

UM ESTUDO COMPARATIVO ENTRE
FLASH LITE E J2ME
PARA DESENVOLVIMENTO DE JOGOS
TRABALHO DE GRADUAÇÃO

Aluna: Laís Xavier (lx@cin.ufpe.br)

Orientador: Prof. Fabio Bueno Queda da Silva (fabio@cin.ufpe.br)

29 de março de 2007

ASSINATURAS

Este Trabalho de Graduação é resultado dos esforços da aluna Laís Xavier, sob a orientação do professor Fabio Queda, sob o título inicial de “*Um Estudo Comparativo Entre Flash Lite e J2ME para Desenvolvimento de Jogos*”. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

Laís Xavier

Fabio Bueno Queda da Silva

*"Don't worry about a thing,
'Cause every little thing gonna be all right."*

(Bob Marley)

À minha família, pai, mãe, irmã e
amor.

Agradecimentos

Esses agradecimentos serão escritos para que as pessoas queridas saibam o quanto são importantes no meu desenvolvimento.

Agradeço aos meus pais (Gerino e Angélica) pela vida, educação e formação. Duas pessoas muito importantes que são meus exemplos de sinceridade, honestidade e garra. Agradeço também minha irmã querida que sempre esteve por perto, me dando força e mostrando o significado do amor fraterno. Não conseguiria nada na vida sem vocês.

Agradeço a Mano, meu namorado, noivo, companheiro, por todo respeito, amor, paixão, compreensão pelas minhas ausências, por compartilhar dos meus sorrisos e lágrimas e por aceitar enfrentar tantos desafios ao meu lado.

Agradeço também a todos os meus amigos que passaram e ficaram durante a minha vida. Vocês são muito importantes para mim.

Faço um agradecimento especial agora a Ricardinho, Teoria e Leleo, sem vocês eu não conseguiria terminar esse trabalho. Gostaria que soubessem o quanto vocês me surpreenderam com uma grande amizade e solidariedade. Nunca mais esquecerei a ajuda.

Aos meus queridos colegas de faculdade, Carol, Dudu, Gera que sempre trabalharam comigo e foram excepcionais sempre! Aos outros colegas de faculdade e também de farra Léo, Zé, Cabelinho, Joabe, Renan, Carlinhos, Farley, Vitão, Ronaldinho, Allan, Gustavo, Acerola, Thiago, BPE, MTA, Martinelli.

Mais um agradecimento aos amigos de faculdade, que me ajudaram a tornar esse trabalho possível, Forró, Marcellus, Bengt, Dudu, Luís e Igor vocês foram maravilhosos.

As minhas queridas amigas de graduação também, Aninha, Carina, Jeane, Ulli, Carol, Raquel, Sylvinha, nós somos vencedoras, falae!

Agradeço a todos os que me ajudaram na minha formação como profissional, Tairone, Alysson, Marcílio e Leo. Vocês tiveram e têm uma participação muito especial na minha vida. Fico muito feliz pelo privilégio de ter vocês como mentores e amigos.

Aos meus amigos queridos que fizeram parte da minha vida profissional e que hoje estão no meu coração. Galera da Educandus, Brunilda, Zé Mário, Paulão, Renatoviski, Polly, Cynthia, Elaine, Val, Luís, Saulo, Cláudio, aos meus sobrinhos, Mário, Caio, Jaum e Thiago (hehehehe). Galera da Policentro, Stenin, Amandita, Brunildo, Daniel Parede de Igreja, Rodrigo, Thales, Thiago (alvirrubro), Luís, Ivanzito, Vêu, Chico e Marcelo. Galera da Icorp, Babita, Júlio, Lea, Igor e Grasi.

Sem esquecer àqueles que são meus mestres, Fábio Queda, Geber Ramalho por todo o apoio acadêmico, além de outros como Anjolina, Ruy, Fernando Fonseca, Sílvio Melo, Sílvio Meira, e todos os professores do CIn.

Finalmente, agradeço aos amigos da vida , a minha amiga Carol Maga, quanta saudade, à toda equipe feminina maravilhosa (Déh, Maga, Beira, Didi, Su, minha irmã linda), meus vizinhos, irmãos de coração (Fabiano, Gordo, Mago, Priscila, Taci, Tércio, Fredinho, João).

A todos que esqueci, mil desculpas, minha memória, como vocês sabem, não é das melhores.

Resumo

Este trabalho concentra seus maiores esforços em comparar as principais características de duas plataformas de desenvolvimento para dispositivos móveis, são elas: Flash Lite e J2ME.

Para isso, foi desenvolvido um jogo em cada uma das tecnologias, para expor, tendo base em dados técnicos, qual delas é mais adequada para determinada situação durante o desenvolvimento de jogos.

Dessa forma, foi também desenvolvido um *framework* de comparação definindo as variáveis e os critérios de avaliação que relacionam como será aferido o desempenho de cada uma das tecnologias.

Também serão expostas algumas características de outras linguagens de desenvolvimento de jogos para dispositivos móveis, tais como BREW.

Por fim, um relatório e um estudo experimental serão apresentados para melhor ilustrar a comparação. Vários pontos positivos e negativos são demonstrados. Neste momento, alguns trabalhos futuros são sugeridos como forma de melhorar e refinar a comparação.

Palavras chave: Flash Lite, J2ME, Dispositivos Móveis, Comparação.

Índice

Agradecimentos	- 5 -
Resumo	- 7 -
1. Introdução	- 11 -
1.1 Problema e justificativas	- 11 -
1.2 Objetivos.....	- 12 -
1.3 Trabalhos Relacionados	- 13 -
1.4 Metodologia	- 13 -
1.5 Estrutura Geral	- 14 -
2. Jogos em Dispositivos Móveis	- 15 -
2.1 Um Pouco de História	- 15 -
2.2 Mercado	- 21 -
3. Características das Plataformas de Desenvolvimento.....	- 24 -
3.1 J2ME	- 24 -
3.1.1 O que é J2ME.....	- 25 -
3.1.2 Configurações.....	- 26 -
3.1.3 Perfis	- 28 -
3.1.4 Arquitetura	- 32 -
3.1.5 API's J2ME	- 33 -
3.1.6 Resumo.....	- 33 -
3.2 Flash Lite	- 34 -
3.2.1 O que é Flash Lite	- 35 -
3.2.2 Características.....	- 36 -
3.2.3 Arquitetura	- 40 -
3.3 Resumo das Características	- 44 -
4. Comparação entre as plataformas	- 45 -
4.1 Variáveis.....	- 45 -
4.1.1 Tempo.....	- 46 -
4.1.2 Complexidade.....	- 47 -
4.1.3 Custo de desenvolvimento.....	- 48 -
4.1.4 Percepção do usuário.....	- 49 -
4.2 Critérios de Avaliação	- 49 -
5. Estudo Experimental	- 50 -
5.1 Resultados	- 50 -
5.2 Análise dos resultados	- 51 -
6. Considerações Finais	- 56 -
6.1 Conclusões	- 56 -
6.2 Trabalhos Futuros	- 57 -
Referências	- 59 -
Apêndice A – Documento de Game Design	- 63 -
Apêndice B – Questionário	- 75 -

Índice de Figuras

Figura 1 – Microvision.....	- 17 -
Figura 2 – Snake.	- 18 -
Figura 3 – N-Gage.	- 19 -
Figura 4 – Dispositivos móveis.	- 20 -
Figura 5 – Jogos multiplayer, online e via rede bluetooth.	- 20 -
Figura 6 – Média de preço de venda de jogos nos EUA.....	- 22 -
Figura 7 – Mercado Mundial de Games Wireless por Grupo de Países.	- 23 -
Figura 8 - Plataformas de desenvolvimento no Brasil.....	- 24 -
Figura 9 – Classificação dos dispositivos segundo suas semelhanças funcionais.	- 26 -
Figura 10 – J2ME Configurações e perfis.....	- 31 -
Figura 11 – – J2ME Arquitetura genérica.	- 32 -
Figura 12 – API’s oferecidas em J2ME.	- 33 -
Figura 13 – Iriver - dispositivo com interface desenvolvida em Flash Lite.	- 34 -
Figura 14 – Exemplos de dispositivos que rodam Flash Lite.	- 35 -
Figura 15 – Um exemplo de aplicação com Flash Lite 2.0.....	- 36 -
Figura 16 – Exemplo de aplicação que carrega dados externos XML.	- 37 -
Figura 17 – Exemplos de uma aplicação que utiliza dados persistentes.	- 37 -
Figura 18 Aplicação com carregamento de mídias dinâmicas.	- 38 -
Figura 19 – Exemplo de aplicação com edição de texto.	- 39 -
Figura 20 – Exemplos de aplicação que desenha formas dinamicamente... -	39 -
Figura 21 – Flash Lite Arquitetura genérica.....	- 41 -
Figura 22 - Flash Lite 2.0Arquitetura detalhada.....	- 42 -

Índice de Tabelas

Tabela 1 – Tabela de resumo de características de J2ME e Flash Lite.	- 44 -
Tabela 2 – Tabela com variáveis do framework de comparação.	- 46 -
Tabela 3 – Tabela de resultados.	- 51 -

1. Introdução

Dispositivos Móveis é o termo utilizado para identificar dispositivos que podem ser operados à distância ou sem fio. Dispositivos que podem ser desde um simples BIP, até os mais modernos Pocket PCs [LinhaCódigo04]. Fabricantes desses dispositivos, operadoras de rede e desenvolvedores de aplicativos têm prestado mais atenção no mercado que está emergindo a partir dos dispositivos móveis. Apesar do interesse em uma indústria que pode ser bastante lucrativa, eles enfrentam muitos obstáculos complexos para oferecer aplicações e serviços aos usuários de dispositivos móveis.

1.1 Problema e justificativas

Dispositivos móveis é uma indústria que está crescendo e que provê novos produtos ao mercado constantemente, sempre existe um modelo mais moderno, com uma tela maior, mais memória, largura de banda mais alta e um CPU mais rápido, além também de trazer na maioria das vezes cores novas aos dispositivos.

Logo, observa-se que o número de consumidores desses dispositivos aumenta a cada dia e, além disso, procuram certas particularidades nos dispositivos e nas operadoras de serviços. Como resultado disto, a necessidade dos fabricantes e operadores de rede de trazerem maiores experiências ao usuário que venham fazer diferença aos seus produtos, tornando-os mais competitivos, é maior que nunca.

O grande desafio que se tem em mente é encontrar a melhor maneira de entregar um produto de qualidade, com desenvolvimento rápido e que

tenha uma boa relação de custo e eficiência. Neste contexto, obter sucesso neste desafio é crítico, é necessário escolher as pessoas e as tecnologias certas, bem como os designers e desenvolvedores qualificados que trabalhem com as melhores ferramentas.

Em relação às tecnologias, existem algumas plataformas para o desenvolvimento em dispositivos móveis tais como J2ME, Brew, Flash Lite, MoPhun. Neste trabalho, se comparam duas plataformas, J2ME e Flash Lite. A primeira é uma tecnologia com uma grande aceitação, que já está estabelecida no mercado e também a mais utilizada para desenvolvimento de aplicativos. A segunda é uma tecnologia relativamente nova e que tem uma proposta de trazer mais facilidade e agilidade no processo de desenvolvimento e *porting* [Talbot06], assim como trazer uma melhor experiência visual para os usuários e convergência entre dispositivos tais como a internet e o celular.

A comparação entre as plataformas será feita baseada no desenvolvimento de um aplicativo específico, um jogo, cujo game design encontra-se no Apêndice A deste documento.

1.2 Objetivos

Este trabalho tem por objetivo realizar um estudo comparativo entre duas tecnologias específicas (Flash Lite e J2ME) para desenvolvimento em dispositivos móveis, focando no desenvolvimento de jogos, para, tendo base em dados técnicos, levantar hipótese sobre qual das tecnologias é mais adequada para determinada situação durante o desenvolvimento de um jogo.

Levando em consideração que os jogos de PC e consoles estão migrando para o celular e em breve a convergência entre estas plataformas

será possível, é muito importante que as empresas de desenvolvimento de jogos tenham dados para escolher a melhor tecnologia para suas necessidades.

De acordo com Marcelo Carvalho, presidente da ABragames (Associação Brasileira das Desenvolvedoras de Jogos), hoje os jogos para celular estão cada vez mais complexos e num futuro próximo direcionarão a fabricação dos aparelhos, como fazem com o PC [Abragames05].

