

# Plataformas para Desenvolvimento de Jogos para Celulares

WALLACE FRANCO NOGUEIRA<sup>1</sup>  
EMERSON CAVALCANTE LOUREIRO FILHO<sup>2</sup>  
HYGGO OLIVEIRA DE ALMEIDA<sup>3</sup>

<sup>1</sup>UFAL - Universidade Federal de Alagoas  
TCI - Departamento de Tecnologia da Informação  
Campus A.C. Simões - Km 97 - Tabuleiro dos Martins  
CEP 37200-000 Maceió (AL)  
wallace@labpesquisas.tci.ufal.br

UFCG - Universidade Federal de Campina Grande  
<sup>2</sup>Departamento de Sistemas de Computação, <sup>3</sup>Departamento de Engenharia Elétrica  
Aprigio Veloso - 882 - Bodocongó  
CEP 58109-970 Campina Grande (PB)  
eclf@dsc.ufcg.edu.br, hyggo@dee.ufcg.edu.br

**Resumo.** A diversidade de plataformas para o desenvolvimento de jogos para dispositivos móveis dificulta uma escolha adequada sobre qual delas utilizar em um projeto de jogo. Uma vez que a decisão da plataforma está diretamente relacionada aos recursos disponíveis ao desenvolvedor, torna-se necessário um estudo das características de cada plataforma para identificar qual delas se adequa melhor ao jogo sendo desenvolvido. Neste artigo é apresentada uma análise comparativa entre as principais plataformas para desenvolvimento de jogos para dispositivos móveis disponíveis atualmente. A análise serve como base para guiar o desenvolvedor na decisão da plataforma que melhor satisfaz as suas necessidades.

**Palavras-Chave:** Dispositivos móveis, jogos eletrônicos, plataformas de desenvolvimento.

## Development Platforms for Cell Phone Games

**Abstract.** The diversity of development platforms for mobile device games makes difficult to choose the right one when developing a game. Such choice is strongly related to the resources available for the developer. For that, it is necessary to study the features of each platform, in order to identify what platform is more suitable for a specific game. This paper presents a comparative analysis of the main development platforms for cell phone games. This analysis can be used for helping the developer to choose the right platform according to the game and device features.

**Keywords:** Mobile Devices, electronic games, development platforms.

(Recebido para publicação em 29 de outubro de 2004 e aprovado em 13 de abril de 2005)

### 1 Introdução

Ao final da década de 90, dispositivos móveis como *hand-helds* e telefones celulares já apresentavam uma evolução considerável, em relação aos seus antecesso-

res, em termos de poder de processamento, armazenamento e conectividade. A partir dessa evolução, tais dispositivos puderam suportar a execução de softwares cada vez mais funcionais e complexos, despertando assim a atenção da indústria de desenvolvimento de soft-

ware.

Dentro dessa nova classe de programas suportados por esses dispositivos, encontram-se os jogos. Partindo de aplicações simples, com poucas cores e som bastante limitado, os jogos para dispositivos móveis evoluíram ao ponto de hoje apresentarem uma grande variedade de recursos, tais como gráficos em 3D, sons polifônicos, etc.

Desenvolver jogos que sejam capazes de apresentar esses recursos envolve fatores dependentes das características de hardware disponíveis a cada dispositivo, tais como capacidade de processamento e armazenamento. No entanto, os dispositivos móveis atuais apresentam entre si características muito particulares, possuindo diferente quantidade de memória disponível para os aplicativos desenvolvidos, capacidade de processamento e sistemas operacionais.

Uma maneira de contornar esse problema é inserir uma camada de software, aqui chamada de plataforma, entre a aplicação (jogo) e o sistema operacional do dispositivo. Essa camada seria responsável por abstrair os detalhes inerentes a cada dispositivo, definindo um conjunto de funções de mais alto nível, o que permitiria a portabilidade de um mesmo jogo entre diferentes dispositivos.

No entanto, o grande número de plataformas existentes hoje no mercado, dificulta a escolha precisa de qual delas utilizar em um projeto de jogo. Esta escolha no entanto é crucial, pois dela depende a possibilidade de criação das características contidas em cada projeto de jogo, bem como a quantidade de usuários que poderão ter acesso àquele produto.

Neste artigo é apresentada uma análise das principais vantagens e desvantagens das principais plataformas de desenvolvimento de jogos para dispositivos móveis. Tem-se como objetivo prover uma base de comparação entre as plataformas, no intuito de auxiliar o desenvolvedor a realizar a sua escolha sobre qual dentre estas plataformas escolher na hora de desenvolver seu projeto.

