

1 CONTEXTUALIZAÇÃO

1.1 JOGOS

Jogos é uma tendência que veio para ficar. (colocar estatísticas sobre jogos). Segundo o Wikipedia Seus componentes incluem: regras, desafios e interações. Geralmente envolvem simulação física e mental. Muitos jogos ajudam a desenvolver habilidades práticas, servem como uma forma de exercício, ou de alguma forma executam um papel educacional, de simulação ou psicológico. Jogos são uma parte universal da experiência humana, presentes em todas as culturas.

1.2 JOGOS E MULTIPLATAFORMA

A pluralidade de dispositivos se tornou atraente para a indústria e consumidores de jogos. É laboriosa a tarefa dos produtores neste panorama, existem muitas plataformas. Além da fragmentação de versões, e hardwares diferenciados. Existem estratégias de abordagem no desenvolvimento de jogos que vão decidir quais plataformas serão atendidas pelos jogos.

1.2.1 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 coter 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.

1.2.2 DESENVOLVIMENTO DE JOGOS WEB

Necessitam de apenas uma base de código e pode rodar em todas as plataformas. Contém a mais vasta gama de desenvolvedores e muitos interessados em aprendê-la. Seus custos também são inferiores, aos do desenvolvimento nativo – pois demandam menos trabalhadores/hora devido a inexistência de duplicação da base. Não 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 à das APIs nativas que fazem chamadas ao sistema diretamente.

1.2.3 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 às 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.

1.3 HTML E MULTIPLATAFORMA

A OWP (Open Web Platform), uma coleção de tecnologias livres, amplamente utilizadas e padronizadas. O HTML Hypertext Markup Language é a peça fundamental da OWP, seu objetivo é especificar o conteúdo das páginas web. Trata-se de uma linguagem de marcação que define a estrutura de elementos que uma página deve ter de modo a fornecer conteúdo aos usuários. O HTML5 é um conceito guarda-chuva para designar as tecnologias da web Javascript, CSS3 e HTML. Em termos práticos significa que quando uma tecnologia se torna muito popular através da adoção de algumas grandes empresas e dos desenvolvedores, ela se torna candidata à adoção. A interatividade necessária para a construção de jogos animados em HTML é algo recente, obtido através dos esforços direcionados no HTML5, anteriormente só se obtinha interatividade de tamanho proporcional na Web com a utilização de ferramentas proprietárias como o Adobe Flash e Microsoft Silverlight.

1.4 LIMITAÇÕES DE JOGOS MULTIPLATAFORMA COM HTML5

Enquanto o HTML é desenvolvido muitas das funcionalidades disponibilizadas são testadas em apenas um pequeno conjunto de navegadores para um pequeno conjunto de versões (referência 2). Isso acarreta em suporte inconsistente. A forma mais segura de garantir suporte é testando em todas as versões alvo, todavia essa solução é imprática. Os desenvolvedores de navegadores podem interpretar/implementar as especificações erroneamente aumentando os problemas de compatibilidade. Nem todos os recursos disponíveis através das SDK's nativas estão presentes através do HTML5.

1.5 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. Não é objetivo deste trabalho demonstrar onde o HTML5 se sobressai, apenas suas limitações. Também não é objetivo deste trabalho comparar o HTML com outras tecnologias de desenvolvimento de jogos, como FlashPlayer, Silverlight ou alternativas desktop.

O escopo deste projeto não me permite tanto. ###1.5.1 O JOGO Para a análise 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. Escolhi um jogo deste tipo pois sua concepção necessitava ser razoavelmente veloz para dar continuidade na pesquisa.

2 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 as plataformas alvo. Os custos introduzidos no ciclo vida de um jogo, para diversas plataformas, é muito alto para ser considerado trivial.

