

Serviços de Adaptação de Jogabilidade para Jogos Multiplataforma Multiusuário

Davi Pedrosa, Fernando Trinta, Carlos Ferraz, Geber Ramalho
Centro de Informática - Universidade Federal de Pernambuco
Av. Professor Luis Freire s/n Cidade Universitária
Caixa Postal 7851, Recife-PE-Brasil
{dvp, famt, cagf, glr}@cin.ufpe.br

ABSTRACT

This paper presents a particular view for future Networked Multiplayer Games. These applications will include features from pervasive computing, allowing players to enjoy them using different devices, such as PDAs and Mobile Phones, almost anywhere, anytime. In this document, we call these applications Pervasive Multiplayer Multiplatform Games (PM2G). We present scenarios that show intended characteristics for these games, describing specific PM2G concepts and the players' interaction. As the main contribution of this paper, the design and implementation of two services are detailed. They perform player's playability adaptation according to context information, such as the current device in use. Finally, common game scenarios are created to validate the functionality.

1. INTRODUÇÃO

Nos últimos anos, tem ocorrido um grande desenvolvimento na área de jogos multiusuário *online* - jogos em que várias pessoas participam utilizando computadores distintos conectados através de uma infra-estrutura de comunicação. Este progresso deve-se principalmente ao avanço das tecnologias de comunicação e ao sucesso da Internet, despertando inclusive um crescente interesse comercial nestas aplicações.

Enquanto isso, também evolui e populariza-se cada vez mais a computação móvel. A inerente conectividade dos dispositivos móveis, somada ao surgimento de plataformas de desenvolvimento como J2ME [9] resultam em uma oferta cada vez maior de serviços para esses dispositivos. Nos últimos anos, o aparecimento de jogos multiusuário *online* para telefones celulares evidencia o potencial da computação móvel para a área de entretenimento digital. Este potencial se traduz em uma grande mudança na maneira como um jogador utilizando um dispositivo móvel se relaciona com um jogo. Uma das possíveis abordagens permite ao jogador permanecer sempre conectado ao jogo a partir de seu dispositivo.

Esta situação se contrapõe à tradicional idéia de jogos multiusuário em computadores pessoais, na qual jogadores possuem máquinas com grande poder computacional, dispendo geralmente de uma conexão de alta velocidade.

Atualmente, vários jogos exploram individualmente estes diferentes cenários, cada um dos quais composto por diferentes tipos de plataformas. Sendo assim, um jogo específico pode oferecer aos usuários características interessantes da plataforma para o qual foi desenvolvido, como mobilidade, para o caso de jogos multiusuário móveis, ou uma interface gráfica rica e altamente interativa, para os jogos de computadores de mesa ou consoles de videogame.

Porém, pode-se imaginar um cenário ainda mais inovador, onde um mesmo jogo seria utilizado a partir de plataformas distintas, com cada uma delas provendo uma experiência de jogo própria ao usuário. Este jogo poderia ser encarado pelos jogadores de diferentes maneiras, dependendo de como eles o estivessem acessando. Desta forma, vislumbra-se a incorporação do conceito de computação pervasiva[7], considerada uma evolução ao atual momento da computação móvel. No entanto, a infra-estrutura sugerida neste trabalho para a incorporação de pervasividade é aquela já presente e disponível em nosso cotidiano, como a aquela disponibilizada por operadoras de telefonia que oferecem acesso à Internet. Fica portanto restrito o ambiente de computação pervasiva a ser utilizado, haja vista a inviabilidade de todas as possibilidades que a mesma oferece. Neste sentido, as características pervasivas abordadas neste trabalho são (i) permitir que os usuários possam ter acesso à informação, de qualquer local, podendo utilizar diferentes dispositivos e suas possibilidades de conexão, e (ii) adaptar a informação recebida ou enviada pelos dispositivos de acordo com o contexto do jogador.

As diferenças existentes nos dispositivos em termos de usabilidade, interfaces de apresentação, características de comunicação e capacidade de processamento exigem que as interações e as formas de apresentação do jogo sejam adaptadas às características do dispositivo utilizado pelo jogador em um dado instante. Desta forma, seria extremamente custoso para desenvolvedores a implementação de um jogo multiplataforma multiusuário. A solução mais viável é a criação de uma infra-estrutura que amenize o esforço requerido para a implementação destes jogos, reduzindo seus custos e tempo de desenvolvimento. Atualmente, já existem diversos *middleware* específicos para jogos, que facilitam a manutenção

de estado compartilhado e comunicação entre jogadores. No Centro de Informática da Universidade Federal de Pernambuco está em desenvolvimento uma tese de doutorado cujo objetivo é a implementação de um *middleware* para jogos pervasivos massivos multiusuário multiplataforma (*Pervasive Multiplayer Multiplatform Game - PM2G*)[11][10]. Uma aplicação PM2G é um jogo multiusuário *online* que permite acesso de forma coerente a partir de diferentes dispositivos usados (Computadores Pessoais, Telefones Celulares ou PDAs), apesar da grande diferença existente entre seus contextos. Além disso, um PM2G também aproveita situações e serviços específicos do cenário móvel para permitir interações mais intensas entre jogadores móveis, como a formação de redes espontâneas através de diferentes tecnologias de comunicação e o uso de recursos baseados na localização do usuário (*Location Based Services - LBS*).