O restante do artigo está organizado da seguinte maneira: na Seção 2 são apresentados os parâmetros a serem utilizados na análise comparativa entre as plataformas selecionadas. A descrição de cada uma delas e a análise final são apresentadas na Seção 3. Na Seção 4, apresenta-se um estudo de caso de desenvolvimento utilizando a análise comparativa. As conclusões são apresentadas na Seção 5.

## 2 Parâmetros de análise

Para que uma plataforma de desenvolvimento possa ser utilizada na criação de um jogo é necessário que ela ofe-

reça suporte aos vários elementos presentes neste tipo de aplicativo. Em outras palavras, a plataforma deve ser capaz de apresentar características de multimídia tais como animações e sons, oferecer suporte à conectividade com outros dispositivos, assim como gerenciar a interação entre os elementos contidos no jogo.

Durante a análise foram considerados não somente critérios relacionados aos aspectos econômicos, como a participação no mercado de cada plataforma e seus respectivos modelos de negócios, como já visto em [14], mas também aspectos técnicos de cada uma delas. Mais especificamente os parâmetros analisados são descritos a seguir.

- Linguagem de desenvolvimento: indica a linguagem utilizada como base para o desenvolvimento na plataforma e quais as conseqüências disto.
- Segurança: indica o grau de segurança da plataforma contra falhas nos aplicativos em execução, aplicativos mal-intencionados, etc.
- Custo para desenvolvimento: indica qual o custo que deve ser arcado pelo desenvolvedor para que seja possível iniciar o desenvolvimento de seu jogo.
- Modelo de negócios: neste ponto é analisado que meios são disponibilizados para o desenvolvedor publicar seu jogo e a que custo esta operação é realizada.
- Carga: mostra a quantidade de recursos que a plataforma utiliza do dispositivo, assim como a quantidade que é disponibilizada pela mesma para a execução de jogos.
- Suporte a elementos de jogo: quais os tipos de elementos de jogo suportados pelas plataformas, tais como a manipulação de gráficos 2D e 3D, reprodução de vídeos, sons, etc.

## 3 Análise comparativa

Nesta seção é apresentada uma análise comparativa das principais plataformas para o desenvolvimento de jogos para dispositivos móveis: *J2ME*, *BREW*, *ExEn* e *Mophun*. Na Figura 3.4 pode-se observar o resumo das principais características das plataformas analisadas de acordo com os parâmetros definidos na seção anterior.

### 3.1 J2ME

A plataforma *J2ME* (*Java 2 Micro Edition*) [18] é uma versão reduzida da plataforma Java. Lançada em 1999,

esta plataforma foi concebida com o propósito de oferecer suporte ao desenvolvimento de aplicativos para dispositivos com recursos reduzidos. Atualmente se encontra na versão 2.0 e conta com o maior número de jogos lançados no mercado [16]. Nessa versão foi criado um suporte muito maior ao desenvolvimento de jogos do que o oferecido pela versão inicial.

A plataforma *J2ME* consiste basicamente de dois módulos, *Configuração* e *Perfil*, organizados como ilustrado na Figura 1.

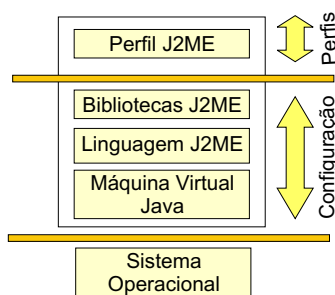


Figura 1: Arquitetura da plataforma *J2ME*

O módulo de configuração define a configuração mínima requerida por um grupo de dispositivos com características similares de hardware, por exemplo, memória e processador. O módulo de perfil é responsável por garantir a interoperabilidade entre uma família de dispositivos. Definindo-se um perfil para telefones celulares, por exemplo, garante-se o funcionamento de um aplicativo em todos os dispositivos desta família.

Além disto, outra vantagem da plataforma é que ela é completamente gratuita, ou seja, o desenvolvedor pode realizar download do *SDK*<sup>1</sup>, desenvolver seus produtos livremente e não precisa disponibilizá-los através de nenhum canal específico. Por outro lado, apesar de possibilitar independência para o desenvolvedor, esta abordagem faz com que não existam mecanismos de prevenção à pirataria, assim como força o desenvolvedor a ser responsável pela publicidade e distribuição do próprio jogo.

Porém, existem outras vantagens de *J2ME* em relação à disponibilização: a vasta quantidade de dispositivos que a suportam. Além disso, a compatibilidade entre os modelos de dispositivos móveis existentes, que oferecem suporte à linguagem, é garantida. Em outras palavras, o desenvolvedor, na maioria dos casos, não precisa se preocupar com o tipo de aparelho no qual o jogo será executado, pois a plataforma garante o funcionamento do jogo em qualquer dispositivo de um determinado perfil.