1.3 Trabalhos Relacionados

Existem algumas outras comparações feitas entre plataformas de desenvolvimento em dispositivos móveis, tais como:

- **Mobile Device Platforms** - Uma Comparação entre RIM Blackberry 4.0 e Microsoft Windows 5.0 Pacote de Empreendimento em Soluções Móveis para mensagens e características de segurança [Goggin05]
- **How Smartphone Platforms Compare** - O guia de comprador de tecnologia para BlackBerry, Linux, Mac OS X, Palm, Symbian, e Windows plataformas Móveis. [Malykhina07]
- **Mobile Platform Benchmarks** - Uma metodologia para avaliar a computação dos dispositivos móveis. [McKenna00]

Nenhum destes estudos faz uma comparação de J2ME e Flash Lite, que é o objetivo deste trabalho.

1.4 Metodologia

A metodologia utilizada para este trabalho foi a pesquisa exploratória [Uniformg04], que teve a finalidade de proporcionar maiores informações

sobre as tecnologias utilizadas; facilitar o desenvolvimento do *framework* de comparação entre elas e definir os objetivos da comparação.

A pesquisa envolveu um levantamento bibliográfico, elaboração de um *framework* de comparação, o desenvolvimento de um aplicativo, aplicação de um questionário com os desenvolvedores, que se encontra no Apêndice B deste documento, e a análise dos dados.

Foi desenvolvido um game design (Apêndice A) para o desenvolvimento do jogo (aplicativo utilizado para fazer o estudo), a seguir, foram montadas duas equipes de desenvolvimento distintas, uma que fez a parte do desenvolvimento relativo à plataforma Flash Lite, e outra que fez a parte do desenvolvimento relativo à plataforma J2ME.

Cada grupo de desenvolvedores tinha conhecimentos avançados da linguagem base de cada uma das tecnologias que usaram pra desenvolver, ActionScript e Java respectivamente.

Ao final do desenvolvimento, foi aplicado um questionário com os desenvolvedores tendo em vista a coleta dos dados para o preenchimento do *framework* de comparação e, posteriormente, a análise desses dados.

1.5 Estrutura Geral

Este trabalho está dividido em seis capítulos e dois apêndices. Neste primeiro capítulo encontra-se a descrição da estrutura do trabalho, o problema encontrado e suas justificativas, os objetivos, bem como a metodologia utilizada para o seu desenvolvimento. No capítulo 2 é apresentada uma introdução sobre o contexto em que este trabalho está

inserido, o “Mundo dos Jogos em Dispositivos Móveis”. O capítulo 3 mostra as características das linguagens que serão comparadas: Flash Lite e J2ME. No capítulo 4 é apresentado o framework de comparação elaborado, ou seja, é explicado de forma detalhada quais serão as variáveis aferidas e quais os critérios de avaliação utilizados. Posteriormente, no Capítulo 5, é feito estudo de caso e a análise dos resultados do mesmo. Então, no capítulo 6 são apresentadas as conclusões sobre este trabalho, analisando vantagens e desvantagens das tecnologias comparadas, e sugerindo futuros trabalhos para melhorar e refinar a comparação. Finalmente, o Apêndice A traz o game design do jogo que foi desenvolvido para o estudo de caso e o apêndice B tem o questionário utilizado na coleta dos dados.

2. Jogos em Dispositivos Móveis

Este capítulo discute os aspectos históricos e mercadológicos dos jogos em dispositivos móveis, como forma de contextualizar e introduzir o ambiente em que este trabalho se concentra. A história será mostrada do ponto de vista dos dispositivos móveis, depois dos jogos e finalmente a junção dos dois.

2.1 Um Pouco de História

Foi realizada a primeira ligação pública utilizando-se um aparelho portátil, no dia 3 de abril de 1973. Na ocasião, o pesquisador da Motorola Martin Cooper ligou para um telefone fixo diretamente de uma esquina do centro de Nova Iorque, nos Estados Unidos. O aparelho utilizado pesava cerca de um quilo e media 25 cm de comprimento por 7 cm de largura, com uma bateria que se esgotava após 20 minutos de conversa. A inovação do início da década de 70 significou um importante avanço tecnológico em

relação aos telefones móveis que vinham sendo utilizados em automóveis desde os anos 40 [Junior06].

A invenção, entretanto, só passou a ser comercializada 10 anos mais tarde, em 1983, quando a própria Motorola lançou no mercado mundial o modelo DynaTAC 8000X. Apesar do alto preço inicial, cerca de quatro mil dólares, o aparelho foi rapidamente aceito pelos consumidores, que chegaram a se inscrever em listas de espera com milhares de nomes. A explosão imediata do consumo gerou uma severa disputa entre as empresas de telefonia, que passaram a investir pesado em busca de avanços tecnológicos que possibilitassem o aumento da qualidade e da quantidade de serviços e produtos, sempre tendo em vista a redução do custo final. [Junior06]

Enquanto isso em 1992, a Apple introduziu no mercado um handheld chamado Newton. O Newton chegou ao mercado com tela sensível ao toque, 1MB de memória total e capacidade de transmissão de dados de 38.5kbps. Este modelo não teve muita repercussão [Mendonça06].

Em 1996, a U.S. Robotics lançou o (Palm) Pilot 1000 e 5000, dispositivos que tiveram uma grande aceitação no mercado e lançaram as bases de toda uma plataforma de “Palms” que chegaram a atingir 80% do mercado mundial e existem até hoje. A U.S. Robotics foi adquirida pela 3Com, que depois desvinculou dela a empresa Palm Inc., sendo esta totalmente focada nesta nova plataforma de dispositivos [Mendonça06].

Também em 1996, começaram a surgir dispositivos baseados no Windows CE 1.0, da Microsoft, como o NEC MobilePro 200 e o Casio A-10. Até o lançamento do Windows CE 3.0 e da plataforma Pocket PC, em 2000, a plataforma Windows CE não teve grande aceitação do mercado. Mas a partir do Sistema Operacional Pocket PC 2000, embutido em dispositivos como o

HP Ipaq e o Compaq Jornada, esta plataforma ganhou aceitação do mercado e começou a crescer [Mendonça06].

Enquanto isso, a empresa Symbian foi formada em 1998 por alguns dos maiores fabricantes de celulares do mundo e a PSION, e entregou ao mercado o sistema operacional Symbian, que roda na maioria dos smartphones e handhelds da Nokia, e detém a maior fatia do mercado europeu atualmente. [Mendonça06]

Paralelo ao avanço dos celulares e dispositivos móveis em geral, a indústria de jogos começou a desenvolver tecnologias para jogos móveis. O primeiro dispositivo com jogo eletrônico foi chamado OXO, desenvolvido em 1972 pela Waco (Toymaker). O jogo que veio nesse dispositivo foi o jogo da velha. A exibição consistia em um grid de nove botões que poderiam virar vermelho ou verde quando pressionados.

O primeiro console em dispositivos móveis com cartuchos trocáveis era o Microvision projetado por Smith Criando, em 1979. [Wiki06] Por conta de algumas restrições de hardware, o Microvision foi descontinuado dois anos depois.



Figura 1 - Microvision.

A história dos Jogos em dispositivos móveis começou no início dos anos oitenta quando a Nintendo introduziu seus jogos portáteis em série "Games & Watch". Antes disso, a Nintendo introduziu um conhecido jogo de console no jogo portátil, o "Mario". "Games & Watch" trouxe para os jogadores uma possibilidade para jogar 70 jogos diferentes e desfrutou de uma grande popularidade [Semenov04]. Durante a próxima década mais produtos semelhantes foram postos no mercado. Para manter sua posição no mercado, a Nintendo introduziu um tipo novo de jogo de console: o Gameboy com cartuchos trocáveis. Gameboy era um sucesso enorme e dominou o mercado a tal extensão que se tornou um sinônimo de console para jogos portáteis. A Nintendo desenvolveu centenas de jogos para Gameboy e suas versões avançadas.

Em 1997 a Nokia desenvolveu um tipo novo de jogo que poderia ser jogado usando um telefone celular. O primeiro jogo foi o "Snake", embora muito simples tornou-se popular e a Nokia desenvolveu mais jogos semelhantes para seus dispositivos [Semenov04]. Àqueles jogos ficavam permanentemente instalados nos dispositivos, mas a situação mudou depois.



Figura 2 - Snake.

No final dos anos 90, surgiram novos tipos de jogos para plataformas móveis, desenvolvedores estavam procurando um modo para proporcionar para jogadores jogos multiplayer. Estes eram jogos WAP e jogos baseados em SMS's. Os jogos via WAP, era jogados usando os dispositivos que tivessem um browser wap (por exemplo, a 7100-série de Nokia), apesar de não tem muita popularidade na Europa devido a conexão lenta, mas teve êxito no Japão onde NTT DoCoMo estava vendendo jogos para I-mode. Em matéria de jogos baseados em SMS, ficaram populares usando broadcastings de TV. [Semenov04]

Em 2001, a história de jogos para dispositivos móveis deu outro passo à frente, quando os dispositivos móveis foram equipados com telas coloridas e a possibilidade de fazer download de jogos foi colocada no mercado. Fabricantes de dispositivos móveis perceberam que para satisfazer os consumidores e deixá-los mais interessados em seus dispositivos eles teriam que desenvolver telefones mais avançados (chamado smartphones) capazes de apoiar gráficos 3D e sons naturais.

A Nokia Em 2003 introduziu seus dispositivos N-Gage e jogos on-line sem fios com N-Gage Arena. O N-Gage é um dispositivo de jogo completo com cartuchos trocáveis e funcionalidades de smartphone. Foi lançado com uma grande campanha de marketing, mas não recebeu a popularidade esperada, foram vendidos apenas 1.5 milhões de unidades no início. Em 2004 a Nokia introduziu uma plataforma de N-Gage redesenhada usabilidade melhorada e preço baixo.



Figura 3 – N-Gage.

Outros grandes fabricantes também entram no mercado com produtos semelhantes que apóiam jogos móveis avançados. Assistentes Digitais pessoais (PDAs), console para jogos portáteis, e smartphones fabricados por diferentes empresas.



Figura 4 – Dispositivos móveis.

Até recentemente, o mercado de jogos para dispositivos móveis concentrou jogos para um único jogador, mas atualmente a situação está mudando com introdução de jogos multiplayer. Jogos multiplayer mudam a experiência do jogador em jogos para dispositivos móveis, porque os jogadores podem desafiar pessoas reais em vez de máquinas. Isto é particularmente importante para jogadores jovens que jogam para se socializar. Como podemos ver até agora, tipos diferentes de portais de Internet e fóruns estão proporcionando os jogadores oportunidades para comunicar entre si, estimulando o interesse deles/delas no processo do jogo.



Figura 5 – Jogos multiplayer, online e via rede bluetooth.

A popularidade de jogos em dispositivos móveis entre usuários também dependerá fortemente de evolução adicional de hardware. Os parâmetros seguintes formarão a base desta evolução: tamanho de tela, resolução, largura da banda de rede, quantia de memória do dispositivo e o

custo. Com introdução hardware mais avançados os desenvolvedores de jogos para dispositivos móveis poderão acrescentar maiores detalhes aos jogos, fazendo-s mais atraentes para os usuários.

2.2 Mercado

O mercado de jogos em dispositivos móveis encontra-se em pleno crescimento. Anualmente, é publicado um estudo – chamado *ESA Facts* - pela *Entertainment Software Association* (ESA), que busca identificar os mais relevantes aspectos com relação ao desenvolvimento de software voltado para o entretenimento. Na edição de 2006 do ESA descobriu-se que 32% dos americanos jogam em dispositivos móveis, um crescimento de 20% em relação a 2002 [ESA06].

Virtualmente todo segmento do entretenimento e indústria de mídia está sendo influenciado por distribuição de conteúdo on-line ou para dispositivos móveis ou através de tecnologias digitais em geral. [PWC06]

Enquanto o uso pelo consumidor e a incidência de download de jogos é uma métrica importante para medir a indústria de jogos em dispositivos móveis, segundo um estudo realizado pelo NPD – grupo que estuda o comportamento dos consumidores da indústria e faz pesquisa de mercado–, a renda também é uma medida precisa do mercado. Por esta medida, jogos para dispositivos móveis estão bem no mercado. No quarto semestre de 2006 a indústria levou em quase \$107 milhões em renda relacionada a jogos para dispositivos móveis. Esta renda só foi batida pelo serviço de mensagem (SMS) que alcançou uma renda de \$482 milhões durante o quarto semestre de 2006 e downloads de toques que alcançou \$172 milhões. [NPD06]

Também segundo o estudo, nos últimos três trimestres de 2006, o preço médio de vendas (do inglês average sales price-ASP) de um jogo para dispositivos móveis não superou \$3.79. É importante lembrar que muitos desses jogos são grátis, e quando este fator é tirado da equação, o ASP para jogos sobe para \$4.41.