A caracterização feita para jogos PM2G levanta uma série de questões relativas à sua efetiva realização pois são agregados problemas de áreas distintas. Primeiramente, no que diz respeito aos jogos multiusuário, destacam-se a necessidade de suporte a escalabilidade de usuários e o combate ao uso de trapaça por parte de alguns jogadores[8]. Em relação aos aspectos pervasivos, a necessidade de suporte a um diversificado conjunto de dispositivos requer adaptações no modo como jogadores percebem e interagem com o jogo[7]. Em consequência, esta heterogeneidade impõe a idéia de uma competição justa (do inglês, *fairness*) entre jogadores, mesmo quando da existência de diferentes condições de acesso. O uso de dispositivos móveis traz consigo problemas relativos ao alto consumo de bateria em virtude da necessidade de comunicação sem fio com o mundo virtual.

Este trabalho aborda parte destes problemas, principalmente aqueles relacionados com a necessidade de adaptação da jogabilidade em face ao uso de dispositivos heterogêneos. Embora todas as questões citadas anteriormente (escalabilidade, combate a trapaça, competição justa, consumo de bateria, etc.) tenham sua relevância na proposta PM2G, estas já vêm sendo estudadas por diversos grupos de pesquisa[1][4][2], e soluções advindas destes estudos podem ser então agregadas ou adaptadas à realidade de nossa proposta.

A primeira contribuição deste trabalho é de natureza arquitetural, onde apresentamos dois serviços que objetivam permitir e facilitar a participação de jogadores em um jogo multiplataforma. São apresentados seus requisitos e arquitetura através de diagramas em UML[6]. Nossa segunda contribuição apresenta resultados obtidos na validação de nossas idéias, através de cenários implementados via prototipagem.

Este artigo está organizado em cinco seções. A segunda seção apresenta os principais conceitos da proposta PM2G. A terceira destaca os serviços de adaptação de jogabilidade, apresentando seus requisitos e suas respectivas arquiteturas. Na quarta seção são apresentados resultados de um protótipo desenvolvido, com objetivo de validar as idéias apresentadas. Por fim, a quinta seção apresenta conclusões e trabalhos futuros.

2. A PROPOSTA PM2G

Cenários para jogos multiusuário futuros indicam uma tendência onde estas aplicações serão ambientes virtuais persistentes, acessados por um alto número de jogadores, através de diferentes dispositivos que variam em condições de comunicação e processamento[12][5]. A proposta PM2G apresenta uma visão onde estes jogos levam em consideração informações de contexto, como localização física do jogador, para guiar a participação do usuário no jogo. São agregadas características advindas de diversos cenários atuais para jogos multiusuário, como Jogos Massivos, Jogos em Dispositivos Móveis e Jogos Pervasivos. Objetiva-se com isso, oferecer ao jogador um ambiente altamente interativo e inovador. O acesso do usuário ao mundo virtual será irrestrito sempre que possível, criando situações onde jogadores utilizando diferentes dispositivos possam estar lado a lado dentro do mundo virtual, competindo ou cooperando de acordo com os objetivos do jogo. Ao mesmo tempo, nossa visão defende que estes jogos aproveitarão situações específicas, como a formação de redes espontâneas via dispositivos móveis, objetivando aumentar a imersão do usuário no jogo.

Em seus trabalhos iniciais, a proposta PM2G estabeleceu cenários que indicam as situações possíveis para jogadores que participem de um jogo com as características pretendidas. A partir destes cenários, foram especificados dois modelos que nos permitiram a definição de conceitos que unificam as idéias estabelecidas pelas interações entre jogadores em jogos PM2G. O primeiro, chamado de Modelo de Aplicação, diz respeito à visão de elementos que compõem o jogo, enquanto o segundo, chamado Modelo de Utilização, ressalta a forma como usuários interagem com a aplicação. A Figura 1 apresenta alguns dos principais conceitos do Modelo de Aplicação PM2G.

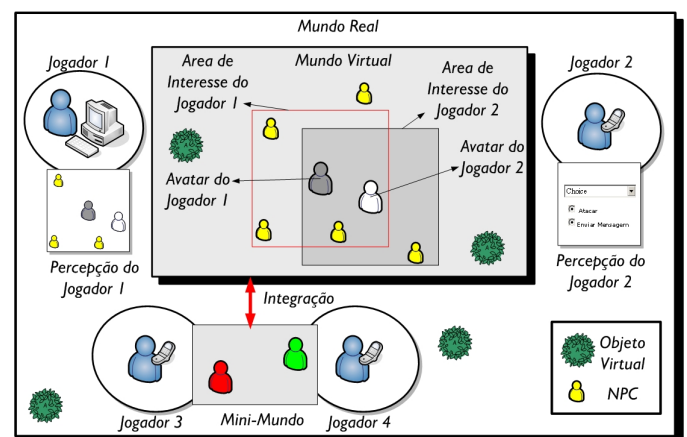


Figura 1: Diagrama de classes do Serviço de Adaptação de Conteúdo

Objeto é o elemento básico que compõe o conjunto de informações dinâmicas do mundo virtual. Em outras palavras, um objeto é qualquer informação que pode ser adicionada, copiada, alterada ou removida do mundo virtual. Exemplos típicos de objetos são comidas, armas ou ferramentas. *Personagens automatizados*, também conhecidos por NPCs (do inglês, *Non-Player Characters*), são personagens cujas ações não são controladas por um jogador. Podem representar inimigos, aliados ou simplesmente espectadores (*bypassers*).