<sup>1</sup>Software Development Toolkit

A máquina virtual *KVM*<sup>2</sup> [20], utilizada pela plataforma, garante a compatibilidade entre vários dispositivos. Porém, esta técnica compromete o desempenho se comparada a outras plataformas. Na Figura 2, ilustra-se a arquitetura de execução da máquina virtual. As bibliotecas de configuração (*CLDC*<sup>3</sup> [17]) e os perfis se encontram acima da máquina virtual que se comunica com o sistema operacional do dispositivo hospedeiro.

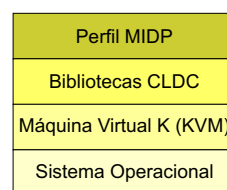


Figura 2: Arquitetura de execução da plataforma *J2ME*

Em sua versão atual (2.0), está presente um suporte mais voltado para aplicações multimídia, mais especificamente jogos. Com isso, a plataforma conseguiu amenizar a falta de suporte ao desenvolvimento de jogos da versão anterior. Essa nova versão já dá suporte à manipulação de imagens (rotação, espelhamento, etc), reprodução de som (inexistente em sua versão anterior), criação de camadas (permitindo uma fácil elaboração de cenários) e, entre outras melhorias, aprimora o tratamento da entrada de dados pelo usuário, permitindo o pressionamento de várias teclas simultaneamente.

### 3.2 BREW

Lançada em 2001 pela empresa *Qualcomm*, a plataforma *BREW* (*Binary Runtime Environment for Wireless*) [9] engloba não só uma plataforma de desenvolvimento, mas também um ambiente de execução, um sistema de distribuição de aplicativos, um serviço de teste de aplicativos e vários outros serviços de auxílio ao desenvolvedor. Esta plataforma foi concebida para ser executada em redes de telefonia *CDMA*<sup>4</sup> [1]. No entanto, testes bem sucedidos já foram realizados com dispositivos operando em redes *GSM*<sup>5</sup> [6] /*GPRS*<sup>6</sup> [7], oferecendo uma promessa de expansão desta solução. A arquitetura da plataforma pode ser vista na Figura 3. No Brasil, a plataforma é disponibilizada pela operadora *VIVO*<sup>®</sup>, única a trabalhar com a tecnologia *CDMA* no país.

<sup>2</sup>Kilo Virtual Machine

<sup>3</sup>Connected Limited Device Configuration

<sup>4</sup>Code Division Multiple Access

<sup>5</sup>Global System for Mobile communications

<sup>6</sup>General Packet Radio Service

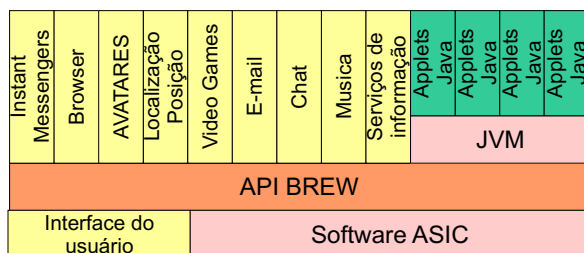


Figura 3: Arquitetura da plataforma BREW

A plataforma BREW possui como linguagem nativa a linguagem C/C++. Porém, atualmente ela já oferece suporte a aplicações desenvolvidas em outras linguagens, como Java. Utilizando a linguagem nativa da plataforma o desenvolvedor perceberá um ganho considerável de desempenho. Isto se dá também pelo fato da plataforma comunicar-se diretamente com o sistema operacional em execução no dispositivo, sem a necessidade de máquinas virtuais.

No modelo de negócios utilizado pela *Qualcomm*, o desenvolvedor só pode disponibilizar seus jogos através do serviço oferecido por esta companhia, pagando-se uma taxa para tal serviço. Esta taxa garante a disponibilização do jogo e a qualidade do mesmo, pois, antes de ser disponibilizado para download, cada jogo ou aplicativo é testado para garantir seu perfeito funcionamento e qualidade.

O problema com esta abordagem é que a plataforma não é livre para o desenvolvimento, não sendo possível desenvolver aplicativos e disponibilizá-los livremente. Isto diminui consideravelmente o número de pequenos desenvolvedores a optar pela plataforma. Este modelo aparenta favorecer mais aos usuários finais, que contam com um ambiente centralizado e seguro para download de jogos e outros aplicativos. Todos os jogos para a plataforma BREW são descarregados através do ambiente disponibilizado juntamente com a plataforma, garantindo segurança, uma vez que todas as operações de download possuem uma assinatura digital. Porém, para os desenvolvedores, a grande vantagem deste modelo é a prevenção contra a pirataria, visto que os jogos não poderão ser adquiridos por outro meio a não ser pelo servidor da própria plataforma.