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 smartphones Android e navegadores Desktop Google Chrome 42 e Firefox 37. Optamos por Android, e não IOS, pois o primeiro contém a vasta maioria do mercado de Smartphone, e por termos maior experiência na já mencionada plataforma. Pretende-se também estudar os seguintes tópicos do desenvolvimento de jogos, relativos ao HTML5: Debugging Diferenças em tamanho de tela Canvas Resizing via Canvas vs DOM Aceleração de GPU API de Audio (reference 2) Performance Empacotadores HTML5 Eventos de entrada Vibração Acelerômetro Storage Disponibilização de assets (cotrole de tamanhos, cache, etc) Aplicações offline Css media queries Elaborar uma lista de limitações e correlacionar os dados de acordo com as plataformas. 4 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; 5 REVISÃO BIBLIOGRÁFICA 5.1 JOGOS

Segundo LEMES (2009, pg 126)

... jogo digital se constitui em uma atividade lúdica composta por uma série de ações e deci

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. Já jogos multiplataforma em HTML5 tem 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 o 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 5.2 GÊNEROS LEMES (2009, p. 43) aponta os seguintes gêneros de jogos. Adventure Ação RPG Estratégia Simuladores Esportes Luta Casuais ‘God’ Games Educacionais Puzzle Online / Massive Multiplayer

5.3 MECÂNICA DOS JOGOS

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

5.4 NAVEGADORES WEB

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 chamar 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 árduo para as plataformas móveis pois necessitam maior interação do usuários e não são “naturais” se comparado ao modo normal de consumir aplicativos nestas

mesmas plataformas – simplesmente adquirindo o app na loja e abrindo-o no sistema operacional. Alguns contornos para estes problema serão descritos nas tecnologias offline.

5.5 ANDROID

É um sistema operacional open-source, largamente suportado pelo Google. Utiliza o kernel Linux para as tarefas mais básicas como: gerência de memória, processos, etc. Os programas para Android são geralmente escritos em Java e executados através da máquina virtual Dalvik. Dalvik é similar a máquina virtual Java, mas roda um formato de arquivos diferenciado (dex), otimizados para consumir pouca memória, que são agrupados em um único 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 é preferível disponibilizar os jogos através da interface nativa pois dá a sensação de continuidade e igualdade para com os demais aplicativos instalados no dispositivo.

5.6 HTML

Linguagem de marcação da web. Criada para definir os elementos de uma página da web. Com o crescimento da demanda de interatividade na internet o HTML foi forçado a evoluir. Em sua quinta versão, o HTML5, grandes melhorias foram adicionadas. Antes do HTML5 para alcançar a interatividade desejada, só se podia recorrer a plugins de terceiros como o FlashPlayer e o Silverlight, estes necessitam de instalação de plugin e ficam limitados a um distribuidor de software. 5.7 CSS Uma linguagem de marcação, como o HTML, que utiliza seletores e regras para definir a representação de um documento HTML e seus elementos. Os navegadores interpretam CSS através da tag

. Sua última versão, o CSS3, introduziu várias funcionalidades multiplataforma como media-queries que possibilitam regras para tamanhos de tela e Transformações 3D. Muitos navegadores também suportam aceleração de GPU (Graphics Process Unit) 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.

5.8 JAVASCRIPT Todo o navegador atual contém um interpretador de javascript tornando-o portátil. Emascript criado pela Netscape se tornou tão popular que foi abraçada pela W3C Atualmente a linguagem de programação mais popular do mundo. Javascript 6 trás conceitos de orientação à objetos. Existem vários transpiladores para javascript que permitem a utilização de linguagens que evoluem mais rapidamente. Na versão estável atual 5.1, é uma linguagem que falha ao impressionar. Especialmente quando comparada com linguagens

mais maduras como C++. O objetivo inicial do javascript nunca foi para ser utilizada em projetos de grande escala. Desenvolvedores superutilizam seus acertos. Existem muitas ferramentas de desenvolvimento designadas para aliviar as falhas do javascript como transpiladores. Seu ciclo de desenvolvimento é demorado pois necessita da aprovação de muitas partes. Tipicamente javascript é criado separadamente do html, através da tag

. Quando os navegadores encontram essa tag eles fazem a requisição para o servidor para injetar o código no documento. Isso pode ser problemático para projetos que incluam vários scripts.