Área de interesse é um subconjunto do mundo virtual com o qual o jogador pode interagir em um dado instante, e que portanto é mais relevante para ele naquele momento.

Contexto de execução relaciona-se com a infra-estrutura de acesso de um jogador ao mundo virtual, incluindo o tipo de rede e o dispositivo utilizado para acessar o jogo, bem como preferências pessoais do jogador. *Percepção* é a forma com que o jogo é apresentado ao jogador, sendo influenciada diretamente pelo contexto de execução do jogador em um dado instante. *Interação* é a maneira como o jogador realiza suas ações no mundo virtual, que também é dependente do contexto de execução e da percepção do jogador em um dado momento. Por fim, o conceito de *Mini-mundo* está relacionado com o aproveitamento de novas oportunidades de interação, específicas para jogadores móveis. Mini-mundos constituiriam-se como cenários paralelos com interação direta entre jogadores via dispositivos móveis, através de redes espontâneas ou redes locais sem fio. Estes cenários podem ser compreendidos como partidas isoladas entre jogadores móveis, porém com conseqüências no mundo virtual simulado, de forma semelhante às conseqüências das interações convencionais entre jogadores dentro do mundo virtual.

Os cenários e modelos descritos impõem uma série de requisitos funcionais e não-funcionais a serem tratados por um modelo de suporte adequado a jogos com tais características. O modelo tradicional de suporte a jogos multiusuário utiliza uma camada de serviços na figura de um *middleware*. A proposta de suporte a um jogo PM2G segue esta abordagem, ampliando os serviços de *middleware* para jogos tradicionais através de serviços de suporte relativos aos conceitos específicos de seus modelos de aplicação e utilização. Uma aplicação PM2G utiliza uma tradicional arquitetura Cliente/Servidor, onde clientes de diferentes plataformas utilizam os serviços do *middleware* para interagirem com a simulação do jogo. Além dos serviços que são o maior foco deste documento, a saber, a Adaptação de Conteúdo e a Adaptação de Interação, são definidos mais quatro serviços:

1. **Gerenciamento de Contexto:** Contexto é um conceito chave em uma aplicação pervasiva[7]. Nestas aplicações, toda informação relevante e que caracterize a situação de uma entidade que compõe (*software/hardware*) ou interage (*usuário*) com a aplicação é chamada contexto. Aplicações pervasivas monitoram mudanças no contexto para adaptar seu funcionamento em relação a estas entidades.
De acordo com o modelo de aplicação PM2G, o contexto de execução de um jogador tem papel fundamental na forma como este usuário percebe e interage com o jogo, guiando o funcionamento dos demais serviços e as possibilidades de interação entre jogadores. O Serviço de Gerenciamento de Contexto tem por finalidade realizar o monitoramento reativo do contexto de execução de um jogador, para que adaptações necessárias para a participação do jogador possam ser realizadas pelos demais serviços.
2. **Serviço de Notificação de Eventos:** Este serviço tem a função de informar aos jogadores sobre acontecimentos de seu interesse, quando tais jogadores esti-

verem em um contexto móvel. Ele é o principal responsável por criar a idéia do jogador móvel estar sempre conectado ao jogo, sendo notificado através de diferentes meios de comunicação, como mensagens SMS (do inglês, *Short Message Service*) ou e-mail. Este serviço visa criar uma abstração para que jogadores móveis possam ser alertados sobre diferentes eventos relativos tanto de sua área de interesse do mundo virtual, quanto àqueles de seu mundo real.

3. **Serviço de Localização Física:** A localização física de um jogador móvel tem um papel importante na proposta PM2G, como dar suporte à inclusão de objetos virtuais no mundo real ou auxílio na formação dos mini-mundos. Este serviço tem por função rastrear a posição do jogador para que a mesma possa ser utilizada para estes fins.

Podem ser utilizadas diferentes abordagens de posicionamento para acompanhar o deslocamento de um jogador móvel. Estas abordagens variam de acordo com a disponibilidade de Serviços de Localização pela operadora de telefonia celular, ou mesmo uso de soluções baseadas no próprio dispositivo, como GPS.

4. **Serviço de Integração de Mini-Mundos:** Por fim, este serviço diz respeito à integração ao jogo dos cenários formados pelo conceito de *mini-mundo* do modelo de aplicação PM2G. Cada partida paralela iniciada precisa repassar ao jogo informações de seus jogadores, localizações e resultados finais destas partidas.

O estabelecimento de um mini-mundo se dá de duas formas. Primeiramente, um jogador móvel pode criar um mini-mundo utilizando seu dispositivo, e esperar adversários que se conectem ao mini-mundo criado. Em um segundo modo, jogadores podem se conectar a um mini-mundo em formação ou pré-estabelecido que ofereça infra-estrutura adequada à conexão entre jogadores móveis.

Maiores informações sobre a Proposta PM2G podem ser obtidas nas referências [10] [11].

3. SERVIÇOS DE ADAPTAÇÃO DE JOGABILIDADE

Além dos serviços citados na seção anterior, fazem parte do *middleware* PM2G dois serviços que visam facilitar a participação de jogadores móveis em um jogo multiusuário. São estes o Serviço de Adaptação de Conteúdo e o Serviço de Adaptação de Interação. O primeiro diz respeito a transformação do conceito de percepção da área de interesse de um jogador, definidos no modelo de aplicação PM2G, em formatos adequados ao contexto de execução de um jogador. O segundo visa permitir que as ações a serem realizadas através de dispositivos móveis possam ser realizadas de forma mais fácil pelos usuários.