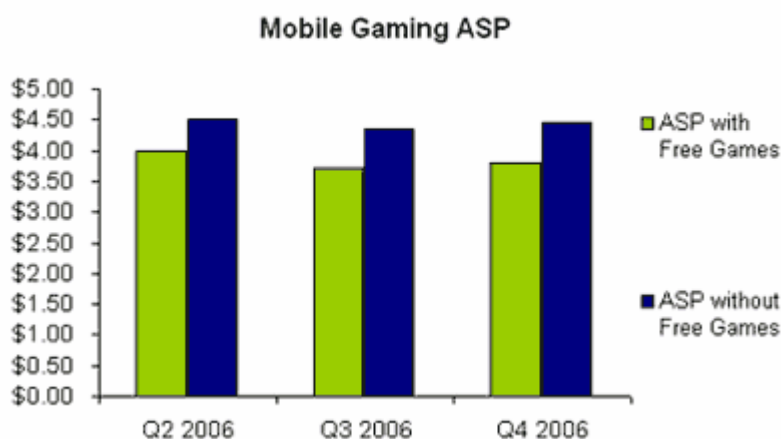


Figura 6 – Média de preço de venda de jogos nos EUA.

O ministério de Ciência e Tecnologia (MCT) juntamente com Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) fizeram uma pesquisa intitulada “Tecnologias de Visualização na Indústria de Jogos Digitais” que traz uma análise do mercado mundial e brasileiro de jogos. Segundo a pesquisa, o mercado mundial de jogos está concentrado na Ásia/Pacífico, nos EUA e na Europa. A Ásia/Pacífico representava 38% do mercado (gasto de US\$ 8,4 bilhões), os EUA respondiam por 34% ou US\$ 7,6 bilhões e a EMEA (Europa, Oriente Médio e África), por 24% ou US\$ 5 bilhões. As três regiões distribuem seu gasto nas quatro plataformas de forma semelhante à média mundial (cerca de 75% para console, 15% para PC e 10% para on-line e wireless). Complementam a lista de países pesquisados pelo relatório, a América Latina (AL) com 2% do mercado ou US\$ 0,5 bilhão e o Canadá, com 3% ou US\$ 0,78 bilhão.).[MCT05]

Também segundo a pesquisa, o mercado mundial de jogos para dispositivos móveis atingiu US\$ 738 milhões em 2003. A região da Ásia/Pacífico responde por quase metade (49%) dos gastos com dispositivos

sem fios (US\$ 367 milhões), por causa da grande difusão de telefones celulares no Japão e na Coréia do Sul. De acordo com a pesquisa, jogos para consoles portáteis representam 25% do gasto total em jogos no Japão. [MCT05]

A EMEA é o segundo maior mercado de wireless (29%) representando US\$ 208 milhões. O grupo de países tem 3,9 milhões de assinantes de jogos. Em 2008 é esperado que o percentual de usuários de jogos seja de 20%, quando se estima que haja 470 milhões de usuários de celular [PWC06]. O segmento de games para celular vem crescendo no Reino Unido, onde 40% dos usuários de celular jogam games.

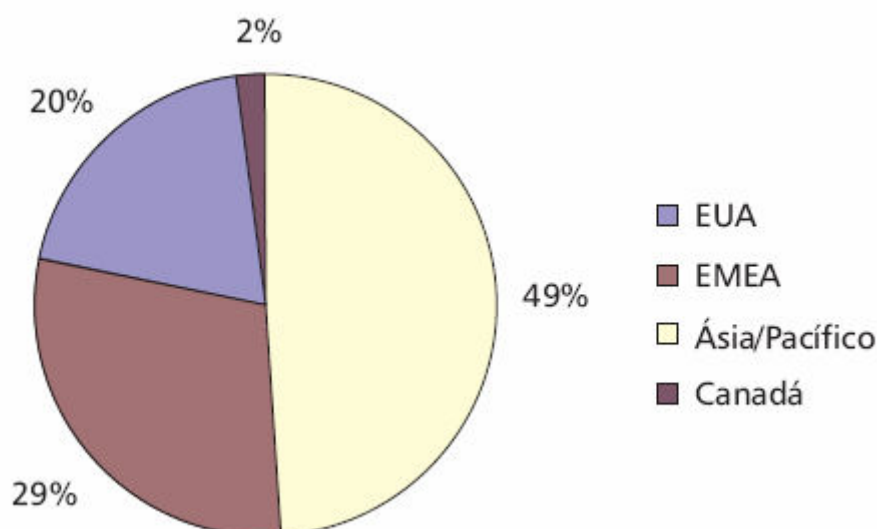


Figura 7 – Mercado Mundial de Games Wireless por Grupo de Países.

Os EUA respondem por 20% do gasto em games para wireless. O país conta com 150 milhões de usuários de celular. Em 2008, é esperado que 30% dos estimados 197 milhões de usuários de celular usem games. O Canadá, em decorrência de sua menor população, tem somente 11,8 milhões de usuários de celular, representando US\$ 16 milhões ou 2% do mercado mundial. Em 2008, é esperado que 25% dos usuários de celular sejam assinantes de jogos. [MCT05]

Já no mercado de jogos brasileiro, segundo a pesquisa da Associação Brasileira das Desenvolvedoras de Jogos Eletrônicos (Abragmes), as

desenvolvedoras brasileiras focam no mercado de PC (63%), mais tradicional. Em segundo lugar, vem o de celulares (22%). [Abragames05]



Figura 8 - Plataformas de desenvolvimento no Brasil.

3. Características das Plataformas de Desenvolvimento

Este capítulo visa demonstrar as principais características técnicas das duas tecnologias utilizadas neste estudo, J2ME e Flash Lite. Será demonstrada a arquitetura de cada uma, bem como será explicado de forma genérica o significado de cada uma das tecnologias. Ao final do capítulo terá um guia rápido com as características mais marcantes de cada uma das plataformas.

3.1 J2ME

Nesta seção, será exposto o J2ME assim como componentes que o compõem antes de, finalmente, rever a arquitetura, as configurações e os perfis. Com mais de dois milhões de programadores, Java é uma plataforma

muito popular entre os desenvolvedores [Wells]. O J2ME, por ser uma edição de Java, é uma linguagem que é bem aceita pelos desenvolvedores que trabalham com dispositivos móveis.

3.1.1 O que é J2ME

J2ME é uma plataforma para dispositivos pequenos e móveis que pretende eventualmente substituir vários produtos baseados do JDK 1.1 com uma solução unificada baseada em Java. O J2ME é uma coleção das especificações que definem um grupo de plataformas, cada qual apropriada para um subconjunto de dispositivos.

O subconjunto do ambiente de programação Java para um dispositivo particular é definido por um ou mais perfis, que estendem as potencialidades básicas de uma configuração. A configuração e o perfil ou os perfis que são apropriados para um dispositivo dependem ambos da natureza de seu hardware e do seu público alvo. [Topley02]

Abaixo a figura 9 mostra os dispositivos que o J2ME suporta agrupados por semelhanças funcionais.



Figura 9 – Classificação dos dispositivos segundo suas semelhanças funcionais.

3.1.2 Configurações

Uma configuração é uma especificação que define o ambiente de software para uma escala dos dispositivos definidos por um conjunto de características sobre a especificação [Topley02], tais como:

- Os tipos e a quantidade de memória disponível
- O tipo e a velocidade do processador
- O tipo de conexão de rede disponível no dispositivo

Uma configuração existe para representar a plataforma mínima para o dispositivo e não definir características opcionais. Os fabricantes são obrigados a implementar toda a especificação de uma determinada configuração, com o

objetivo de tornar as aplicações desenvolvidas para seu dispositivo o mais portátil possível. J2ME define atualmente duas configurações:

- Connected Limited Device Configuration (CLDC)
- Connected Device Configuration (CDC)

Abaixo detalharemos cada uma delas.

3.1.2.1 Connected Limited Device Configuration (CLDC)

CLDC é uma configuração utilizada para dispositivos mais limitados em relação a hardware. Uma plataforma típica de CLDC é um celular ou um PDA com aproximadamente 512 KB da memória disponível. Por esta razão, CLDC é associado com o Java wireless, que é conhecido por permitir que os usuários do celular comprem e façam download de aplicações Java conhecidas como MIDlets. [Topley02]

3.1.2.2 Connected Device Configuration (CDC)

O CDC dirige-se às necessidades dos dispositivos que se encontram entre CLDC e todos os sistemas que funcionam J2SE. Estes dispositivos têm mais memória (aproximadamente 2MB ou mais) e processadores mais capazes, portanto podem suportar um ambiente de software muito mais completo de Java. O CDC pode ser encontrado em PDAs, smart phones, set-top-boxes e outros. [Topley02]

Cada configuração consiste em uma máquina virtual Java e uma coleção das classes de Java que fornecem o ambiente de programação para o software de aplicação. O processador e as limitações da memória, particularmente em dispositivos de baixo nível, podem tornar impossível que uma máquina virtual de J2ME suporte todas as características da linguagem Java ou instruções do byte code e otimizações do software fornecido por uma

máquina virtual de J2SE. Conseqüentemente, as máquinas virtuais de J2ME são definidas geralmente como parte de uma especificação da máquina virtual Java. [Topley02]

É importante notar que as especificações da configuração não requerem implementações para usar uma máquina virtual específica. Os fabricantes estão livres criar sua própria máquina virtual, contanto que se cumpra com as exigências mínimas da especificação. As implementações de referência para uma máquina virtual de cada uma das configurações são:

- CLDC - Kilobyte Virtual Machine (KVM), uma funcionalidade reduzida da máquina virtual que incorpora o garbage collector que é otimizado para um ambiente memória limitada. [Topley02]
- CDC - CVM que implementa todas as características da máquina virtual de J2SE requeridas pela especificação do CDC. Entretanto, não inclui a tecnologia de HotSpot encontrada na versão 1.3 da máquina virtual de J2SE. [Topley02]

Uma configuração inclui também um subconjunto de classes de Java.

3.1.3 Perfis

Um perfil complementa uma configuração incluindo as classes adicionais que fornecem as características apropriadas a um tipo particular de dispositivo ou a um segmento de mercado vertical específico. Ambas as configurações de J2ME têm um ou mais perfis associados. Figura 10 mostra os perfis que são definidos atualmente ou estão no processo de definição e das configurações que eles são dependentes. [Topley02] Alguns perfis com suas características gerais são descritos abaixo:

3.1.3.1 *Mobile Information Device Profile (MIDP)*

Este perfil traz rede sem fio, componentes de interface com o usuário e o armazenamento local a CLDC. Este perfil é caracterizado primeiramente pelas limitações de tela e as facilidades no armazenamento em celulares, e fornece também uma interface com o usuário relativamente simples através de gráficos bitmap[Barbagallo04] e uma rede básica. O MIDP é o perfil mais popular e mais utilizado pelos desenvolvedores de J2ME porque é a base para Java Wireless Toolkit. [Topley02]

3.1.3.2 *PDA Profile (PDAP)*

O perfil PDA é similar a MIDP, mas o PDAs é caracterizado por ter telas melhores e mais memória do que celulares. O perfil PDA, oferece uma biblioteca de interface com usuário mais sofisticada e baseada na API de Java para alcançar características úteis do sistema operacional [Topley02].