Atualmente, a plataforma BREW conta com suporte de várias operadoras ao redor do mundo, já contando com mais de 180 milhões de downloads de aplicações nela baseadas [9]. Porém, a longa curva de aprendizado desta plataforma e o modelo de negócios, que impede a entrada de pequenos desenvolvedores, fazem com que ela ocupe o segundo lugar no número de jogos distri-

buidos e, conseqüentemente, vendidos no mercado.

A plataforma BREW conta com um excelente suporte ao desenvolvimento de jogos e sua API está entre as mais completas para o desenvolvimento deste tipo de aplicação. Nesta plataforma é possível criar arquivos de recursos, que englobam todos os elementos a serem utilizados (imagens, sons e textos), facilitando assim o gerenciamento de tais elementos. A plataforma oferece suporte à exibição de imagens 2D, criação de camadas gráficas que facilitam a construção de *backgrounds* e labirintos, além de acesso às funções de som e vibração do aparelho. É possível também a criação e manipulação de arquivos e de bancos de dados para armazenamento permanente de dados do jogo.

### 3.3 ExEn

Desenvolvida pela *In-Fusion*, empresa que hoje lidera o mercado europeu de jogos para dispositivos móveis [15], a *Execution Engine*, ou *ExEn* [8], foi a primeira plataforma dedicada à execução e desenvolvimento de jogos para dispositivos móveis voltada ao mercado massivo europeu [14]. Estando distribuída através de 6 operadoras de telefonia, envolvendo assim sete países europeus e um asiático (China), a *In-fusion* fornece toda a infraestrutura necessária à publicação, desenvolvimento e licenciamento de jogos para a *ExEn*.

Em seu modelo de negócio são definidos dois programas, ou níveis: o Programa de Desenvolvedor Padrão (*Standard Developer Program* ou *SDP*) e o Programa de Desenvolvedor Premium (*Premium Developer Program* ou *PDP*). O *SDP* provê todo o suporte de software necessário para que o desenvolvedor possa não só implementar, mas também simular seus jogos. Portanto, estão inclusos neste programa um *SDK*, contendo um *IDE* e editores de telas e animação, e um simulador para os jogos da plataforma, ambos distribuídos de forma gratuita. O membros do programa *SDP* contam ainda com suporte técnico *on-line*, também gratuito. Os desenvolvedores que participam do *PDP* possuem, além das vantagens oferecidas pelo *SDP*, a chance de disponibilizar seus jogos através das operadoras de telefonia. Enquanto qualquer desenvolvedor pode fazer parte do *SDP*, aqueles que desejam participar do *PDP* devem passar por um processo seletivo, no qual são levados em conta dois critérios: experiência na área de jogos (i.e., video-games) e de tecnologia móvel. O pré-requisito básico para desenvolver jogos voltados à plataforma é possuir algum conhecimento de Java, linguagem na qual *ExEn* está baseada.

Ocupando pouco espaço no dispositivo do usuário (100Kb de memória ROM e 32Kb de RAM), a plataforma conta com uma máquina virtual cujo desempe-

nho pode superar em até 30 vezes uma genérica e permite, além da execução, o download de jogos através das operadoras telefônicas. A implementação de sua arquitetura, ilustrada na Figura 4, é realizada sobre a especificação CLDC<sup>7</sup> e funções nativas do sistema operacional, de forma a acelerar o processamento gráfico da plataforma. Com isso, *ExEn* provê aos seus desenvolvedores uma API Java que, estando implementada sobre as bibliotecas nativas, fornece interoperabilidade entre os dispositivos que a suportam.

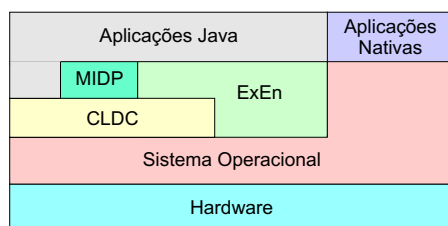


Figura 4: Arquitetura da plataforma *Exen*

As aplicações que rodam sobre *ExEn* são executadas em uma única linha de execução (i.e., *single threaded*) e dirigidas a eventos. Cada aplicação possui uma classe principal, *Gamelet*, a partir da qual os jogos podem ser carregados e executados, assim como acontece com as classes *Applet* existentes em *applets* Java.