De forma geral, foram definidos os seguintes requisitos para o projeto dos serviços: (i) Facilidade de uso: ambos os serviços devem ser fáceis de utilizar e de configurar pelo desenvolvedor do jogo; (ii) Flexibilidade: os serviços devem ser o mais genéricos possíveis, não se restringindo a uma categoria específica de jogos multiusuário; (iii) Extensibilidade: os

serviços devem ser extensíveis, permitindo a qualquer desenvolvedor que utilize o *middleware* aperfeiçoar ou especializar suas funcionalidades se assim desejar.

3.1 Serviço de Adaptação de Conteúdo

A utilização de diferentes plataformas traz a reboque o problema da heterogeneidade das características de cada uma delas. Diferentes tamanhos de tela, capacidade de memória e de processamento, largura de banda e latência da conexão com a rede são alguns exemplos. Este problema demanda que as informações enviadas ao jogador sejam adaptadas de acordo com o seu dispositivo utilizado, sendo apresentadas da forma mais adequada. Desta forma, é função do Serviço de Adaptação de Conteúdo realizar a transformação destas informações.

O serviço foi concebido para ser utilizado da seguinte forma: conhecendo o conteúdo da área de interesse do jogador e sabendo qual seu contexto de execução em um dado momento, o serviço é requisitado pelo servidor do jogo para que as informações a serem enviadas ao jogador sejam adaptadas para o seu dispositivo. Tais informações são então retornadas ao jogo, que as repassa ao jogador.

3.1.1 Modelagem

Para definir como cada dispositivo determinaria a adaptação de conteúdo a ser realizada, foi criado o conceito de *estratégia de adaptação*. Uma estratégia de adaptação é associada a cada dispositivo, e determina quais transformações são realizadas nas informações a serem enviadas ao jogador.

Para modelar este serviço, foi observado que o conjunto de dispositivos abordados pode ser ordenado no que diz respeito à quantidade de informações recebidas pelo jogador, característica que se relaciona com a sensação de imersão provocada no mesmo. Com isto em mente, para cada dispositivo apto a ser utilizado por um jogador, foi criada uma propriedade que representa seu nível de abstração. Quanto maior for este valor, menos informações o jogador que o utiliza irá receber. Um computador pessoal, por exemplo, pode ser considerado o dispositivo com o menor nível de abstração em um determinado jogo. Visto que o jogador que utiliza um computador pessoal deseja uma maior sensação de imersão no jogo, as informações precisam ser apresentadas ao jogador de maneira detalhada, exigindo muita troca de dados entre a aplicação cliente e o servidor. Já um aparelho celular de limitados recursos computacionais oferece uma mínima sensação de imersão ao jogador; sendo assim, são poucas as informações recebidas pela aplicação cliente.

Cada elemento da área de interesse também deve conter uma informação a respeito da sua importância, para que o serviço saiba quais objetos podem ser descartados e quais devem ser enviados ao jogador. Para isto, cada objeto do jogo possui uma propriedade que representa sua prioridade. Esta informação permite ao desenvolvedor do jogo criar categorias como, por exemplo, *essencial*, *importante* ou *opcional*, para cada ator, item ou personagem automatizado contido em uma área de interesse.

Diversos tipos de processamento podem ser realizados sobre as informações de conteúdo enviadas ao jogador. Neste trabalho, estes processamentos são divididos em duas ca-

tegorias: *processamentos lógicos* e *processamentos de apresentação*. A primeira categoria engloba procedimentos que modificam o conteúdo da área de interesse, como, por exemplo, a quantidade de objetos ou a posição deles no mundo virtual. Por sua vez, processamentos de apresentação preparam os dados para serem enviados à aplicação cliente. Dependendo da forma com que estas informações são enviadas e também do dispositivo utilizado, pode ser necessário que estas informações recebidas pela aplicação cliente sejam apresentadas ao jogador com a menor quantidade de processamento local possível.

A partir destas definições, cada forma de processamento é representada por um módulo de adaptação de conteúdo, sendo classificados como lógicos ou de apresentação. A Figura 2 apresenta a arquitetura do serviço através de um diagrama de classes em UML[6]. As interfaces *LogicModule* e *PresentationModule* representam as operações necessárias para módulos lógicos e de apresentação.

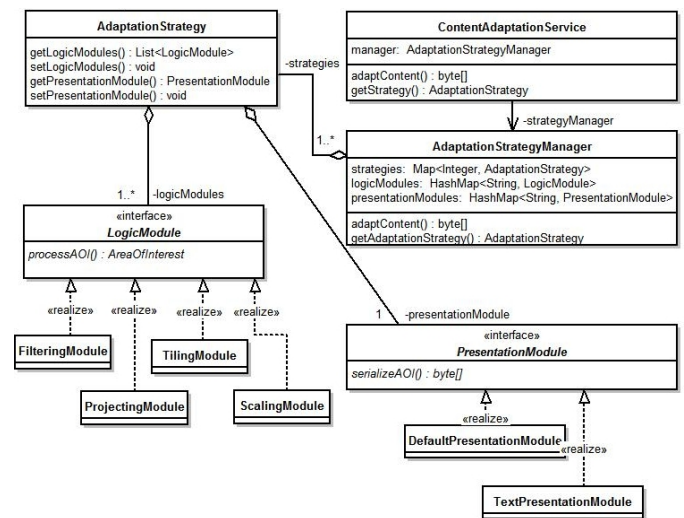


Figura 2: Diagrama de classes do Serviço de Adaptação de Conteúdo

A princípio, foram definidos os seguintes módulos lógicos:

- **Módulo de filtragem de informações:** baseando-se no nível de abstração do dispositivo e na prioridade de cada elemento da área de interesse, este módulo é responsável por descartar os elementos que não precisam ser enviados ao jogador. Isto traz duas vantagens: a primeira é a redução do tráfego na rede, o que pode ser importante para jogadores utilizando algum tipo de conexão cuja cobrança é dada pela quantidade de bytes transmitidos. Outra vantagem é a menor carga de processamento e memória exigidos do dispositivo, já que são menos objetos a serem processados pela aplicação cliente;
- **Módulo de escala bidimensional:** ajusta as posições dos objetos de uma área de interesse (bidimensional) baseando-se na resolução do dispositivo utilizado. Apesar de ser uma operação simples, o fato de ser realizada pelo serviço e não no dispositivo cliente diminui a carga de processamento neste último;
- **Módulo de tiling:** adapta as posições dos objetos de

uma área de interesse (bidimensional) para que possam ser apresentados como em jogos baseados em *tiles* (*tile-based games*). Nestes jogos, um mapa é formado por blocos bidimensionais de igual tamanho, que o preencham completamente e sem sobreposição. Logo, a posição de qualquer objeto apresentado neste mapa deve assumir uma quantidade limitada de valores, que podem ser mapeados por uma matriz com N linhas e M colunas;

- **Módulo de projeção tridimensional:** realiza a projeção tridimensional das posições dos objetos para valores bidimensionais. Isto é muito útil em jogos 3D que precisem ser apresentados em dispositivos cujas características inviabilizem a utilização de gráficos tridimensionais (como, por exemplo, o tamanho da tela ou a capacidade de processamento). A posição dos objetos da área de interesse enviada ao jogador será definida, portanto, de maneira bidimensional.

No serviço, para cada estratégia de adaptação, podem ser utilizados quantos módulos lógicos for necessário. Apesar de somente quatro terem sido concebidos, o serviço é extensível para que outros possam ser criados e acoplados. Desta forma, o resultado da transformação realizada por um módulo lógico serve de entrada para o módulo seguinte.

Os módulos lógicos descritos possuem a propriedade de que, não importa a ordem em que eles processam um mesmo conjunto de dados de uma área de interesse, o resultado será sempre o mesmo. Contudo, o custo de processamento pode variar dependendo desta ordem. Por exemplo, caso o módulo de escala bidimensional trate o conteúdo de uma área de interesse antes do módulo de filtragem, será ajustada a posição de elementos que possivelmente serão descartados no módulo seguinte. Desta maneira, percebe-se que é mais vantajoso ordenar os dois módulos para que o de filtragem seja executado antes do de escala.

Duas recomendações, portanto, são feitas a partir destas observações. A primeira é que novos módulos lógicos criados e adicionados ao serviço mantenham a propriedade de garantia do mesmo resultado final independentemente da ordem em que os módulos realizam o processamento. A segunda recomendação é que o funcionamento dos módulos seja compreendido pelo desenvolvedor do jogo para ordená-los da melhor maneira, se quiser utilizá-los de maneira otimizada.

Por sua vez, em relação aos processamentos de apresentação, foram definidos os seguintes módulos:

- **Módulo de apresentação textual:** baseado no idioma escolhido pelo jogador, este módulo cria um texto informando resumidamente o conteúdo da área de interesse. O idioma é armazenado como uma propriedade do dispositivo, obtido a partir do seu contexto. Desta forma, são informados quantos atores, itens e personagens automatizados estão presentes, sendo também incluída a representação textual de cada um deles. É permitido ao desenvolvedor do jogo escolher como é criada a representação textual de cada elemento, através das classes que definem cada elemento do jogo;
- **Módulo de apresentação padrão:** neste módulo todas as informações resultantes da área de interesse

são enviadas de uma maneira padronizada, de acordo com um protocolo definido pelo desenvolvedor do jogo. Esta forma de apresentação sugere como o estado do jogo é transmitido para jogadores em computadores pessoais.

Cada módulo de apresentação define inteiramente a forma com que o conteúdo da área de interesse é enviado ao jogador. Assim, faz sentido que somente um módulo de apresentação seja utilizado em cada estratégia de adaptação. De forma semelhante aos módulos lógicos, novas formas de apresentação podem ser incluídas no serviço, seguindo a especificação definida pela interface **PresentationModule**.

Para explicar melhor o funcionamento do serviço, o fluxo de mensagens da operação **adaptContent** ocorre de acordo com a Figura 3. Ela possui dois parâmetros: a área de interesse do jogador e o seu contexto. Logo que a operação é chamada a partir da simulação servidora do jogo, o serviço delega a lógica da operação para o **AdaptationStrategyManager**, entidade responsável pelo armazenamento de todas as estratégias de adaptação previamente definidas. Este verifica qual a estratégia de adaptação que será utilizada, baseando-se no contexto do jogador, mais especificamente no dispositivo utilizado. Para cada módulo lógico da estratégia de adaptação (se houver algum), a área de interesse sofre um processamento até ser enviada ao módulo de apresentação da estratégia. O resultado é, então, retornado à simulação do jogo, que o repassa a aplicação cliente.