O profile Personal Digital Assistant (PDA) é uma especificação da JSR000075 e executa sobre dispositivos com CLDC. Provê APIs para interface de usuário e armazenamento de dados em dispositivos pequenos, de mão e com recursos limitados como PDA's com os seguintes restrições de recursos [Sun072]:

- Não menos que 512K memória total (ROM + RAM) disponível para execução Java e bibliotecas, e não mais que 16 MB. [Sun072]
- Poder limitado, e tipicamente operado por bateria. [Sun072]
- Interface com usuário com variados graus de sofisticação, mas tendo telas com uma resolução total de pelo menos 20,000 pixels e caracteres de input. [Sun072]

3.1.3.3 Foundation Profile

O Foundation Profile estende o CDC para incluir quase todas as bibliotecas da versão 1.3. Criado pra ser a base da maior parte dos perfis utilizados com a configuração CDC. [Topley02]

A especificação do Foundation Profile, JSR000046, é satisfatória para dispositivos que precisam de suporte para uma plataforma Java habilitada para usar bem as funcionalidades de rede, mas não requer uma GUI. Também provê um perfil básico para outros perfis que precisam construir em sua funcionalidade somando um GUI ou outras características. [Sun072]

3.1.3.4 Personal Basis and Personal Profiles

O Personal Basis and Personal Profile adiciona a funcionalidade básica de interface ao Foundation Profile. Criado para ser usado nos dispositivos que têm uma potencialidade de interface que não é sofisticada, e conseqüentemente não permite que mais de uma janela seja ativa em algum momento. As plataformas que podem suportar uma interface mais complexa usarão uma instância do Personal Profile. [Topley02]

Combinado com o Connected Device Configuration (CDC), Personal Profile provê um ambiente completo para aplicações J2ME para produtos e dispositivos embutidos [Sun071]. As principais características do Personal Profile são:

- Compatibilidade total com AWT
- Suporte à programas modelo de aplicação applet

3.1.3.5 RMI Profile

O perfil do RMI adiciona as bibliotecas de invocação remota de métodos (Remote Method Invocation -RMI) de J2SE ao perfil the Foundation Profile. [Topley02]

O perfil RMI com especificação na JSR00066, suporta relacionamento entre aplicações RMI em cima de conexões de TCP/IP para aplicações escritas para o perfil Foundation profile. A especificação do perfil RMI também tem interoperabilidade com a API de RMI 1.2.x de J2SE ou maior. [Sun72]

3.1.3.6 GameProfile

O Game Profile, que está ainda no processo de ser definido, fornecerá uma plataforma para o desenvolvimento de jogos em dispositivos do CDC. Na altura da escrita, não está absolutamente certo se este perfil estará derivado do perfil da fundação ou baseado diretamente no CDC. [Topley02]

Abaixo, temos uma visão geral das configurações e os perfis associados a cada uma delas:

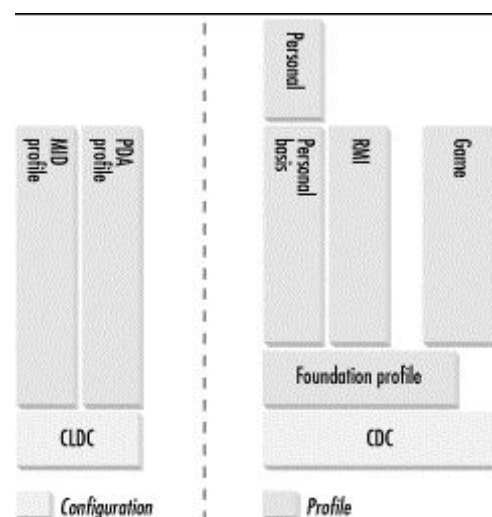


Figura 10 - J2ME Configurações e perfis.

3.1.4 Arquitetura

Até agora se cobriu uma variedade de informações sobre J2ME, sendo assim, será mostrado a partir de agora a arquitetura genérica do J2ME.

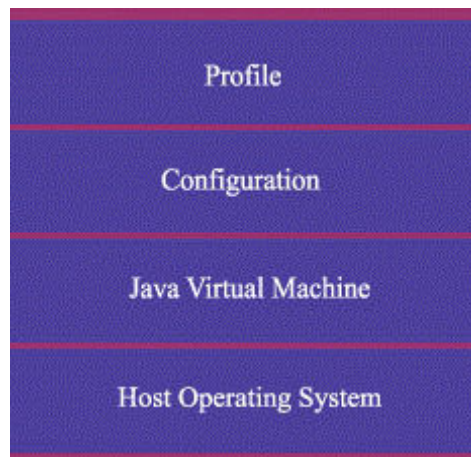


Figura 11 – J2ME Arquitetura genérica.

Começa com o sistema operacional (SO) como a base (ver figura 11), seguida pela máquina virtual (VM) [Muchow01]. A máquina virtual fará uma das duas formas:

- Para os sistemas que compilam com o CDC, será a máquina virtual “tradicional”; isto é, a mesma característica do J2SE.
- Para os sistemas que compilam com o CLDC, será o KVM ou uma máquina virtual que se encontre com as especificações requeridas pelo CLDC.

As bibliotecas do CLDC ou CDC estão na camada acima da máquina virtual. Os perfis estão na camada mais acima e são projetados para fornecer um toolkit para desenvolver aplicações para uma particular família de dispositivos [Muchow01], como visto anteriormente.

3.1.5 API's J2ME

Abaixo, a figura 12 representa genericamente as API's de J2ME que dão suporte ao desenvolvimento de aplicativos. As API's que estiverem contornadas com vermelho são as API's que têm implementação de referência.

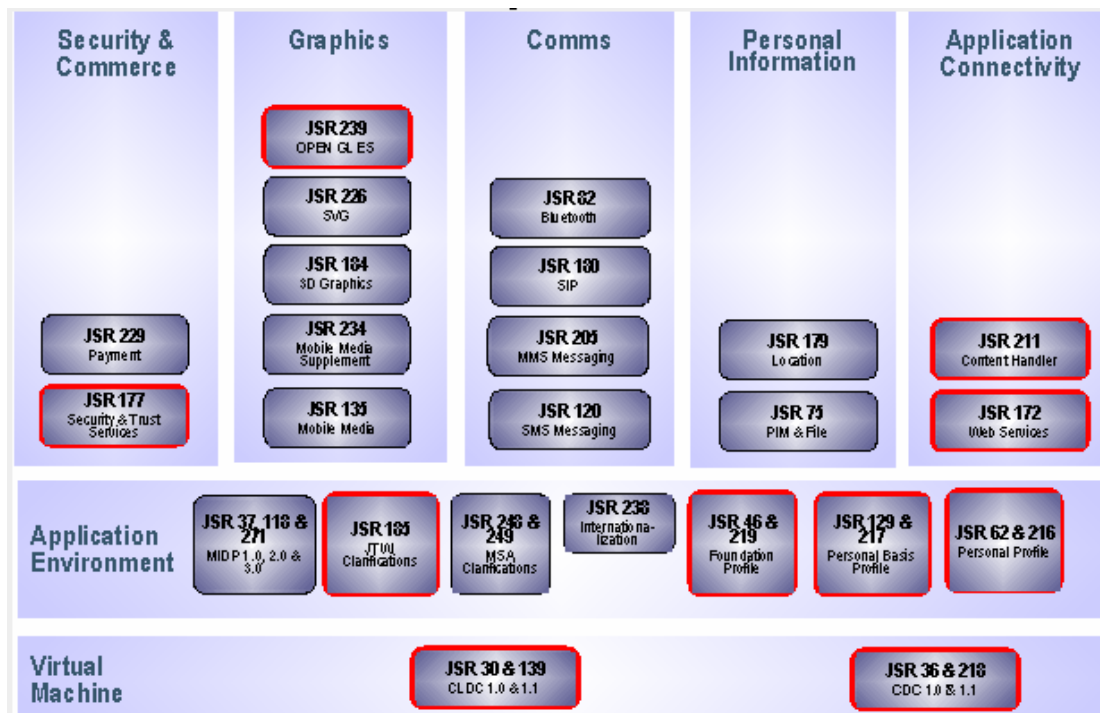


Figura 12 – API's oferecidas em J2ME.

3.1.6 Resumo

O J2ME foi criado para permitir o desenvolvimento de aplicações de Java para os dispositivos que não têm os mesmos poderes de processamento e memória encontrados em uma plataforma desktop típica. Os produtos podem incluir telefones celulares, PDAs, pagers, para nomear apenas alguns. J2ME é dividido em duas amplas categorias, conhecidas como configurações. O CDC é um conjunto de APIs para suportar dispositivos “fixos” tais como um set-top-box. CLDC é um conjunto de APIs para dispositivos que limitaram o poder de processamento, a tela e a memória. A maioria destes dispositivos será também móvel (por exemplo, telefones celulares e pagers). [Muchow01]

Uma configuração fica próxima a uma máquina virtual de Java. Para o CDC, a máquina virtual é compatível com a máquina virtual J2SE. O KVM, uma máquina virtual que considerasse os recursos limitados disponíveis nos dispositivos que coube esta configuração, foi desenvolvido para o CLDC. [Muchow01]

No alto das configurações estão os perfis do dispositivo. Aqui são encontradas as APIs para o projeto interface do usuário, suporte de rede e o armazenamento persistente.

3.2 Flash Lite

Neste capítulo, será exposto o Flash Lite assim como sua arquitetura e seus componentes. A Plataforma de Flash permite que os desenvolvedores entreguem aplicações que rodem em desktops e dispositivos móveis, entre outros [Adobe01]. Flash Lite especificamente é o perfil do Flash Player projetado para dispositivos móveis. O Flash Lite é usado para diversos fins nos dispositivos móveis, de conteúdo para aplicações até a própria interface do dispositivo.



Figura 13 - Iriver - dispositivo com interface desenvolvida em Flash Lite.



Figura 14 – Exemplos de dispositivos que rodam Flash Lite.

3.2.1 O que é Flash Lite

Macromedia Flash Lite é uma versão leve do Flash Player para dispositivos móveis. Funciona em uma variedade de sistemas operacionais embutidos, suporta a integração com o sistema operacional do dispositivo, provê suporte de rede para dados dinâmicos em aplicações, e utiliza codecs de hardware quando disponível. [Lynch05].

O Flash Lite está otimizado para trabalhar com possíveis restrições de recursos dos dispositivos, além de permitir ao desenvolvedor da aplicação um acesso às capacidades específicas do dispositivo e suportar o mesmo formato de arquivo SWF com Flash Player 7 para desktops. [Chanda06]

3.2.2 Características

3.2.2.1 Suporta o Flash Player 7

O Flash Lite 2.x está baseado no padrão para conteúdo do Flash 7 . Isto significa que aquele conteúdo desenvolvido no mais recente ambiente de desenvolvimento Flash pode ser colocado num dispositivo móvel ou em dispositivos eletrônicos. [Adobe07]

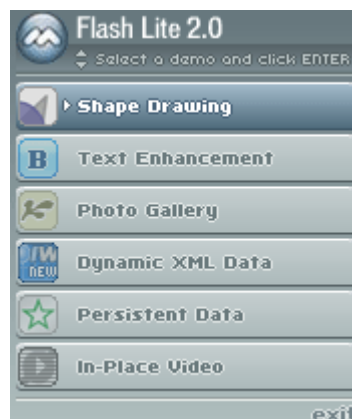


Figura 15 – Um exemplo de aplicação com Flash Lite 2.0.

3.2.2.2 Suporta Multi-plataforma

Flash Lite 2.1 provê suporte estendido por plataformas abertas fundamentais, inclusive Symbian S60 v2/v3, Qualcomm BREW 2.x/3.x e Microsoft Windows Mobile 5. [Adobe07]

3.2.2.3 Dados de XML dinâmicos

Flash Lite 2.x suporta carregamento e analisa gramaticalmente dados de XML externo usando o mesmo métodos que controlam XML no Flash Player 7. [Adobe07]



Figura 16 – Exemplo de aplicação que carrega dados externos XML.

3.2.2.4 Dados Persistentes

Flash Lite 2.x suporta a habilidade para armazenar localmente e recuperar informação relevante de aplicação específica como preferências, pontuações de jogos, logins, etc. Provendo um ambiente de desenvolvimento muito mais robusto. [Adobe07]



Figura 17 – Exemplos de uma aplicação que utiliza dados persistentes.

3.2.2.5 Mídias dinâmicas

Flash Lite 2.x disponibiliza carregamento dinâmico de conteúdo de multimedia como imagens, som e vídeo, baseado em codecs suportados e disponíveis no dispositivo. Isto inclui carregamento e controle de dados XML e conteúdo de SWF. O Flash Lite 2.x também provê suporte a vídeo e suporte de multimedia externo. Isto inclui em vídeo in place como também carregamento de imagem (jpeg, gif, png com transparência) e auditivo. [Adobe07]

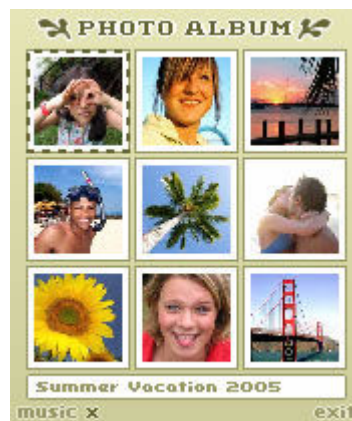


Figura 18 Aplicação com carregamento de mídias dinâmicas.