O módulo gráfico de *ExEn* suporta até 65K de cores e contém, além dos mecanismos básicos para a construção de jogos em 2D, pacotes com suporte a gráficos 3D. Dentre esses mecanismos básicos encontram-se o *zoom*, *flipping* e *mirroring* de *sprites* (inclusive *sprites* em *flash*), *scrolling parallax*, *raycasting*, rotação de tela, dentre outros. O suporte 3D da plataforma inclui funções para operações em vetores e matrizes 3D e primitivas de renderização que utilizam triângulos em *wireframes* ou texturizados.

*ExEn* possui ainda pacotes específicos que permitem aos jogos reproduzir melodias e até mesmo fazer o dispositivo do usuário vibrar. As melodias são adicionadas às aplicações através do *SDK*.

Através de um componente especial da plataforma, o *ExEn connectivity*, é possível comunicar os jogos com servidores centrais das operadoras, via protocolo *SMS*<sup>8</sup> [3]. No entanto, essa funcionalidade só é acessível aos desenvolvedores que fazem parte do programa *PDP*, descrito anteriormente.

Algumas questões de segurança também são consideradas pela *ExEn*, prevenindo os dispositivos contra

falhas na execução dos jogos e garantindo que nenhum deles possa realizar alguma ação maléfica.

### 3.4 Mophon

Desenvolvida pela empresa *Synergenix*, *Mophon* [21] é uma solução para desenvolvimento de jogos para dispositivos móveis que visa principalmente aproximar o desenvolvimento de jogos para tais dispositivos do desenvolvimento de jogos tradicional. Sendo considerada a mais avançada e completa solução de desenvolvimento de jogos para dispositivos móveis [12], a plataforma foi desenvolvida ao longo de dois anos, estando dividida em quatro módulos principais, descritos a seguir.

- *Mophon RTE (Run Time Engine)*: mecanismo que gerencia a execução dos jogos nos dispositivos. Como forma de segurança, o *Mophon RTE* é executado em um ambiente de *sandbox* [19], impedindo os jogos de realizarem ações maliciosas.
- *Mophon API (Application Programming Interface)*: conjunto de funções disponíveis aos desenvolvedores e suportadas pelo *Mophon RTE*.
- *Mophon SDK (Software Development Kit)*: provê um conjunto de ferramentas integradas, liberadas gratuitamente, que permitem aos desenvolvedores fazerem uso da API da plataforma.
- *Mophon VST (Vendor Signing Tool)*: ferramenta que permite ao desenvolvedor distribuir seu jogo, determinando o dispositivo sobre o qual o mesmo deverá ser executado.

Contando com mais de 23.000 desenvolvedores registrados [23], a plataforma possui versões executáveis para telefones celulares, *smart phones* e *PDAs*, estando o seu modelo de negócios dividido basicamente em quatro passos. Inicialmente, os desenvolvedores podem, opcionalmente, submeter à *Synergenix* a idéia do jogo, sendo aceito aqueles que se encaixam nos requisitos da empresa. No segundo passo, de certificação, o jogo deve ser submetido à empresa, que irá verificar se o mesmo possui algum conteúdo ofensivo ou falhas de segurança. Outros testes serão executados para verificar se o funcionamento do jogo segue os padrões definidos pela empresa e, em caso de aprovação, o mesmo recebe uma assinatura digital. Estando certificado, o jogo poderá então ser publicado (passo 3) e posteriormente distribuído (passo 4), quando os usuários terão a possibilidade de obtê-lo utilizando *WAP*<sup>9</sup> [13] ou *MMS*<sup>10</sup> [2].

<sup>7</sup> Connected Limited Device Configuration

<sup>8</sup> Short Messaging System

<sup>9</sup> Wireless Application Protocol

<sup>10</sup> Multimedia Message Service

A plataforma, cujos jogos são desenvolvidos em C ou C++, ocupa 50Kb de memória ROM, necessitando de mais 100Kb quando incluída uma biblioteca 3D otimizada. Seu módulo gráfico encontra-se dividido em três sub-módulos: o *Mecanismo de sprites ultra-rápido* (*Ultra fast sprite engine*); o *Mecanismo de mapas de background* (*Background map engine*); e o *Sistema de colisão* (*Collision system*). No primeiro mecanismo, são fornecidas características como transparência, conversão de cores entre diferentes plataformas, ampla diversidade de modos gráficos (10 no total) e *sprites* de tamanhos variados. O *Mecanismo de mapas de background* provê, dentre outras coisas, camadas transparentes e dois modos de *scrolling* de múltiplas camadas, normal e *parallax*. Dentre as características do *Sistema de colisão* pode-se citar o cálculo automático de colisões em *background*. Funções de renderização estão disponíveis tanto para gráficos em 2D quanto em 3D. A plataforma ainda provê uma API contendo funções para cálculo de matrizes e vetores 3D, definição de materiais e aplicação de diferentes tipos de luz.