Javascript se tornou portátil e isso é um componente chave no desenvolvimento de jogos.

5.9 ALTERNATIVAS AO JAVASCRIPT

Abaixo seguem algumas tecnologias que servem de alternativa ao Javascript.

###5.9.1 TYPESCRIPT Conhecido como uma versão estendida do Javascript que compila para javascript normal. Isso significa que os desenvolvedores podem continuar escrevendo Javascript normalmente. Typescript oferece classes, interfaces e módulos.

###5.9.2 DART Google. DartVM é uma máquina virtual que está embutido no google-chrome. Significante melhorias em performance quando comparado ao Javascript. Existe o dart2js que compila código em dart para javascript.

5.10 DOCUMENT OBJECT MODEL (DOM)

É uma plataforma e interface agnóstica a linguagem que permite os 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.

5.11 CANVAS

A nova tag

define um layer gráfico em documentos HTML que pode ser desenhado através de Javascript. Permite desenhar diagramas, gráficos e animações [7]. É 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 uma aplicativo híbrido que preenche a fraca implementação de OpenGL nos dispositivos móveis possibilitando se desenvolver em WebGL. ##5.12 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 jogos 3D. Versão da especificação atual? ##5.13 VIDEO ##5.14 AUDIO Audio é um componente vital para oferecer grande satisfação aos usuários de jogos. Provê feedback e imerge o usuário. Efeitos de som e música podem servir como mecanismo. Jogadores tem baixa tolerância a volume, deve ser utilizado com cautela. ###5.14.1 TAG AUDIO A tag 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 audio do navegador. Existem muitas discrepâncias entre os formatos aceitáveis pelos navegadores. É um tanto limitada quanto comparada ao áudio de múltiplos canais disponibilizados por SDKs nativas. ###5.14.2 API DE AUDIO É uma interface de audio experimental para Javascript. Provê maior flexibilidade na manipulação de audio. Essa tecnologia é muito mais nova do que a tag audio. FORMATOS DE ÁUDIO ##5.15 CAMERA ##5.16 ENTRADA DE COMANDOS Na construção da grande maioria dos jogos é 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 é 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 browser do cliente existem duas possibilidades. Pode-se implementar testes para cada funcionalidade utilizada abordando os detalhes de implementação de cada uma ou então fazer uso de alguma biblioteca especializada neste processo, o Modernizr é uma opção open-source deste tipo de biblioteca, este gera uma lista de booleanos sobre grande variedade dos recursos HTML5, dentre estes, geolocalização, canvas, áudio, vídeo e local storage. ##5.17 CACHE Aplicações offline. Algumas tecnologias desta classe são: ##5.18 OFFLINE E ARMAZENAMENTO ###5.18.1 LOCAL STORAGE Também conhecido como WebStorage na especificação do HTML5. Provê uma forma de armazenar os dados como chave-valor dentro do navegador. Os dados são persistido mesmo que o navegador seja fechado. ###5.18.2 WEB SQL Simplesmente um banco de dados SQLite embebido no navegador. Permite tabelas relacionais. O tamanho padrão do banco de dados é 5Mbytes e pode ser extendido pelo usuário.

5.19 RECURSOS NATIVOS ATUALMENTE INDISPONÍVEIS PARA O HTML5

- Suporte à câmera;

- Suporte à calendário; ##5.20 DEBUG ###5.20.1 WEINRE Debugger remoto depreciado. 5.21 TECNOLOGIAS POLYFILL Acarretando assim, que muitos browsers não implementam algumas funcionalidades, completa ou parcialmente especificadas, daí surge a necessidade dos polyfills (tecnologias de preenchimento de lacunas) para implementar estas camadas.

