. respeito e fantasia .

Mencionar algum jogo (como WOW) e como ele faz para prender a ateno dos usurios. Candy crush saga

4.1 GNEROS

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

- Adventure Ao RPG Estratgia Simuladores Esportes Luta Casuais - 'God' Games Educa-
cionais Puzzle Online / Massive Multiplayer

4.2 MECNICA

A mecânica composta pelas regras do jogo. Quais as aes disponveis aos usurios, fortemente influenciada pela categoria do jogo em questo.

4.3 ARQUITETURA

Existem algumas estratgias relativas s plataformas alvo de como efetuar construo de jogos.

4.4 DESENVOLVIMENTO DE JOGOS NATIVOS

Habilita a melhor experincia de usurio pois permite utilizar ao mximo os recursos e funcionalidades dos aparelhos. Porm, devido a cada plataforma conter seu prprio sistema operacional, com seus prprios *SDK's* totalmente incompatveis, os desenvolvedores so forados a desenvolver uma verso do jogo para cada plataforma alvo. Alm da replicao dos fontes, esta abordagem requer mais pessoas, e maior custo com possivelmente parte do mercado no atendido de qualquer forma.

4.5 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 aprend-la. Seus custos também so inferiores, aos do desenvolvimento nativo pois demandam menos trabalhadores/hora devido a inexistncia de duplicao da base. No obstante, esta opo devido a incompletude da especificao de padres carece de alguns recursos e outros no esto completamente implementados. Performance também pode ser um limitador, visto que estas tecnologias so executadas atravs de um navegador, criando uma camada de abstrao superior das APIs nativas que fazem chamadas ao sistema diretamente.

4.6 DESENVOLVIMENTO DE JOGOS HBRIDOS

Jogos hbridos so jogos geralmente desenvolvidos com tecnologias da . web: beneficiando-se da no necessidade de duplicao. 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 uma ferramenta deste tipo. Permite acessar . os dispositivos utilizando sua API JavaScript. Funciona encapsulando . todo o código HTML5. Este tipo de abordagem permite acessar cmera, . acelermetro, GPS, etc .

5 WEB

5.1 OPEN WEB

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

Neste conjunto inclui-se:

- Descentralizao; - Transparncia; - Relevncia; - Imparcialidade; - Consenso; - Disponibilidade;
- Manutibilidade;

!- Ref <http://codinginparadise.org/weblog/2008/04/what's-open-web-and-why-is-it-important.html>
<http://tantek.com/2010/281/b1/what-is-the-open-web> -

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

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

6 HTML

Hyper Text Markup Language (HTML) uma linguagem de marcao que define a estrutura semntica do contedo das pginas da web, criada por Tim Berners Lee e oficialmente mantida pela W3C. O "Hyper Text" refere-se a links que conectam pginas umas as 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*, abraando assim suas premissas:

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

No obstante, os criadores de navegadores constantemente introduziam elementos de apresentao com o *blink*, *i* itlico, *b* bold, que eventualmente acabavam por serem inclusos 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.

7 CSS

uma linguagem de folhas de estilo, criada por Hkon 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 Hkon 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 –`

8 JAVASCRIPT

EMAScript, 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 já estão disponíveis através de uma interface de usuário, uma linguagem de script 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 a 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 renomados já foram produzidos que fazem extensivo uso de JavaScript, são 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 não há que especificado de outra forma, iniciam sua execução.

8.1 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 transpilagem de outras linguagens como C.

↳ Muita da performance adicional em relação ao JavaScript devido à consistência de tipo e à 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.

8.2 AJAX

8.3 ALTERNATIVAS AO JAVASCRIPT

Abaixo seguem algumas tecnologias que servem de alternativa ao JavaScript.

8.4 TYPESCRIPT

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

8.5 DART

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

9 DOCUMENT OBJECT MODEL (DOM)

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 uma aplicação de única página (single-page) que se baseia fortemente no DOM para gerar conteúdo dinâmico e interativo oferecido pelo DOM.