O suporte à multimídia é também bastante variado, suportando por exemplo, a reprodução de toques MIDI polifônicos. É possível ainda reproduzir efeitos sonoros simultâneos e em *background*, e controlar tanto a frequência quanto o volume dos sons reproduzidos.

Um outro ponto forte de *Mophun* é seu suporte à segurança. Como citado anteriormente, o mecanismo de execução *Mophun RTE* é executado em um ambiente de *sandbox*, protegendo assim tanto o hardware quanto o conteúdo do dispositivo de aplicações maliciosas. Ainda no quesito segurança, a plataforma conta com um sistema de DRM<sup>11</sup> [22] embutido, fornecendo aos desenvolvedores proteção contra pirataria. Além disso, todos os jogos são assinados utilizando um algoritmo de chave pública.

No aspecto de conectividade, *Mophun* permite que aplicações em diferentes dispositivos se comuniquem através de *sockets*, possivelmente através de conexões paralelas, fazendo uso das interfaces de comunicação do dispositivo (e.g., *Bluetooth* [5], *GPRS*). Uma característica importante é o fato de que o código utilizado para implementar a comunicação entre os dispositivos é o mesmo, independente da conexão utilizada (e.g, *TCP/IP*<sup>12</sup>, *Infra-vermelho*). Além dessas funcionalidades, ainda é possível o envio de *stream* de dados e mensagens *SMS*. Porém, mesmo com todo o avanço apresentado pela plataforma para o desenvolvimento de jogos, a mesma está disponível em uma quantidade muito pequena de dispositivos, ficando seu mer-

cado bastante comprometido se comparada com outras plataformas.

## 4 Estudo de caso

Nesta seção é apresentado um estudo de caso de forma a ilustrar o processo de escolha de uma plataforma no desenvolvimento de um projeto de jogo. Esta escolha depende da estratégia a ser adotada pelo desenvolvedor, suas necessidades e até mesmo seu orçamento. A escolha da plataforma para a construção do jogo aqui apresentado foi feita com base nos parâmetros analisados neste artigo.

### 4.1 Apresentação do jogo

O estudo de caso em questão se trata de um jogo de ação, chamado de “A balada do cangaceiro”, no qual o jogador deve pressionar teclas no celular que correspondem às posições onde os inimigos aparecem na tela, conforme apresentado na Figura 5. Caso o inimigo esteja visível no momento em que a tecla for pressionada o mesmo é dado como morto e um valor é acrescido à pontuação do jogador.



Figura 5: Tela do jogo com indicações de posição.

A seguir é apresentada a análise de cada plataforma baseando-se no parâmetros aqui considerados e nas características do jogo desenvolvido.

### 4.2 Linguagem de desenvolvimento

As plataformas analisadas possuem como linguagem de desenvolvimento as linguagens C, C++ e Java. O desenvolvedor conta com um conhecimento mais aprofundado em Java. Sendo assim, as plataformas que inicialmente se apresentam mais propícias são *J2ME* e *ExEn*.

- Resultado: *J2ME* e *ExEn*.

<sup>11</sup>Digital Rights Management Systems

<sup>12</sup>Transmission Control Protocol/Internet Protocol

	Custo para desenvolvimento	Modelo de Negócios	Carga
<b>J2ME</b>	- Completamente gratuito	- Desenvolvedor responsável por todas as etapas, do desenvolvimento à distribuição	- Interpretado - 128Kb à 512Kb de memória (RAM e ROM)
<b>BREW</b>	- Pago para disponibilização do jogo - IDE's pagas	- Empresa realiza testes e disponibiliza as aplicações	- Aplicativos compilados - 90Kb de ROM
<b>ExEn</b>	- Completamente gratuito - PDP permite disponibilização através de operadoras	- Programas (PDP e SDP) regem as ferramentas disponibilizadas	- Rápido, pois utiliza a API nativa do dispositivo - 100Kb de ROM e 32K de RAM.
<b>Mophun</b>	- Plataforma gratuita - IDE's pagas	- Ferramentas distribuídas com a plataforma auxiliam na distribuição	- Processamento rápido. - 50Kb à 150Kb na memória.