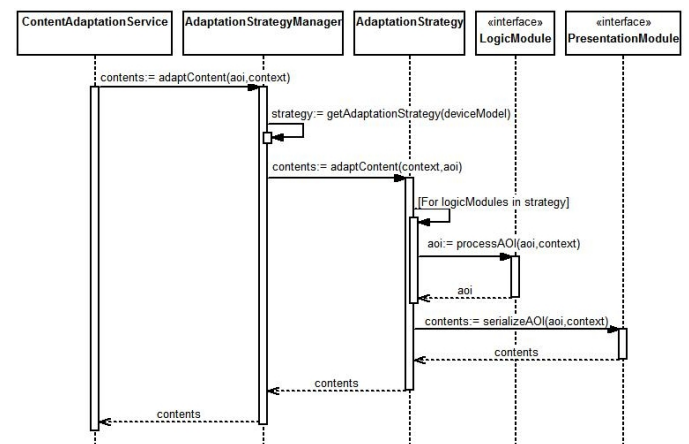


Figura 3: Diagrama de seqüência do Serviço de Adaptação de Conteúdo

A operação **getStrategy** pode ser utilizada pela simulação de jogo para realizar a adaptação de conteúdo de jogadores estacionários, cuja taxa de atualização do estado do jogo é muito freqüente. Desta forma, a simulação obtém a estratégia de adaptação do jogador e a armazena em uma *cache*, fazendo com que ela realize a adaptação localmente da mesma maneira que é feita no serviço. O fluxo de mensagens é mostrado na Figura 4.

É importante ainda observar que o caráter genérico do serviço exige que as estratégias de adaptação a serem utilizadas pelo jogo devem ser previamente definidas pela simulação do jogo. Deste modo, na inicialização do servidor

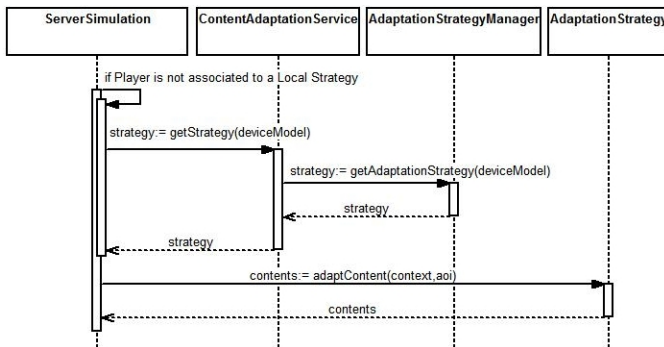


Figura 4: Diagrama de seqüência da operação `getStrategy` do Serviço de Adaptação de Conteúdo

do jogo, as estratégias são criadas e repassadas ao serviço.

3.2 Serviço de Adaptação de Interação

O uso de diferentes dispositivos permite que as entradas das ações dos jogadores móveis e estacionários sejam realizadas de formas diferentes. Porém, independentemente do contexto de execução do jogador, estas diferentes ações devem ser mapeadas para ações comuns dentro do mundo virtual do jogo. O jogo multiplataforma deve permitir que jogadores estacionários e móveis interajam diretamente no mundo virtual realizando ações equivalentes.

O Serviço de Adaptação de Interação propõe-se a transformar as diferentes formas de entradas dos jogadores em ações padronizadas dentro de cada área de interesse do jogo. Em um contexto estacionário a atuação do serviço pode ser dispensável, uma vez que o jogador estará efetivamente controlando todas as ações do seu personagem. No contexto móvel, por sua vez, a adaptação de interação é imprescindível, visto que a limitação da interface homem-máquina dos dispositivos móveis faz com que as típicas ações baseadas em movimentações ágeis do avatar sejam inadequadas neste contexto. Sendo assim, as ações de um jogador móvel devem ser realizadas sem a necessidade de uma conexão constante com o jogo ou de respostas imediatas aos seus comandos, podendo o jogador disparar uma ação a ser executada no mundo virtual enquanto ele realiza outras atividades fora do jogo, como deslocamento físico ou atendimento de chamadas no aparelho celular.

No Serviço de Adaptação de Interação, estes requisitos são acatados através da idéia proposta por [3]. As ações de jogadores móveis são realizadas no mundo virtual através de comandos de alto nível, que são interpretadas pelo jogo em um conjunto de ações concretas. Por comandos alto nível, entenda-se que a ação a ser realizada por personagem de um jogador pode ser determinada por comandos como: “Fique em posição de defesa”, “Colete objeto” ou “Ataque adversário”. Não seria necessário que o jogador controlasse efetivamente seu personagem durante toda a execução da ação. O comportamento do personagem após o comando executado seria semelhante a um NPC, controlado por uma inteligência artificial.

3.2.1 Modelagem

Para modelar este serviço, foi necessário definir como seria a interação do usuário com o jogo em um dispositivo móvel de interface homem-máquina limitada. Uma solução encontrada seria fazer com que o jogo enviasse ao jogador o conjunto de ações de alto nível possíveis do seu personagem. O jogador, então, informaria ao servidor do jogo a ação escolhida e este se responsabilizaria - utilizando o Serviço de Adaptação de Interação - a mapear o comando em ações padrões do jogo.