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. ##5.22 FER-
 RAMENTAS ###5.22.1 NODEJS Permite rodar Javascript fora do browser. Utiliza um modelo dirigido à eventos sem bloqueio, tornando-o rápido e eficiente. ##5.22.2 SISTEMAS DE BUILDING Arquivos javascript são requisitados do servidor assincronamente. Isso pode levar a tempos de requisição pouco desejáveis. Uma saída seria escrever o código em apenas um arquivo mais isso leva a gerência de código bagunçada. A saída mais comum entre desenvolvedores é utiliza rum a ferramenta que junta todos os arquivos e disponibiliza apenas um para o usuário. ##5.22.3 GRUNT Aplica as modificações separadamente em cada arquivo. ##5.22.4 GULP Utiliza o conceito de streams para aplicar todas as modificações sobre um arquivo de uma vez só. Minify, obfuscation ##5.22.5 SOURCE MAPS Para encontrar os arquivos minificados a fim de ajudar o desenvolvedor a debugar a aplicação. ##5.22.6 MINIFY Remover caracteres desnecessários do javascript como espaços vazios, diminuindo o tamanho dos nomes, fazendo o tempo de loading diminuir. ##5.22.7 GERENCIADORES DE PACOTES ##5.22.8 BOWER Package manager para a web ##5.22.9 NPM Package manager para o NODE ##5.23 DISPONIBILIZAÇÃO DA APLICAÇÃO Links com manifestos

5.23.1 INSTALAÇÃO

Este método é benéfico pois possibilita ao usuário a mesma experiência ao adquirir uma aplicação normal. Este tipo de aplicação é comumente referido como “híbrido”.

5.23.1.1 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. ###5.23.1.2 PHONEGAP

5.23.2 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.

###5.24 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.

A 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 receberá.

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

5.24.1 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 é assim que se desenvolve.

5.25 FEEDBACK

6 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

7 METODOLOGIA

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 às 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)

8 RESULTADOS

Abaixo constam as limitações encontradas durante a pesquisa e concepção do jogo ##8.1 LIMITAÇÕES 1. VERSÕES A grande maioria dos dispositivos atualmente no mercado utilizam obsoletas de seus softwares. Isso dificulta o desenvolvimento. Se a tecnologia de tradução para o browser utilizar o 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. VIDEO Codecs

5. ASSETS

Trafegar muitos assets deixa o sistema lento.

Contorno Utilizando páginas de carregamento e/ou cache;

5. UI

É muito custoso desenvolver uma interfaces que pareçam nativas para cada dispositivo sem a utilização de plugins e ferramentas especializadas. Em termos gerais, trabalhar com proporções é positivo. Não obstante há casos, como o dos botões de certo e errado que a proporções ficam exageradas, nesses casos a utilizada de max-width é uma solução conveniente.

6. PERFORMANCE

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.

10. CAMERA

Falar sobre câmera.

Limitação Desktop/Firefox Desktop/Google Chrome Smartphone/Android

9 CONCLUSÕES

Não pude testar todos os métodos e ferramentas e versões à disposição, 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 precisarão 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. Só poderemos considerar o HTML como uma jkkespecificação pronta quando for possível fazer tudo o que se faz nativamente com os dispositivos através de uma API web padronizada.

9.1 TRABALHOS FUTUROS

EMACSCRIPT 7

10 CRONOGRAMA

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

Início Término 1. Pesquisa explanatória sobre o estado da arte das tecnologias e conceitos. Documento 12/08 30/08 2. Identificar as tecnologias que melhor satisfazem as necessidades do projeto. Documento 01/09 02/09 3. Análise do jogo Documento 02/09 17/09 4. Construção de protótipo Documento, Software 17/09 10/10 5. Construção de versão final Documento, Software 10/10 30/10 7. Exportação dos binários específicos de plataforma Software 1/11 2/11 8. Disponibilização da aplicação online - 2/11 3/11 9. Levantamento das experiências obtidas como HTML5, e especificidades da implementação do controle de movimento e da interface de usuário. Documento 4/11 7/11 10. Criação de comparativo de usabilidade dentre as plataformas Documento 7/11 15/11 11. Análise dos resultados e conclusões Documento 15/11 30/11 Escrita do TCC Documento 12/08 05/12

BIBLIOGRAFIA

LEMES, David Oliveira. Games independentes. Pontifícia Universidade Católica de São Paulo 2009.