(a) Tabela comparativa 1

	Linguagem de desenv.	Segurança	Suporte
<b>J2ME</b>	Java	- Os jogos não possuem o funcionamento garantido. - A plataforma protege o dispositivo de aplicações mal intencionadas.	- Utilização de camadas facilitando a inserção de elementos visuais - Verificação automática de colisão entre sprites - Construção simplificada de labirintos com a classe TiledLayer
<b>BREW</b>	C++	- Aplicativos testados pela <i>Qualcomm</i> e assinados digitalmente	- Utilização de arquivos de recursos facilita a gerência de elementos do jogo - Maior desempenho ao utilizar a linguagem nativa
<b>ExEn</b>	Java	- Previne os dispositivos contra falhas nos jogos e ações maliciosas.	- Várias operações nativas para tratamento de gráficos 3D - Suporte nativo à raycasting
<b>Mophun</b>	C / C++	- Mecanismo de execução em um ambiente de <i>sandbox</i> . - Os jogos são assinados digitalmente e protegidos contra pirataria. - Cada jogo é exaustivamente testado antes de ser liberado.	- Detecção automática entre sprite e background - 3 módulos providos com o SDK: sprites, background e collision - API 3D provendo vários tipos de materiais e aplicações de luz

(b) Tabela comparativa 2

### 4.3 Segurança

Dentre as plataformas consideradas, *J2ME* e *ExEn* não possuem mecanismos para testes e criação de assinaturas digitais. Porém, todas as plataformas previnem o dispositivo contra ações maliciosas dos aplicativos. Devido à maior segurança contra pirataria, as plataformas *BREW* e *Mophun* foram consideradas mais adequadas neste parâmetro.

- Resultado: *BREW* e *Mophun*.

### 4.4 Custo para desenvolvimento

O custo envolvido para o desenvolvimento de um jogo leva em conta vários aspectos, tais como: valor de li-

cença de uma *IDE*<sup>13</sup>, custo para aquisição da plataforma e custo para disponibilização do jogo. Nestes pontos, tem-se o resultado descrito a seguir.

- *J2ME*: *SDKs* gratuitos estão disponíveis no mercado, possivelmente integrados com algumas *IDEs*, também gratuitas, como Eclipse [4] e Netbeans [10]. A disponibilização dos jogos fica a critério do desenvolvedor.
- *BREW*: as *IDEs* encontradas para desenvolvimento nesta plataforma são pagas, como *.NET* [11], assim como a disponibilização dos jogos, realizada através dos servidores da plataforma. Apenas a

<sup>13</sup>Integrated Development Environment

aquisição do *SDK* e o desenvolvimento a nível de testes é gratuito.

- *ExEn*: a plataforma disponibiliza gratuitamente um *SDK* e um *IDE*. A disponibilização dos jogos, também gratuita, pode ser realizada de duas maneiras, dependendo do programa do qual o desenvolvedor faz parte, *SDP* ou *PDP*. No programa *SDP* a disponibilização fica a critério do desenvolvedor ao passo que, no *PDP*, a mesma é feita através das operadoras. Porém, para participar do programa *PDP* os desenvolvedores precisam passar por um processo de seleção.
- *Mophun*: o *SDK* da plataforma é completamente gratuito, ficando a disponibilização dos jogos a critério do desenvolvedor. No entanto, boas *IDEs* gratuitas para a linguagem C++ são difíceis de serem encontradas. Dessa forma, é provável que exista algum custo relacionado à aquisição de um bom ambiente de desenvolvimento.
- Resultado: *J2ME* e *ExEn*.

#### 4.5 Modelo de negócios

Como mencionado anteriormente no apresentado, a plataforma *BREW* é a única que traz algum custo associado à disponibilização de jogos. Portanto, considerando este parâmetro e as plataformas apresentadas, *BREW* é a única que não se mostra interessante para o estudo de caso em questão, no qual o objetivo é realizar todas as etapas de desenvolvimento e disponibilização com o menor custo possível.

- Resultado: *J2ME*, *ExEn* e *Mophun*.

#### 4.6 Carga

Como o jogo a ser construído não requer muitos recursos de hardware e software, todas as plataformas são classificadas como adequadas ao projeto.

- Resultado: *J2ME*, *BREW*, *ExEn* e *Mophun*.

#### 4.7 Suporte a elementos de jogo

Para a criação do projeto busca-se uma plataforma que ofereça mecanismos para um bom tratamento de entrada de dados e colocação de imagens em camadas. Com base nestes requisitos todas as plataformas são adequadas ao desenvolvimento do jogo em questão.

- Resultado: *J2ME*, *ExEn* e *Mophun*.