Ações do jogador móvel - Para realizar a obtenção das ações possíveis de um avatar em um dado instante, convém definir quais aspectos do jogo influenciam este procedimento. Em primeiro lugar, foi definido que o estado do avatar é um ponto de partida para descobrir quais ações ele pode realizar. Cada ação pode ser interpretada como uma transição entre estados, sendo indicado o comportamento que o personagem deverá assumir a partir daquele momento.

Entretanto, é importante notar que não apenas o estado do jogador determina suas ações possíveis em um dado momento do jogo. Dependendo do conteúdo da área de interesse, determinadas ações podem ser realizadas ou não. Por exemplo, um jogador não poderá realizar nenhuma ação de interação com itens virtuais caso não haja nenhum item virtual visível na sua área de interesse. Do mesmo modo, ele também não poderá interagir com outro personagem caso ele seja o único personagem em sua área de interesse naquele momento. Assim, foi definido que o conteúdo da área de interesse também influencia na obtenção das ações do jogador.

Com isto em mente, foi criada o conceito de *ação*, que representa as ações de alto nível dos jogadores. Esta entidade possui uma descrição textual, usada na exibição das ações para os jogadores, além de uma propriedade que informa se a ação está relacionada ao conteúdo da área de interesse do jogador. Esta informação é importante para permitir que algumas ações possam ocorrer sempre, independentemente do conteúdo da área de interesse do jogador. Um exemplo é uma ação para se desconectar do jogo, que a princípio pode ser realizada a qualquer momento.

Uma ação está sempre relacionada a um evento de jogo. Esta relação determina de maneira simples como o jogo irá interpretar a ação, pois um *Evento* é a forma padrão com que a simulação de jogo trata os eventos recebidos dos jogadores.

Uma ação, como já foi dito, pode estar intimamente relacionada a um elemento específico da área de interesse do jogador. Pode ser, por exemplo, um outro personagem com o qual deseja-se interagir, ou um item qualquer. Deste modo, deve ser possível ao jogador, no momento em que seleciona uma ação deste tipo, especificar o objeto do jogo ao qual a ação se refere. Isto é possível com o uso de uma lógica do serviço que, ao oferecer uma ação a um jogador, liste opções relativas àquela ação. Por causa disto, a cada ação é atribuído um conjunto de objetos, cada um representando uma opção daquela ação que pode ser escolhida pelo jogador.

Para saber se um objeto da área de interesse será considerado uma opção de uma determinada ação, foi criada uma operação, que deve determinar para cada ação do jogo se

o objeto se relaciona ou não com ela. Vale ressaltar que o serviço, por ser genérico, não tem como identificar que ações podem ser realizadas para cada estado de um avatar. Na realidade, o jogo é que define quais são os seus estados e ações. Isto ocorre na inicialização do servidor do jogo, quando este mapeamento é definido e informado ao serviço.

Cada avatar possui um conjunto de estados que o mesmo pode assumir em dado instante. Definiu-se que cada estado está diretamente relacionado a um nível de abstração do contexto de execução. Cada dispositivo possui um nível de abstração que além de ser utilizado pelo Serviço da Adaptação de Conteúdo para determinar a quantidade de informações enviadas ao jogador, passa a indicar o nível de abstração das suas ações.

Existem, na realidade, vários conjuntos de estados para cada contexto que o jogador pode assumir no jogo. Desta forma, o serviço tem condições de saber quais ações o jogador poderá realizar para um dado contexto, baseando-se apenas no estado atual do seu avatar. Caso o jogador esteja em um contexto móvel, seu avatar deverá assumir um entre os possíveis estados relacionados a este contexto. Se ele passar a jogar em um computador pessoal, o estado do seu avatar deverá refletir a mudança de contexto. Este processo também é realizado pelo Serviço de Adaptação de Interação. Ele deve saber, a partir do estado de um avatar e do novo contexto do jogador, qual o novo estado que será assumido.

A Figura 5 exemplifica a utilização de diferentes conjuntos de estados para cada contexto de execução, bem como as associações entre estados de diferentes contextos, necessárias para atualizar o estado do avatar na mudança de contexto do jogador.

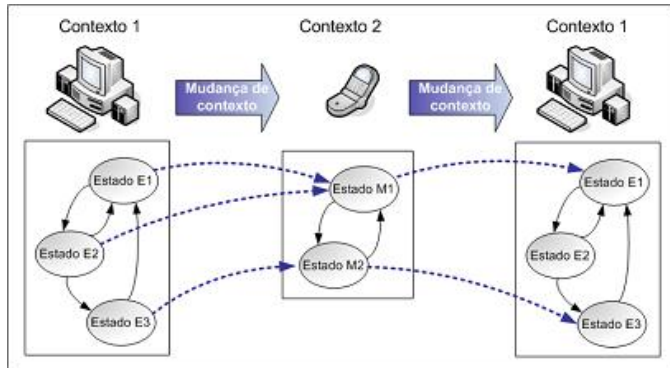


Figura 5: Exemplo de mapeamento entre estados relacionados a diferentes contextos

Devido ao seu caráter genérico, independente do jogo, o serviço deve ser informado pelo jogo como este procedimento deve ser realizado. Portanto, o jogo deve associar cada estado a um certo número de outros estados, para cada contexto diferente do original.