9.1 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 uma aplicação híbrida que preenche a fraca implementação de OpenGL nos dispositivos móveis possibilitando-se desenvolver em WebGL.

10 WEBGL

Baseado no OpenGL.

Web GL 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?

11 VIDEO

12 AUDIO

Audio 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 ao volume, deve ser utilizado com cautela.

12.1 TAG AUDIO

A tag `<audio>` define um som dentro de um documento html. Quando o elemento 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.

12.2 API DE AUDIO

uma interface de audio experimental para JavaScript. Prov maior flexibilidade na manipulao de audio. Essa tecnologia muito mais nova do que a tag audio. FORMATOS DE UDIO

13 CAMERA

14 ENTRADA DE COMANDOS

Na construo da grande maioria dos jogos muitas vezes imprescindvel alta flexibilidade na gesto de entrada de dados. Este fator muito se amplia na criao de jogos multiplataforma, seja atravs de teclado, tela sensvel ou sensor de movimentos, o importante oferecer a melhor experincia possvel por plataforma. O HTML5 trata todos estes casos abstratamente na forma de eventos, os quais podem ser escutados atravs de listeners. Os eventos bsicos so: 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 implementao de cada uma ou ento fazer uso de alguma biblioteca especializada neste processo, o Modernizr uma opo open-source deste tipo de biblioteca, este gera uma lista de booleanos sobre grande variedade dos recursos HTML5, dentre estes, geolocalizao, canvas, udio, vdeo e local storage.

15 CACHE

Aplicaes offline.

Algumas tecnologias desta classe so:

15.1 OFFLINE E ARMAZENAMENTO

Uma das grades limitaes do HTML era a ausncia de capacidade de armazenamento de dados. Armazenamento no lado do cliente um requerimento bsico para qualquer aplicao moderna. Essa rea era onde as aplicaes nativas detinham grande vantagem sobre as aplicaes web. O HTML5 solucionou este problema introduzindo vrias formas de armazenamento de dados. (HASAN et al, 2012)

15.2 LOCAL STORAGE

Tambm conhecido como WebStorage na especificao do HTML5. Prov uma forma de armazenar os dados como chave valor dentro do navegador. Os dados so persistido mesmo que o navegador seja fechado.

15.3 WEB SQL

Simplesmente um banco de dados SQLite embebido no navegador. Permite tabelas relacionais. O tamanho padro do banco de dados 5 megabytes e pode ser estendido pelo usurio.

15.4 RECURSOS NATIVOS ATUALMENTE INDISPONVEIS PARA O HTML5

- Suporte cmera; - Suporte calendrio;

15.5 DEBUG

15.6 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.

15.7 FERRAMENTAS

15.8 NODEJS

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

15.9 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 utiliza ruma ferramenta que junta todos os arquivos e disponibiliza apenas um para o usurio.

15.10 GRUNT

Aplica as modificaes separadamente em cada arquivo.

15.11 GULP

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

15.12 SOURCE MAPS

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

15.13 Minificar

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

15.14 GERENCIADORES DE PACOTES

15.15 BOWER

15.16 NPM

15.17 DISPONIBILIZAO DA APLICAO

Links com manifestos

16 INSTALAO

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

17 CROSSWALK

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

18 PHONEGAP

19 PHONEGAP CLOUD

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

19.1 O JOGO

Devido ao fato deste trabalho explorar as limitações dos jogos em HTML5, optei por evitar a utilização de plugins e ferramentas de terceiros que pudessem ocultar alguma limitação. Escolhi a simplicidade para não precisar ficar muito tempo aprendendo as coisas em detrimento do refinamento da pesquisa.

20 MECÂNICA

O jogo consiste em simplesmente em uma tela que apresenta equações e um possível resultado. Cabe ao jogador decidir se o resultado está certo ou errado. O tempo é um fator levado em consideração, quanto mais rápido o jogador acertar se a afirmação está correta ou não, mais pontos ele recebe.