#### 4.8 Considerações sobre a escolha

De acordo com a análise realizada, para o estudo de caso deste artigo, *J2ME* aparece como a melhor escolha dentre as plataformas consideradas. Isto se deve ao prévio conhecimento de sua linguagem nativa por parte do desenvolvedor e ao menor custo envolvido no desenvolvimento e disponibilização de seus jogos.

Além disso, como descrito anteriormente, a plataforma com o maior número de aparelhos no mercado e, conseqüentemente, maior número de usuários é *J2ME*. Como o objetivo de produção do jogo é obter a maior quantidade possível de jogadores, *J2ME* torna-se a melhor opção.

### 5 Conclusões

Neste artigo foram apresentadas algumas características, vantagens e desvantagens das principais plataformas para desenvolvimento de jogos para celulares existentes. Foram considerados tanto os aspectos técnicos quanto econômicos de cada plataforma.

A análise realizada deverá auxiliar o desenvolvedor na escolha da plataforma que melhor se adequa às características de um projeto de jogo - linguagem, carga, modelo de negócio, custo para desenvolvimento e segurança.

Para ilustrar a escolha de uma determinada plataforma baseada na análise realizada, foi apresentado um estudo de caso, desenvolvido utilizando *J2ME*.

### Referências

- [1] CDMA Development Group. <http://www.cdg.org> - Acessado em 06/04/2005.
- [2] Favorites, G. Introduction to Multimedia Messaging Service (MMS). <http://www.gsmfavorites.com/mms> - Acessado em 06/04/2005.
- [3] Favorites, G. Short Message Service (SMS). <http://www.gsmfavorites.com/sms> - Acessado em 06/04/2005.
- [4] Foundation, E. <http://www.eclipse.org> - Acessado em 06/04/2005.
- [5] Group, B. S. I. <https://www.bluetooth.org/> - Acessado em 06/04/2005.
- [6] GSM Association. <http://www.gsmworld.com> - Acessado em 06/04/2004.
- [7] GSM Association. GPRS Platform. <http://www.gsmworld.com/technology/gprs> - Acessado em 06/04/2005.



- [8] In-Fusio. <http://developer.in-fusio.com> - Acessado em 06/04/2005.
- [9] Inc., Q. <http://brew.qualcomm.com> - Acessado em 06/04/2005.
- [10] Inc., S. M. <http://www.netbeans.org> - Acessado em 06/04/2005.
- [11] Microsoft. Visual C++ .Net. <http://msdn.microsoft.com/visualc> - Acessado em 06/04/2005.
- [12] Obigo. Mophon gaming platform. <http://www.obigo.com/PSUser/servlet/com.ausys.ps.web.user.servlet.AuthorizedPageServlet?nodeid=1708&pageversion=2> - Acessado em 06/04/2005, 2004.
- [13] Open Mobile Alliance. <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html> - Acessado em 06/04/2005.
- [14] Pedro Amaro. The Clash of Mobile Platforms: J2ME, ExEn, Mophon and WGE. <http://www.gamedev.net/reference/business/features/clashmobile/> - Acessado em 06/04/2005, 2003.
- [15] Phonegamereview.com. In-fusio's EGE Gaming Solution For Orange Signature Phones Across Europe. <http://www.phonegamereview.com/phone-game-news/article.asp?a=714> - Acessado em 06/04/2005, Julho 2004.
- [16] Review, W. G. Wireless Game Directory: Advanced Search. [http://wgamer.com/gamedir/adv\\_search.php](http://wgamer.com/gamedir/adv_search.php) - Acessado em 06/04/2005, 2005.
- [17] Sun Microsystems Inc. Connected Limited Device Configuration (CLDC). <http://java.sun.com/products/cldc> - Acessado em 06/04/2005.
- [18] Sun Microsystems Inc. Java 2 Platform Micro Edition (J2ME). <http://java.sun.com/j2me> - Acessado em 06/04/2005.
- [19] Sun Microsystems Inc. Java Security Architecture. <http://java.sun.com/security> - Acessado em 06/04/2005.
- [20] Sun Microsystems Inc. The K virtual machine (KVM). <http://java.sun.com/products/cldc/wp> - Acessado em 06/04/2005.
- [21] Synergix. <http://www.mophon.com> - Acessado em 06/04/2005.
- [22] Systems, I. Mobile Digital Rights Management. <http://www.iprsystems.com/whitepapers/Mobile-DRM-WP.pdf> - Acessado em 06/04/2005, Outubro 2003.
- [23] Teleca. Let Teleca and Mophon Give you Instant Access to the Mobile Entertainment and Gaming Market. <http://www.obigo.com/PSUser/mediacache/1702/1710/mophon040203.pdf> - Acessado em 06/04/2005, 2003.