3.2.2.6 Edições de texto

Flash Lite 2.x permite aos usuários modificar cor de texto, tamanho, e outras propriedades aem tempo de execução, melhorando a exibição e controlando as fontes. Desenvolvedores também podem escolher embutir fontes de vetor nas suas aplicações que são usadas para fazer texto. Flash Lite 2.x também suporta renderização de texto em idiomas que se ler da direita pra esquerda como árabe e hebreu. [Adobe07]

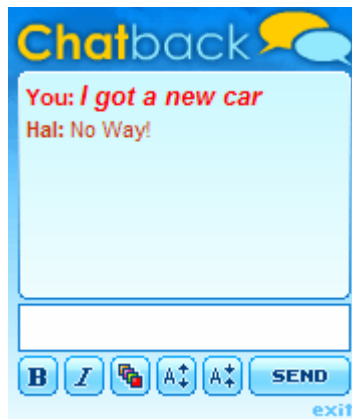


Figura 19 – Exemplo de aplicação com edição de texto.

3.2.2.7 Desenho de formas a partir da API de ActionScript

Flash Lite 2.x permite desenvolvedor criar gráficos de vetor sofisticados e formas animadas em tempo de execução usando ActionScript 2.0. [Adobe07]

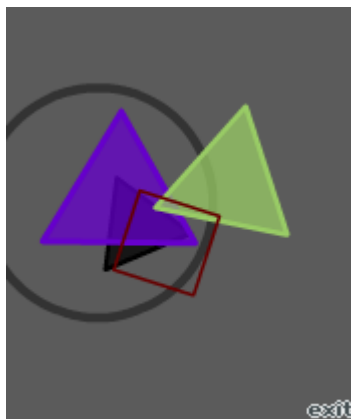


Figura 20 – Exemplos de aplicação que desenha formas dinamicamente.

3.2.2.8 Suporta ActionScript 2.0

Flash Lite 2.x suporta ActionScript 2.0, baseado no ECMA 262 padrão. Um conteúdo Flash Lite pode ser desenvolvido com um modelo de eventos mais moderno em relação a outra versão (clipe de filme e eventos de objeto), controle de índice de tabelas, desenho de forma através API, e compressão de SWF melhor em relação a outra versão. [Adobe07]

3.2.2.9 Som do dispositivo sincronizado

Flash Lite 2.x permite ao desenvolvedor de conteúdo sincronizar a animação com dados de som do dispositivo em formatos específicos como MIDI, SMAF, etc. [Adobe07]

3.2.2.10 Compressão de SWF's

Flash Lite 2.x suporta a renderização de arquivos SWF que estão comprimidos por desenvolvedores de conteúdo que usam a ferramenta de desenvolvimento do Flash. Flash Lite 2.x descomprimirá o arquivo SWF antes de começar o processo de renderização dos dados. [Adobe07]

3.2.3 Arquitetura

A figura 21 demonstra a arquitetura genérica do Flash Lite dentro dos dispositivos móveis.

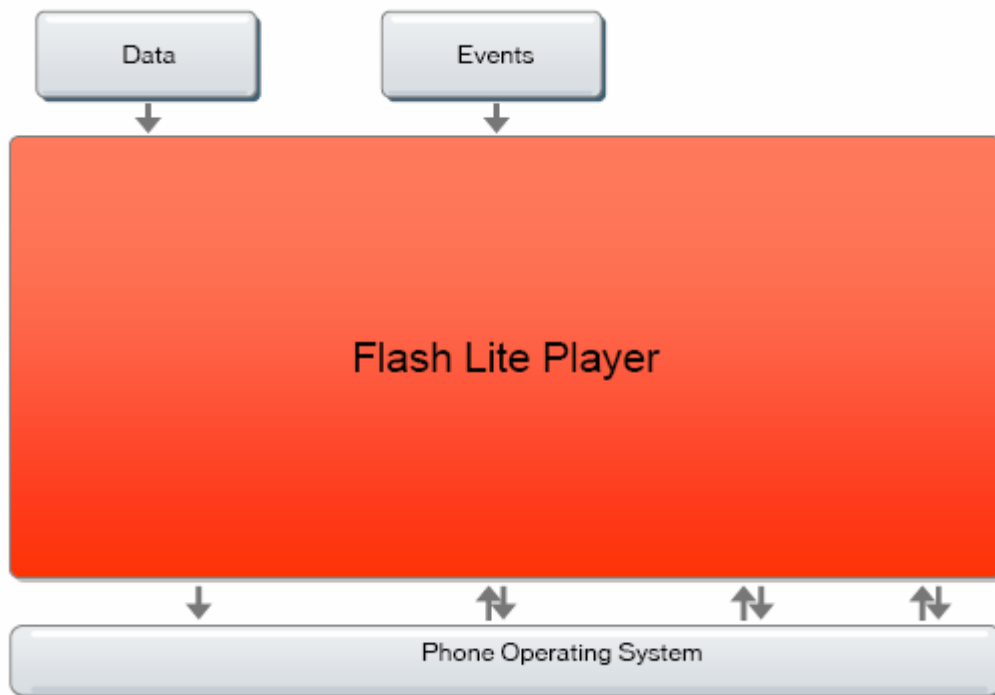


Figura 21 – Flash Lite Arquitetura genérica.

O código base de Flash Lite tem 280K e um adicional 117K para apoio de SVG-T. Estes tamanhos são gerados pelo compilador ARM – o tamanho do código é dependente de CPU, compilador, e sistema operacional. As configurações mínimas indicadas para um dispositivo executar o Flash Lite são 1 MB ROM, 50 MIPS, 2 MB RAM, e um bit de dados de 32-bits. [Lynch05]

O Flash Lite Player executa em cima sistema operacional do dispositivo, interagindo com o mesmo. Abaixo, a mesma figura da arquitetura, porém com um maior grau de detalhamento.

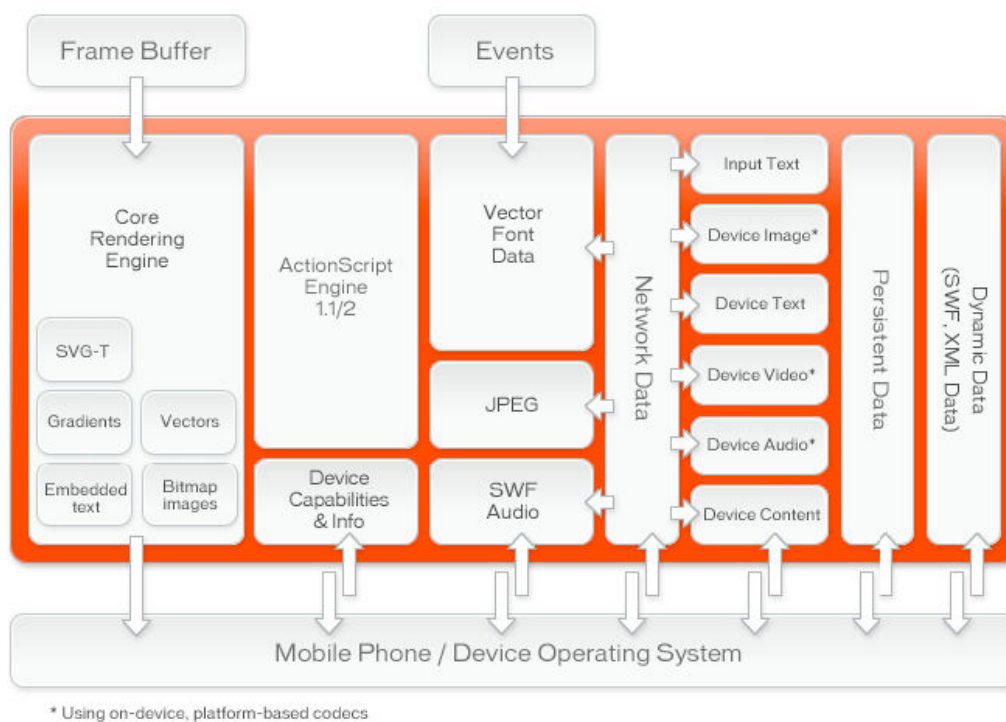


Figura 22 - Flash Lite 2.0Arquitetura detalhada.

3.2.2.1 Engine de renderização

As funcionalidades de mídia estão na engine de renderização do Flash Player, que suporta gráficos de vetor, bitmaps, gradientes e animação baseada em frames. Podem ser usadas fontes do próprio dispositivo, ou podem ser incorporadas novas fontes para exibição, e são suportados entradas de texto e texto dinâmico.

Existe também uma flexibilidade quanto à qualidade de renderização: Baixo, alto, médio (mais satisfatório para dispositivos móveis).

SVG-T está disponível em Flash Lite como uma opção, suportando vetor gráfico de exibição baseado em XML.. Capacidades auditivas incluem acesso aos sons do dispositivo e streaming de áudio. [Lynch05]

3.2.2.2 Engine ActionScript

O código é desenvolvido com ActionScript que em Flash Lite 2 está ao nível de Flash 7.

O Flash Lite suporta a integração do script com as capacidades do dispositivo, inclusive navegação através de teclas, pressionamento dos botões, notificação e integração com funcionalidades do sistema operacional do dispositivo em geral. [Lynch05]

3.2.2.3 Networking

O Flash Lite Player permite que o desenvolvedor faça requisições à rede HTTP/s para qualquer servidor de HTTP e stream ao player. A aplicação pode carregar diretamente outro SWF, como também pode carregar dados como dados de texto, dados de XML etc, além de carregar vários tipos de imagens e dados de som. [Chanda06]

O Action Script pode registrar eventos de comunicação de rede, tais como `onConnect` , `onLoad`.

A comunicação também pode ser estabelecidas via Sockets, o Player permite que um socket baseado em XML envie e carregue dados de um XML. As mensagens XML são enviadas numa full-duplex TCP/IP de conexão socket via stream. Um número ilimitado de mensagens de XML pode ser enviado e pode ser recebido em cima de uma única conexão de XMLSocket. [Chanda06]

3.2.2.4 Outros aspectos

O Flash Lite Player provê toda a parte multimídia dinâmica (imagens, áudio, vídeo), porém depende dos codecs específicos de cada dispositivo. [Adobe01].

3.3 Resumo das Características

Depois de descritos alguns aspectos de cada uma das linguagens utilizadas nesse estudo, serão apontadas, na tabela abaixo, algumas características de maneira mais breve e resumida de cada uma das linguagens.

	Características	J2ME	Flash Lite
Desenvolvimento	Linguagem	Java	ActionScript 2.0
	Curva de aprendizado	Média	Média
	Bibliotecas Disponíveis	Não aferido	Não aferido
	Debugger disponível	Sim	Sim
	Emulador disponível	Sim	Sim
	Ambiente integrado ao desenvolvimento disponível	Eclipse, NetBeans Mobility Pack	Macromedia Flash MX2004/8 / Eclipse
	Portabilidade pra outras plataformas	Média (muitas implementações da máquina virtual, necessitando de builds para cada dispositivo por existirem bugs diferentes para cada um)	Excelente (Bundled - Top 5 mobile manufacturers, limited handset model support as of 3/06, best web compatibility)
	Forma de empacotamento	Jad/Jar	SIS / CAB ou OTA/IR/Bluetooth, arquivos de SWF
	Custo da ferramenta de desenvolvimento	Grátis	Varia, porém é pago.
Potencialidades	Interface Gráfica	gráficos 2D, 3D,	gráficos 2D, Many widgets
	Comunidades e suporte para o desenvolvedor	Extensa	Extensa
	Dispositivos que suportam (quantidade)	Não aferido	Não aferido

Tabela 1 – Tabela de resumo de características de J2ME e Flash Lite.

4. Comparação entre as plataformas

Este capítulo visa detalhar as variáveis da comparação entre as tecnologias e seus agrupamentos, assim como explicar os critérios de avaliação utilizados.

Neste trabalho, foi montado um *framework* de comparação com o objetivo de levantar qualitativamente quais seriam os objetos a serem comparados entre as plataformas. Os objetos são as variáveis de comparação.

Foi definido um intervalo para cada variável a ser aferida, esse intervalo, varia entre um a cinco, sendo um a nota mínima e cinco a nota máxima. Para cada variável, esses valores representam qualitativamente um adjetivo que será demonstrado ainda neste capítulo.