Referência 3 é uma ótima fonte de explicação de como tecnologias relacionadas ao assunto funcionam.

Referência 2 é uma ótimo trabalho para pegar ideias sobre como escrever as coisas.

Referência 7 is a good font from feature detection systems.

HTML5: A blessing or a curse <http://www.develop-online.net/tools-and-tech/html5-a-blessing-or-a-curse/0117393>

5 common mistakes that html5 developers make <http://www.toptal.com/html5/top-5-mistakes-that-html5-developers-make>

GLOSSÁRIO

API -

ANEXOS

10.1 CONVERSORES PARA HTML5 Além da possibilidade de escrever em HTML, pode-se optar pela alternativa de utilizar-se um conversor de linguagens.

10.2 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE PARA A CONSTRUÇÃO DE GAMES

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

- OpenUP: este é bem detalhado e de característica iterativa e incremental. Gerando assim, um levantamento mais apurado dos riscos, requisitos e outros detalhes do sistema e a criação incremental do sistema, com requisitos maleáveis;
- Cascata: processo antigo, caracteriza-se por ser pouco maleável aos requisitos mapeados posteriormente ao processo de análise;
- Processo ágil - SCRUM: sua utilização é flexível e sendo um método ágil especifica pouca documentação, ou como dizem, somente a documentação necessária, este processo é bem conhecido e aceito na comunidade de desenvolvimento de software. Suas principais características são: divisão do processo de desenvolvimento através uma série de iterações chamadas sprints. Cada sprint consiste tipicamente em duas a quatro semanas. É bem aplicado a projetos que mudam constantemente e que demandam rápidas adaptações;
- Processo ágil – XP: tem muitas características similares ao SCRUM por este também ser um processo ágil. Dentre suas especificidades destaca-se: versões frequentes, pequenos ciclos de desenvolvimento que buscam aumentar a produtividade, introduzem checkpoints onde os clientes podem agregar novas funcionalidades;

10.3 AMBIENTES PARA DESENVOLVIMENTO HTML5

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 (client side) e dá suporte à comunicação com o servidor através de RPCs Remote Procedure Calls (ou procedimento de chamadas remotas). Ele não é um framework para aplicações clássicas da web, pois deixa a implementação da aplicação web parecida com implementações em desktop. Este é utilizado em muitos produtos de grande porte como o Google Adwords e Google Wallet. Outra característica interessante é que a plataforma opera sobre a licença Apache versão 2; - construct 2 - é um editor na nuvem focado para usuários sem conhecimento prévio em programação orientado a comportamento; - PlayCanvas - é uma plataformas para a construção de jogos 3D na nuvem, desenvolvida com foco em performance. Permite a hospedagem, controle de versão e publicação

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

10.4 HTTP

FRAMEWORKS DE DESENVOLVIMENTO DE JOGOS EM HTML5;

10.5 FRAMEWORKS PARA DESENVOLVIMENTO DE JOGOS HTML5

Com o intuito de simplificar o processo para os desenvolvedores, auxiliando-os a focarem-se apenas nas soluções que estão desenvolvendo, foram criados os frameworks para desenvolvimento de jogos. Não obstante, o intuito deste trabalho é desenvolver um jogo sem auxílio de frameworks pois estes muitas vezes escondem possíveis limitações, desenvolvendo *workarounds* próprios. - enchant.js: dentre suas funcionalidades constam: orientação à objetos, orientado à eventos, contém um motor de animação, suporta WebGL e Canvas, etc; - three.js: considerada leve, renderiza WebGL e Canvas, arquitetura procedural; - quintus: bom para plataformas 2D - limeJs: bom para 2d

10.6 INTERFACE E ESCOLHAS DE DESIGN Mobile first