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

21 IMPLEMENTAÇÃO

Não tenho grande experiência com o desenvolvimento de jogos nem com o desenvolvimento em HTML5. Também para não interferir na pesquisa busquei não me distanciar do que é considerado padrão em ferramentas e métodos. Comecei escrevendo o aplicativo para o Navegador do desktop pois era o que estava mais acessível no momento. Mais tarde descobri que de fato assim que se desenvolve.

22 TRABALHOS SIMILARES

(Referência 2) Faz uma revisão de aspectos do HTML5 através da construção de um jogo. O autor foca muito nos aspectos de criação de jogos e feedback do desenvolvimento. Troca de tecnologias e não especificamente nas limitações conforme o meu trabalho. Em outras palavras seu escopo é mais genérico e não tão preciso quanto este.

23 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 a 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-se disponibilizar os jogos através da interface nativa pois dá a sensação de continuidade para com os demais aplicativos instalados no dispositivo.

24 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 rápido para as plataformas móveis pois necessitam maior interação dos usuários 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 interativo.

24.1 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.

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 fiquem 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 a 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 é 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)

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.

24.2 8.1 LIMITAÇÕES

⌋ 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 úteis 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)

1. 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.

⌋ 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

Trazer muitos assets deixa o sistema lento.

Contorno Utilizando páginas de carregamento e/ou cache;

5. UI

Muito custoso desenvolver uma interface que pareça nativa para cada dispositivo sem a utilização de plugins e ferramentas especializadas. Em termos gerais, trabalhar com propósitos positivos. No entanto há casos, como o dos botões de certo e errado que as proporções 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 usuário não se incomoda com ela. Entretanto, se a tarefa leva aproximadamente 10 segundos para terminar o usuário então começa a ficar aborrecido e esse é o limite que algum feedback deve ser dado para um usuário.

ACELERAÇÃO DE GPU

7. Acelerômetro

8. IMPLEMENTAÇÃO 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 estratégia. Jogadores com telas menores podem sair em desvantagem.

9. CÂMERA

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

Desktop/Firefox Desktop/Google Chrome Smartphone/Android

CONCLUDES

No pude testar todos os métodos e ferramentas e verses disposio, um trabalho completo demandaria esforços conjuntos de muitos indivíduos ou um período de tempo bem mais extenso. Se uma empresa deseja produzir jogos nativos elas precisaro de vários desenvolvedores. Eu sozinho fui capaz de produzir um jogo em tempo razoável trabalhando apenas com a plataforma web.

Por não utilizar frameworks e bibliotecas estou me distanciando dos casos da vida real. Seremos considerar o HTML como uma especificação pronta quando for possível fazer tudo o que se faz nativamente com os dispositivos através de uma API web padronizada.

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

Baixa fricção quer dizer que você pode ir de um site para outro sem ter que instalar, o serviço está em demanda, você não obrigado a tê-lo em sua home screen.

24.3 TRABALHOS FUTUROS

EMACSCRIPT 7

ANEXOS

24.4 CONVERSORES PARA HTML5

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

24.5 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE PARA A CONSTRUÇÃO DE GAMES

Como o jogo 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: verses frequentes, pequenos ciclos de desenvolvimento que buscam aumentar a produtividade, introduzem checkpoints onde os clientes podem agregar novas funcionalidades;

24.6 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 dá suporte 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 que a plataforma opera sobre a licena Apache versio 2; - construct 2 - um editor na nuvem focado para usuarios 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 versio 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 Development Environment (ambiente de desenvolvimento HTML5) da Intel, este fornece uma soluo na nuvem, completa para o desenvolvimento em plataforma cruzada, com servios de empacotamento, servios para a criaao e testes de aplicativos com montagem de interfaces drag and drop (Intel XDK) e bibliotecas para a construo de jogos utilizando aceleracao 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 multiplataforma 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.

24.7 HTTP

24.8 FRAMEWORKS DE DESENVOLVIMENTO DE JOGOS EM HTML5;

24.9 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

24.10 INTERFACE E ESCOLHAS DE DESIGN

24.11 PROGRESSO CONTNUA

24.12 JAVASCRIPT NO OBSTRUTIVO

24.13 Arquitetura Cliente Servidor