4.1 Variáveis

As variáveis utilizadas na comparação das tecnologias estão agrupadas em quatro subgrupos, são eles:

- **Tempo**
- **Complexidade**
- **Custo de desenvolvimento**
- **Percepção do usuário**

Abaixo segue a tabela com todas as variáveis e seus agrupamentos:

	Variáveis
Tempo	tempo de aprendizado
	tempo de configuração da IDE
	tempo de desenvolvimento
	tempo de execução do aplicativo
Complexidade	complexidade na aprendizagem
	complexidade no desenvolvimento
	complexidade na portabilidade
Custo de desenvolvimento	produtividade
	ferramenta de testes
	recursos gráficos
Percepção do usuário	interface amigável
	qualidade do produto final

Tabela 2 – Tabela com variáveis do framework de comparação.

Cada um desses subgrupos serão detalhados no decorrer deste tópico.

4.1.1 Tempo

Essa variável está associada à impressão que o desenvolvedor teve em relação ao tempo decorrido para ele executar alguma tarefa ou mesmo o próprio aplicativo.

Dentro desse escopo o que se avalia dentro do grupo de comparação de tempo é:

- **Tempo de aprendizado:** corresponde a quanto tempo ele gastou para construir um “Hello word” na plataforma, assim

como o tempo necessário para alcançar as características móveis avançadas.

- **Tempo de configuração da IDE:** significa quanto tempo foi necessário para, a partir da ferramenta (IDE) aberta, começar a programar utilizando todas as ferramentas da plataforma, assim como o tempo gasto para integrar o emulador.
- **Tempo de desenvolvimento:** corresponde ao tempo gasto para desenvolver o aplicativo inteiro – neste caso o jogo *Bus Surfing* que está com o game design no primeiro anexo deste documento – desde quando a ferramenta está configurada até chegar à fase de testes.
- **Tempo de execução do aplicativo:** esta variável serve para analisar a diferença entre a “rapidez” que o aplicativo executa no emulador e “rapidez” que ele executa no próprio dispositivo.

4.1.2 Complexidade

Essa variável está associada à impressão que o desenvolvedor teve em relação a quão complexo foi a execução de determinada tarefa.

Dentro desse escopo o que se avalia dentro do grupo de comparação de complexidade é:

- **Complexidade no aprendizado:** esta variável corresponde a quão complexo foi o processo de aprendizado para o desenvolvimento de um “Hello word” na plataforma, assim como a complexidade encontrada para alcançar as características móveis avançadas.

- **Complexidade no desenvolvimento:** significa quão fácil ou difícil, esclarecendo que quanto mais difícil maior o grau de complexidade e quanto mais fácil menor o grau de complexidade – foi desenvolver o aplicativo inteiro, utilizando todos os recursos necessários do dispositivo.
- **Complexidade no *porting* do aplicativo:** corresponde ao grau de dificuldade encontrado para portar o aplicativo de um dispositivo para outro. Essa variável tem um grau de importância alto para esse estudo, porém não foi aferida.

4.1.3 Esforço de desenvolvimento

Essa variável está significando quão custosa foi realizar uma determinada tarefa durante o desenvolvimento do aplicativo. Nesta variável quando se fala em custo não se fala em valor monetário, o significado aqui é quanto trabalho gerou pra que determinada tarefa fosse alcançada.

Dentro desse escopo o que se avalia dentro do grupo de comparação de custo de desenvolvimento é:

- **Produtividade:** a quantidade de trabalho despendido antes do desenvolvimento para que o mesmo alcançasse uma boa produtividade.
- **Ferramenta de testes:** variável associada à quão custoso foi organizar e utilizar o ambiente para execução dos testes do aplicativo.

- **Recursos gráficos:** corresponde à quantidade de trabalho realizado para utilizar todos os recursos gráficos necessários para o desenvolvimento do aplicativo.

4.1.4 Percepção do usuário

Essa variável está relacionada à qual impressão que o desenvolvedor, como usuário da aplicação, teve ao ver o aplicativo finalizado e pronto para ser apresentado para o usuário final.

Dentro desse escopo o que se avalia dentro do grupo de comparação de Percepção do usuário é:

- **Interface amigável:** variável relacionada a quanto à interface traz para o usuário bem-estar, facilidade de utilização, suavidade nos gráficos.
- **Qualidade do produto final:** associado à avaliação final do produto como um todo, juntando todas as percepções do usuário em relação ao aplicativo.

4.2 Critérios de Avaliação

Conforme foi colocado no começo deste capítulo, foi definido um intervalo para cada variável, esse intervalo, varia entre um à cinco. Para cada variável, esses valores representam qualitativamente um adjetivo. Os adjetivos utilizados para caracterizar cada nota no intervalo escolhido para os grupos das variáveis “tempo”, “complexidade” e “esforço de desenvolvimento” foram:

- 1 - muito baixo
- 2 - baixo
- 3 - médio
- 4 - alto
- 5 - muito alto

Já pras o grupo de variável “percepção do usuário” os adjetivos utilizados para caracterizar cada nota no intervalo escolhido foram:

- 1 - péssima (o)
- 2 - ruim
- 3 - média (o)
- 4 - boa (bom)
- 5 - ótima (o)

5. Estudo Experimental

Este capítulo visa mostrar e analisar os resultados obtidos no estudo experimental. Estudo esse que foi feito baseado no desenvolvimento de um mesmo aplicativo, um jogo, nas duas plataformas estudadas.

Como explicado na metodologia, foram formados dois grupos de desenvolvedores, os quais já tinham experiência com as tecnologias base das plataformas, Java ou ActionScript.

5.1 Resultados

Durante o estudo experimental, dois desenvolvedores participaram do desenvolvimento em Flash Lite e cinco em J2ME. A média de tempo de

experiência (seja acadêmica ou profissional) com a linguagem base de desenvolvimento da equipe de Flash Lite é 6 anos e meio, no caso a linguagem base é ActionScript e de J2ME é 5 anos, sendo a linguagem base Java.

Já o tempo médio de contato dos desenvolvedores com a plataforma utilizada para o desenvolvimento do aplicativo, em Flash Lite foi 1 ano e em J2ME foi 6 meses.

Segue abaixo a tabela com as variáveis aferidas e os resultados médios obtidos através do questionário.

	Variáveis	Média Flash Lite	Média J2ME
Tempo	tempo de aprendizado	1,5	3
	tempo de configuração da IDE	1	3,2
	tempo de desenvolvimento	2	3,2
	tempo de execução do aplicativo	3	3,2
Complexidade	na aprendizagem	1,5	3,2
	no desenvolvimento	1,5	3,2
	na portabilidade	-----	-----
Esforço de desenvolvimento	produtividade	2,5	3,2
	ferramenta de testes	2	2,8
	recursos gráficos	2	3
Percepção do usuário	interface amigável	5	3,4
	qualidade do produto final	5	3,6

Tabela 3 - Tabela de resultados.

5.2 Análise dos resultados

Diante dos resultados coletados na tabela acima, foram levantadas algumas hipóteses que serão discutidas abaixo, separadas pelos subgrupos de variáveis.

5.2.1 Tempo

Em relação a este subgrupo de variáveis, as hipóteses levantadas foram:

- **Tempo de aprendizado:** Diante de um indivíduo que já tenha conhecimento na linguagem base, migrar o aprendizado (que teve o significado descrito no capítulo quatro) para a plataforma de desenvolvimento móvel é uma tarefa que pode ser desempenhada em menos tempo utilizando a plataforma Flash Lite, já que a mesma teve uma nota mais baixa em relação a J2ME. A primeira teve uma nota média igual a um e meio (1.5), que na escala utilizada representa um conceito entre baixo e muito baixo, enquanto a outra teve uma nota média igual a três (3) que significa um tempo médio na mesma escala.
- **Tempo de configuração da IDE:** Em relação ao tempo que o desenvolvedor leva para configuração todo o ambiente de desenvolvimento foi aferido que o Flash Lite teve um desempenho melhor. Levou-se menos tempo pra esta tarefa ser realizada, já que a mesma teve uma nota mais baixa em relação a J2ME. A primeira teve uma nota média igual a um (1), que na escala utilizada representa um tempo muito baixo enquanto a outra teve uma nota média igual a três e dois décimos (3.2) que significa um tempo entre médio e alto, porém mais próximo de médio.
- **Tempo de desenvolvimento:** O tempo de desenvolvimento foi menor quando o aplicativo foi desenvolvido em Flash Lite. Para executar esta tarefa os desenvolvedores levaram menos tempo em Flash Lite, já que a mesma teve uma nota mais baixa em relação a J2ME. A primeira teve uma nota média igual a dois (2), que na escala utilizada representa um tempo baixo enquanto a outra teve uma nota média igual a três e dois

décimos (3.2) que significa um tempo entre médio e alto, porém mais próximo de médio.

- **Tempo de execução do aplicativo:** Em relação a esta variável, o comportamento aferido das duas tecnologias foi quase o mesmo. O Flash Lite recebeu uma pontuação igual a três (3), que representa um tempo médio na escala de conceitos e o J2ME recebeu uma pontuação igual a três e dois décimos (3.2) que significa um tempo entre médio e alto, porém mais próximo de médio.

5.2.2 Complexidade

Em relação a este subgrupo de variáveis, as hipóteses levantadas foram:

- **Complexidade no aprendizado:** Diante de um indivíduo que já tenha conhecimento na linguagem base, a complexidade que existe para migrar o conhecimento para a nova plataforma é mais baixa utilizando a plataforma Flash Lite, já que a mesma teve uma nota inferior ao J2ME. A primeira teve uma nota média igual a um e meio (1.5), que na escala utilizada representa um conceito entre muito baixo e baixo, enquanto a outra teve uma nota média igual a três e dois décimos (3.2) que significa uma complexidade entre média e alta, porém mais próxima de média.
- **Complexidade no desenvolvimento:** Em relação à complexidade para desenvolver o aplicativo, o Flash Lite teve um desempenho melhor. O desenvolvimento foi menos árduo para esta plataforma, já que a mesma teve uma nota mais baixa em relação a J2ME. A primeira teve uma nota média igual um e meio (1.5), que na escala utilizada representa uma

complexidade entre baixa e muito baixa enquanto a outra teve uma nota média igual a três e dois décimos (3.2) que significa uma complexidade entre média e alta, porém mais próxima de média dentro da escala utilizada.

- **Complexidade no *porting* do aplicativo:** Esta variável não foi aferida nesta pesquisa.

5.2.3 *Esforço de desenvolvimento*

Em relação a este subgrupo de variáveis, as hipóteses levantadas foram:

- **Produtividade:** Levando-se em consideração a quantidade de trabalho despendido antes do desenvolvimento para alcançar boa produtividade, as duas plataformas se comportaram de maneira parecida, o Flash Lite apenas levou um pouco mais de vantagem já que a mesma teve uma nota mais um pouco inferior ao J2ME. A primeira teve uma nota média igual a dois e meio (2.5), que na escala utilizada representa um esforço entre baixo e médio enquanto a outra teve uma nota média igual a três e dois décimos (3.2) que significa um esforço entre médio e alto, porém mais próxima de médio dentro da escala utilizada.
- **Ferramenta de testes:** O esforço utilizado para organizar e utilizar o ambiente para execução dos testes do aplicativo foi um pouco menor em Flash Lite, as duas plataformas tiveram uma pontuação próxima, porém uma ficou mais perto de um conceito e outra de outro. O Flash Lite teve uma pontuação média igual a dois (2), que na escala utilizada representa um esforço baixo enquanto J2ME teve uma nota média igual a dois e oito décimos (2.8) que significa um esforço entre baixo e

médio, porém mais próximo de médio dentro da escala utilizada.

- **Recursos gráficos:** O esforço gasto para utilizar todos os recursos gráficos necessários para o desenvolvimento do aplicativo foi menor em Flash Lite. O esforço foi menos para esta plataforma, já que a mesma teve uma nota mais baixa em relação a J2ME. O Flash Lite teve uma pontuação média igual a dois (2), que na escala utilizada representa um esforço baixo enquanto J2ME teve uma nota média igual a três (3) que significa um esforço entre médio.

5.2.4 Percepção do usuário

Em relação a este subgrupo de variáveis, as hipóteses levantadas foram:

- **Interface amigável:** Levando-se em consideração o quanto à interface traz para o usuário bem-estar, facilidade de utilização e suavidade nos gráficos o Flash Lite se saiu bem melhor que o J2ME, já que o Flash Lite teve uma nota bem maior. O Flash Lite teve uma nota média igual a cinco (5), a nota máxima, que na escala utilizada representa uma interface ótima enquanto a outra teve uma nota média igual a três e quatro décimos (3.4) que significa uma interface entre média e boa, porém mais próxima de média dentro da escala utilizada.
- **Qualidade do produto final:** Esta variável está associada à avaliação final do produto como um todo, juntando todas as percepções do usuário em relação ao aplicativo, como explicado no capítulo quatro. Para esta variável o Flash Lite também teve um nota superior a J2ME. A primeira teve uma nota média igual a cinco (5), nota máxima, que na escala utilizada

representa uma interface ótima enquanto a outra teve uma nota média igual a três e seis décimos (3.6) que significa uma qualidade entre média e boa, porém mais próxima de boa dentro da escala utilizada.

6. Considerações Finais

Este capítulo discute as conclusões obtidas a partir deste trabalho e sugere possíveis modificações futuras para tornar este trabalho melhor e mais completo.

6.1 Conclusões

No desenvolvimento de aplicativos, tais como jogo, para dispositivos móveis é muito importante que se tenha uma relação de custo eficiência boa, pois para os fabricantes e vendedores de aplicativos isso é um grande diferencial.

Tendo em vista esse cenário, a relação de custo eficiência foi aferida neste estudo através da medição qualitativa de variáveis como tempo, complexidade, esforço no desenvolvimento e percepção do usuário.

Dessa forma, através do framework de comparação foi levantada as seguintes hipóteses:

- Em relação às variáveis de **tempo**, o Flash Lite tem um desempenho superior ou igual ao J2ME. Sendo assim, o Flash Lite é uma plataforma de que traz agilidade e rapidez ao

processo de desenvolvimento de aplicativos, especificamente, jogos. Enquanto que o J2ME tem um desempenho mediano.

- Em relação às variáveis de **complexidade**, o Flash Lite tem um desempenho melhor que J2ME. Desta forma, o Flash Lite é uma plataforma que provê facilidade no entendimento da ferramenta e desenvolvimento do aplicativo. Já o J2ME, é uma plataforma que não provê tanta facilidade, é uma plataforma que tem uma complexidade mediana.
- Em relação às variáveis de **esforço de desenvolvimento**, o Flash Lite teve um desempenho um pouco superior ao J2ME. Sendo assim, as duas plataformas despendem quase a mesma quantidade de esforço na interação do desenvolvedor com o ambiente, o J2ME despende ligeiramente um pouco mais de esforço.
- Em relação às variáveis de **percepção do usuário**, o Flash Lite teve um desempenho muito superior ao J2ME. Isto significa que visualmente é mais interessante na visão do usuário. O J2ME é uma plataforma que tem um desempenho gráfico mediano.

Logo, a hipótese levantada neste estudo é que Flash Lite tem uma melhor relação custo-eficiência se comparado com o J2ME, enfatizando no desenvolvimento de jogos. Esta hipótese precisa ser validada em um estudo quantitativo, envolvendo mais equipes e em projetos de maior porte.

6.2 Trabalhos Futuros

A metodologia utilizada neste trabalho pode ser melhorada em alguns aspectos tais como:

- Fazer um estudo **quantitativo** com o objetivo de colher dados mais representativos para validar a hipótese levantada neste trabalho.
- Desenvolver **outros aplicativos**, além de jogo, para comparar outros recursos do dispositivo que não são usados no desenvolvimento de um jogo.
- Escrever uma **pesquisa mais ampla** pra que possa ser aplicada a mais desenvolvedores e não só aos que participaram do desenvolvimento do mesmo aplicativo.
- **Definir mais variáveis** para melhorar a comparação, algumas muito importantes como portabilidade, por exemplo, não foram abordadas nesse trabalho.

Referências

[**LinhaCodigo04**] Netto, Mosimann. Linha de Código -. Dispositivos Móveis. Ano: 2004. Disponível em: http://www.linhadecodigo.com.br/artigos.asp?id_ac=206. Acessado em: Março/2007.

[**Abragames05**] A Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil. Pesquisa publicada pela Abragames, Associação Brasileira das Desenvolvedoras de Jogos Eletrônicos. Ano: 2005. Disponível em: <http://www.abragames.org/docs/PesquisaAbragames.pdf> . Acesso em: Novembro/ 2006.

[**Wells**] Wells, Martin J., J2ME Game Programming. Premier Press. Series Editor.

[**Topley02**] Topley , Kim. J2ME in a Nutshell. Editora O'Reilly. Ano: 2002.

[**Muchow01**] Muchow , John W. Core J2ME™ Technology & MIDP. Editora Prentice Hall PTR. Ano:2001.

[**Lam**] Lam, Jason. J2ME & Gaming

[**Barbagallo04**] Barbagallo, Ralph. Wireless game development in Java with MIDP 2.0. Editora Wordware Publishing. Ano:2004.

[**Talbot06**] Talbot, James. Creating Flash Lite Applications. Adobe Systems Incorporated. Ano: 2006.

[Uniformg04] Estrutura de um Projeto de Pesquisa. Uniformg. Ano: 2004.
Disponível em:
http://www.uniformg.edu.br/imagens/depcom/outros/Estrutura_Projeto_Pesquisa_uniformg.pdf Acesso em: Março/ 2007.

[Lynch05] Lynch, Kevin. The Flash Platform - Delivering Effective User Experiences Across Browsers, Operating Systems, and Devices. Macromedia, Inc. Ano: 2005.

[Chanda06] Chanda, Rupen. Creating Flash Applications on BREW Handsets. Brew Conference 2006. Adobe Systems Inc. Ano: 2006.

[Janousek06] Janousek, Scott. Flash Lite Overview. Vermont Flash User Group. Ano: 2006.

[Adobe06] Adobe Flash Lite 2.1- Delivering exceptional mobile experiences. DATASHEET. Adobe Systems Inc. Ano: 2006.

[Adobe07] Flash Lite 2.x Features. Adobe Systems Inc. Ano: 2007. Disponível em: <http://www.adobe.com/products/flashlite/productinfo/features/>.
Acessado em: Março/2007.

[Pelkonen04] Pelkonen, Tommi. Mobile Games. Publicado na série E-Content Reports da ACTeN. Ano: 2004.

[Semenov04] Semenov , Alexei. MOBILE GAMES. Ano: 2004.

[Mendonça06] Mendonça, Aderval Mobilidade em análise. Ano: 2006.
Disponível em: http://www.amsyst.com.br/artigos_20061106.shtml.
Acessado em: Março/2007

[Wiki06] Wikipedia - Enciclopédia Online. História dos Jogos de Vídeo Game. Disponível em:

http://en.wikipedia.org/wiki/History_of_video_game_consoles. Acessado em: Março/2007.

[Junior06] Junior, Antonio. Segurança em telefonia Móvel. Imasters. Ano: 2006. Disponível em: http://www.imasters.com.br/artigo/5194/mobile/seguranca_em_telefonia_movel//imprimir/ Acessado em: Março/2007.

[MCT05] Tecnologias de Visualização na Indústria de Jogos Digitais - Potencial econômico e tecnológico para a indústria brasileira de software. Pesquisa realizada pelo Softex e MCT. Ano: 2005

[Esa06] 2006 Essential Facts about Computer and Video Game Industry. Pesquisa publicada pela Entertainment Software Association (ESA). Ano: 2006.

[Hull06] Hull, Drew. Power Players in Mobile Gaming. Ano: 2006. Disponível em: http://wireless.npd.com/bulletinpowerplays_mobile.html Acessado em: Março/2007.

[NPD06] Mobile Gaming, Licensing and Original IP. NPD's Wireless Research Team. Disponível em: http://wireless.npd.com/bulletin_mobilegaming_032007.html Acessado em: Março/2007.

[PWC06] Global Entertainment and Media Outlook: 2006–2010. Pesquisa publicada pela PricewaterhouseCoopers(PWC). Ano:2006

[Sun071] Personal Profile Overview. Sun Microsystems, Inc. Ano: 2007. Disponível em: <http://java.sun.com/products/personalprofile/overview.html>. Acessado em: março/2007

[Sun072] Pawlan, Monica. Introduction to Consumer & Embedded Technologies. Sun Microsystems, Inc. Ano: 2007. Disponível em: <http://java.sun.com/developer/technicalArticles/ConsumerProducts/intro/index.html>. Acessado em: Março/2007

[Goggin05] Goggin, Richard., Mittal , Rajat. Mobile Device Platforms. Microsogt. Ano :2005.

[Malykhina07] Malykhina, Elena. How Smartphone Platforms Compare. InformationWeek. Ano: 2007. Disponível em: <http://www.informationweek.com/news/showArticle.jhtml?articleID=196902226>. Acessado em: Marco/2007

[McKenna00] McKenna, Daniel. Mobile Platform Benchmarks. Transmeta Corporation. Ano: 2000. Disponível em: http://www.charmed.com/PDF/BenchmarkWhitePaper_2-3-00.pdf. Acessado em: Março/2007

Apêndice A – Documento de Game Design



Bus Surfing

Game design
Versão 1.0

1. Histórico do Projeto

Esta seção tem com objetivo apresentar a evolução deste documento. É apresentado aqui o conteúdo da versão inicial do documento e, para cada versão seguinte, o que de novo foi acrescentado ou o que foi removido.

Título do documento		do		Plano de Projeto
Equipe		Bengt Karlsson <mauhur@gmail.com> Carlos Cunha <carloscunha4@gmail.com> Fernando Brayner <fbrayner@gmail.com> Igor Colares <igor.colares@gmail.com> Lais Xavier <lausinha@gmail.com> Luis Otávio <zarathus@gmail.com> Marcellus Tavares <marcellustavares@gmail.com>		
Comentários				
Nome do arquivo		bus_surfing_gamedesign.doc		
HISTÓRICO DE REVISÕES				
Revisão	Data	Descrição		Autor
01	02/12/06	Esboço inicial do documento		Marcellus
02	04/12/06	Elaboração do documento		Laís

2. Visão Geral do Jogo

2.1 Filosofia

→ 2.1.1 Aonde queremos chegar?

O objetivo do Bus Surfing é atender as necessidades de entretenimento de usuários de celulares que jogam casualmente.

→ 2.1.2 Tipo do jogo

Bus Surfing é um jogo bidimensional, onde o jogador controla um personagem que tem o objetivo de chegar a um lugar predeterminado da melhor forma possível.

→ 2.1.3 Plataformas de execução

O jogo será desenvolvido para ser executado inicialmente em celulares que suportem a tecnologia J2ME e celulares que suportem Flash Lite.

2.2 Questões Comuns

→ 2.2.1 O que é o jogo?

Bus Surfing é baseado em jogos de surf tradicionais. O diferencial está em fatores como a jogabilidade, a necessidade de alcançar um objetivo além dos obstáculos e no fato dele evoluir (em nível maior ou menor) de acordo com os tipos de obstáculos que ele ultrapassar.

→ 2.2.2 Por que criar o jogo?

Uma série de motivos estimulou a criação deste jogo. Entre eles, os principais são:

- Fazer um estudo comparativo entre a jogabilidade, facilidade de implementação, gráficos entre duas tecnologias distintas para criação de jogos em dispositivos móveis.
- A oportunidade de desenvolver um jogo satisfazendo uma encomenda, simulando assim o funcionamento do mercado de jogos.
- Diversão e paixão por jogos.

➔ 2.2.3 Onde se ambienta o jogo?

Os cenários do jogo são ambientados nas ruas de uma cidade urbana inspiradas em Recife e Olinda.

➔ 2.2.4 O que se controla no jogo?

O jogador controla um personagem, podendo ter duas opções de movimento, pular e agachar.

➔ 2.2.5 Quantos personagens se controla no jogo?

Apenas um: o personagem que está disponível ao usuário no começo do jogo.

➔ 2.2.5 Qual o foco principal do jogo?

O foco principal do jogo consiste em passar por todos os estágios sem se acidentar fatalmente e acumulando o maior número de pontos.

➔ 2.2.6 O que o jogo apresenta de diferente?

É um jogo inovador com diferenças marcantes sobre os similares, são elas:

- Cenários regionais
- O Jogo é desenvolvido para ser executado em plataformas móveis.
- Recompensas fora da linha principal do jogo, permitindo que um novo jogo tenha um desfecho diferente do anterior.
- Uma temática próxima ao dia a dia de uma grande gama de usuários

3. Conjunto de Características

3.1 Características Gerais

- Jogo 2D com vista lateral;
- Possui apenas um personagem que é o próprio jogador;
- O cenário do jogo é baseado nas ruas da cidade do Recife e Olinda.
- O dispositivo de entrada para jogar é o teclado do celular;
- O jogo é destinado a apenas um jogador.

3.2 Gameplay

- O Bus Surfing possui apenas um modo de jogo, onde o objetivo é passar por todos os estágios sem se acidentar fatalmente. Em cada estágio o jogador precisa chegar a um determinado local surfando sobre os ônibus que passam pelas ruas.
- O jogador controla um personagem que pula e se abaixa.
- O jogador deve controlar o personagem de modo a livrá-lo dos objetos que possam vir a atingi-lo, como postes, fios, árvores, etc.
- Inicialmente o jogador tem que se deslocar a um destino que não oferece muitos riscos. Os desafios vão aumentando ao avançar dos níveis.
- O jogador coleta itens que aumentam a sua pontuação.

4. O Mundo do Jogo

4.1 Cenários

O jogo é dividido em quatro cenários diferentes. Cada cenário apresenta uma característica própria que interfere na jogabilidade. Cada cenário será descrito a seguir, na ordem de aparição durante o jogo:

1. **Cerveja com amigos:** Este cenário se passa a caminho da praia. Os itens que podem ser pegos tem como tema a praia. As ruas são mais amplas, bem cuidadas (sem muitos buracos) e poucas curvas ou curvas suaves.
2. **Um amor de matar:** Este cenário se passa nas partes menos urbanas da cidade com algumas árvores e as ruas com curvas um pouco mais acentuadas.
3. **Ó Olinda:** Este cenário se passa em uma cidade inspirada em Olinda. O movimento do jogador se torna mais difícil, por conta das ladeiras e buracos da cidade. Os itens seguem o motivo do cenário.
4. **Hora do Rush:** Este cenário se no centro urbano da cidade. Trânsito caótico, túneis, semáforos e fios integram o cenário tornando mais complexo o jogo. Os itens continuam seguindo o motivo do cenário.

4.2 Pontuação

A pontuação do jogo se baseia em manter-se em pé e desviando dos obstáculos, onde cada movimento bem sucedido é recompensado com um número de pontos. Além disso, existem itens que podem ser coletados tanto para recuperar a vida do personagem no caso de acidentes, quanto itens relacionados ao estágio, que oferecem um número de pontos ainda maior que os outros desafios.

Tarefas obrigatórias

- Desviar de árvores, fios, placas e pássaros.

Bônus de vida

- Item que aparecerá aleatoriamente no cenário. O jogador pode ganhar a bonificação através de vida.

Itens que dão pontos extras

- O jogador ao pegar o item de ponto extra recebe 50 pontos. Este item tem relação com o cenário do jogo.

4.3 Sistema de Renderização

O jogo é isométrico. Cada sprite é desenhado usando os métodos da classe Canvas da API do J2ME a partir de figuras em formato png. No flash lite as figuras serão desenhadas utilizando o Flash 8, onde as figuras são vetorizadas.

4.4 Game Engine

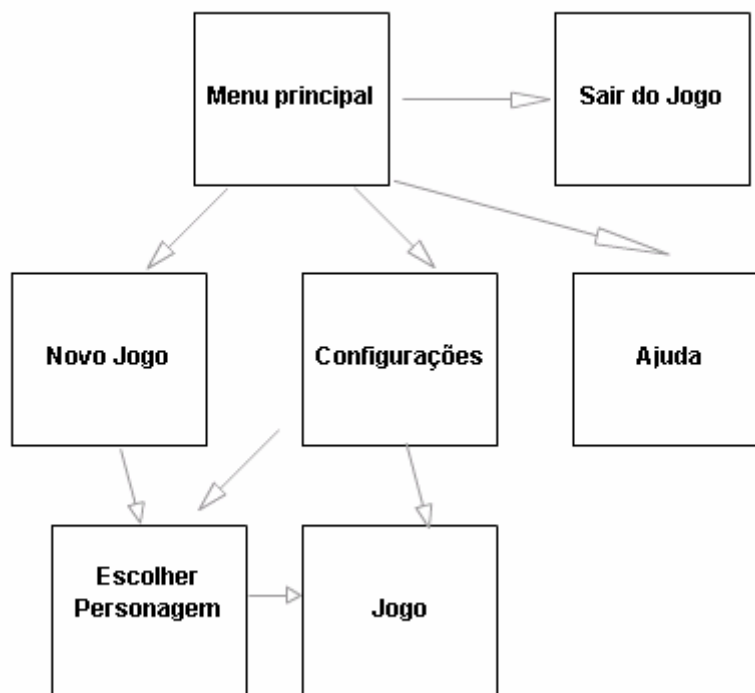
A engine do jogo tratará basicamente das seguintes tarefas do jogo:

- **Tratamento de colisão:** Parte da engine que verifica se ocorreu colisão, seja do personagem com os obstáculos, com os itens ou com o ônibus.
- **Movimentação:** Cuida da movimentação dos objetos na tela. Esses objetos são o personagem e o ônibus.
- **Física do jogo:** Responsável pela física do jogo, sem sempre representando a física real. Tratará da aceleração do ônibus e seu efeito sobre o personagem.
- **Tratamento do teclado:** Toda a navegação e controle do jogo serão tratados por esse sub-conjuto da engine.

5. Interface com o Usuário

5.1 Visão Geral

A interface do Bus Surfing é constituída de menus navegáveis que permite ao usuário configurar parâmetros, obter ajuda, iniciar um jogo e sair. Abaixo é mostrada a hierarquia de menus do jogo e a definição de cada tela:



Menu principal: É constituído por uma figura artística do Bus Surfing e abaixo uma lista de ações que o usuário pode executar. São elas:

- **Novo jogo:** Inicia um novo jogo, com apenas um cenário habilitado, com o placar e vidas do jogador nos valores iniciais.
- **Configurações:** Permite ao jogador configurar parâmetros do jogo e outros utilitários relacionados à jogabilidade.
- **Ajuda:** Introduz o jogador novato ao Bus Surfing.

Sair do Jogo: Encerra a execução do Bus Surfing.

Novo Jogo: Nesta tela o jogador é informado do início de um novo jogo.

Configurações: Nesta tela o jogador pode configurar os seguintes parâmetros:

- Ativar/Desativar luz permanente do celular durante o jogo.
- Ativar/Desativar vibração do celular durante o jogo.

Ajuda: Nesta tela são exibidas informações a respeito do jogo (controles, objetivos, etc) para o jogador leigo.

Escolher Personagem: Nesta tela são exibidas figuras representando cada personagem. A partir desta tela o jogador poderá escolher qual personagem irá jogar.

Jogo: Nesta tela é onde se passa o jogo em si.

5.2 Controles do jogo

No Bus Surfing, as teclas 2, e 8 são utilizadas para a navegação dos menus. A tecla 5 é utilizada para fazer a confirmação da seleção, assim como para controlar o personagem, clicar e soltar boneco pula, clicar e pressionar ele se abaixa.

6. Itens

6.1 Visão Geral

Durante o jogo, itens aparecerão passando pelo personagem e devem ser coletados para trazer algum benefício ao jogador. Os itens podem ser de recuperação de vida, que irão curar o personagem de danos sofridos em acidentes, ou de bônus, que aumentam a pontuação do jogador e melhoram o resultado final de cada fase.

Os Itens mudam de acordo com o estágio. Na fase da praia o personagem pode se deparar com itens relacionados a essa atividade como caixas de cerveja, isopor, etc.

6.2 Itens do cenário “Cerveja com amigos”

6.2.1 – Itens de Vida

- **Cerveja**
- **Galeto**

6.2.2 – Itens de Bônus

- **Isopor**
- **Bola de futebol**

6.3 Itens do cenário “Um amor de matar”

6.3.1 – Itens de Vida

- **Chocolate**
- **Vinho**

6.3.2 – Itens de Bônus

- **Presente**
- **Flores**
- **Cartão**

6.4 Itens do cenário “Ó Olinda”

6.4.1 – Itens de Vida

- **Tapioca**
- **Acarajé**

6.4.2 – Itens de Bônus

- **Lembrancinhas**
- **Sombrinha de frevo**

6.5 Itens do cenário “Hora do Rush”

6.5.1 - Itens de vida

- Café

6.5.2 - Itens de Bônus

- Relatório

- Laptop

7. Jogo Single Player

8.1 Visão Geral

O modo Single Player de Bus Surfing possui quatro cenários, cada cenário possui 1 nível. Inicialmente o jogador tem acesso apenas ao primeiro cenário (Cerveja com Amigos). Ao completar o cenário, é habilitado o cenário seguinte. A ordem dos cenários é Cerveja com Amigos, Um amor de matar, Ó Olinda, Hora do Rush.

8.2 Horas de Jogo

O jogo proporciona ao jogador cerca de 10 à 12 minutos de jogo.

8.3 Condições de Vitória

Para poder superar o nível atual, é necessário não morrer. Ao se completar todos os cenários com vida, termina com a vitória do jogador.

Quebrar recordes de pontuação é uma condição alternativa de vitória.

Apêndice B – Questionário

PESQUISA: INVESTIGAÇÃO DAS CARACTERÍSTICAS
TECNOLOGIAS PARA DESENVOLVIMENTO EM
DISPOSITIVOS MÓVEIS, J2ME E FLASH LITE

TRABALHO DE GRADUAÇÃO

Aluno: Laís Xavier (lx@cin.ufpe.br)

Orientador: Prof. Fabio Bueno Queda da Silva(fabio@cin.ufpe.br)

29 de março de 2007

- ∴ Este questionário é parte de uma investigação científica sobre as características de duas tecnologias para desenvolvimento em dispositivos móveis, J2ME e Flash Lite .
- ∴ Não há respostas certas ou erradas. Suas respostas devem se basear naquilo que você realmente conhece, sente ou faz.
- ∴ A participação nesta pesquisa é voluntária.
- ∴ Leia com atenção todas as questões.
- ∴ Caso tenha alguma dúvida, solicite ajuda ao moderador que estiver aplicando a pesquisa.
- ∴ Todas as suas respostas serão CONFIDENCIAIS, sendo o preenchimento deste questionário ANÔNIMO.
- ∴ Para que este estudo seja válido, é importante que você responda cada questão de maneira **cuidadosa e sincera**.
- ∴ Marque sua(s) resposta(s) com um X nas três primeiras questões, a partir delas será uma .

Identificação

1. Qual o plataforma você trabalhou?

- ☐ Flash Lite
- ☐ J2ME
- ☐ Nenhuma das opções

Se a sua resposta à primeira questão for “Nenhuma das opções”, o questionário está encerrado, obrigada por respondê-lo, se não foi, continue respondendo.

2. Qual o sua experiência com a linguagem base (Java ou ActionScript)? Se já teve alguma experiência, assinalar ao lado o tempo.

- ☐ Nunca teve
- ☐ Experiência acadêmica Tempo:
- ☐ Experiência profissional Tempo:
- ☐ Outro(s). Especifique:

3. Qual o sua experiência com a tecnologia trabalhada (J2ME ou Flash Lite)?

Se já teve alguma experiência, assinalar ao lado o tempo.

- ☐ Nunca teve
- ☐ Experiência acadêmica Tempo:
- ☐ Experiência profissional Tempo:
- ☐ Outro(s). Especifique:

Da questão quatro à questão treza as respostas serão dadas numa escala de 1 à 5, sendo 1 (um) a nota mínima e 5 (cinco) a máxima, abaixo segue as correspondências para cada nota:

- 1 - muito baixo**
- 2 - baixo**
- 3 - médio**
- 4 - alto**
- 5 - muito alto**

Nas questões quatorze e quinze as respostas serão dadas numa escala de 1 à 5, sendo 1 (um) a nota mínima e 5 (cinco) a máxima, abaixo segue as correspondências para cada nota:

- 1 - péssimo
- 2 - ruim
- 3 - médio
- 4 - bom
- 5 - ótimo

		Variáveis	Avaliação
Tempo	4	tempo de aprendizado	
	5	tempo de configuração da IDE	
	6	tempo de desenvolvimento	
	7	tempo de execução do aplicativo	
Complexidade	8	na aprendizagem	
	9	no desenvolvimento	
Custo de desenvolvimento	11	produtividade	
	12	ferramenta de testes	
	13	recursos gráficos	
Percepção do usuário	14	interface amigável	
	15	qualidade do produto final	

Observações:

Muito obrigada pela colaboração!