Expiração de ações - Outro mecanismo oferecido pelo serviço é a utilização de um tempo de expiração para as ações enviadas ao jogador. Isto ameniza problemas de inconsistência que podem ocorrer quando um jogador móvel recebe um conjunto de ações e só executa a ação selecio-

nada algum tempo depois. Neste interim, o conteúdo da área de interesse pode ter sido modificado de forma que determinadas ações não possam mais ser realizadas. Através deste recurso, o serviço de adaptação de interação verifica se o prazo de realização da ação requisitada pelo jogador expirou ou não. Em caso positivo, o jogador deverá receber novamente um conjunto atualizado de ações para escolher qual ação tomar. A Figura 6 apresenta a arquitetura proposta para o Serviço de Adaptação de Interação como um diagrama de classes em UML[6].

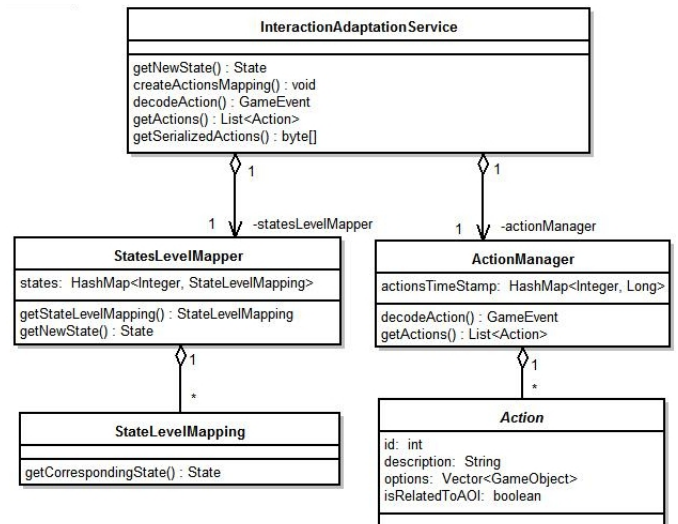


Figura 6: Diagrama de classes do Serviço de Adaptação de Interação

Para uma melhor compreensão das relações entre as classes, são apresentados diagramas de sequência das principais operações fornecidas pelo Serviço. A Figura 7 apresenta os passos realizados para se obter as ações de um jogador em um dado instante do jogo. Com a operação **getActions**, o Serviço de Adaptação de Interação obtém as ações de um jogador móvel. Esta operação é chamada pela simulação do jogo sempre que o jogador requisita as ações. Ela tem dois parâmetros: a área de interesse do jogador e o seu personagem.

Assim que a operação é chamada, o serviço chama a operação de mesmo nome do **ActionManager**, que é a classe responsável por armazenar as ações para cada estado dos personagens de jogadores e também realizar a lógica de obtenção dos estados a partir do conteúdo da área de interesse. Em seguida, é criado um conjunto vazio de ações ao qual serão adicionadas aquelas que satisfizerem determinadas restrições. O **ActionManager** obtém, então, as ações associadas ao estado do avatar para analisar quais delas devem ser enviadas ao jogador.

Para cada uma destas ações, é verificado se ela tem relação com a área de interesse do jogador. Em caso negativo, a ação é imediatamente adicionada ao conjunto resultante de ações, visto que nenhuma verificação adicional a respeito dela precisa ser feita. Este é o caso, como já mencionado, de uma operação de se desconectar do jogo, por exemplo. Contudo, caso a ação seja relacionada a uma área de interesse, deve-

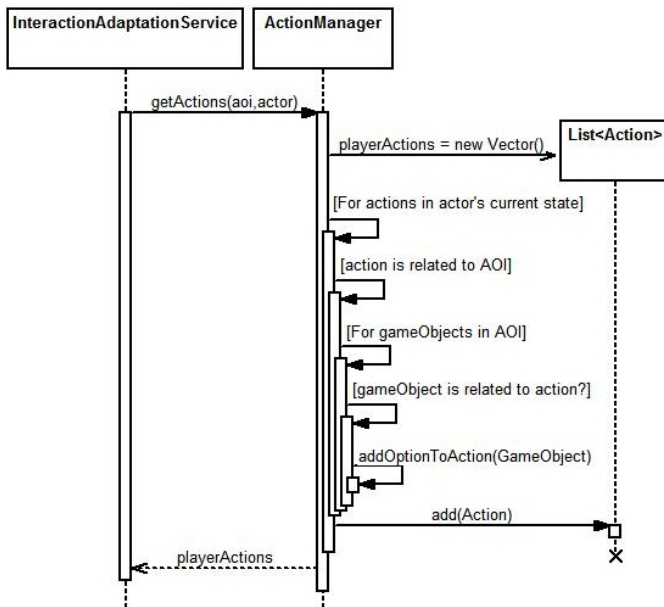


Figura 7: Diagrama de seqüência - Obter ações de um jogador pelo Serviço de Adaptação de Interação

se verificar se a área de interesse atual possui os elementos necessários para que a ação possa ser realizada. Isto é feito verificando-se a relação de cada objeto presente na área de interesse com aquela ação. Caso o objeto não tenha relação com a ação, é ignorado. Caso contrário, o objeto será incluído como uma opção daquela ação. Se houver possuir pelo menos uma opção relativa a uma ação, já é suficiente para indicar que a ação poderá ser realizada naquela área de interesse, e portanto, deve ser incluída no conjunto possível de ações de jogador. Caso contrário, ela será descartada no momento e não será enviada ao jogador.

A Figura 8 representa o processo de envio de uma ação por um jogador para a simulação do jogo. A operação `decodeAction` do Serviço de Adaptação de Interação é responsável por mapear uma ação do jogador móvel em ações padrões do jogo. Esta operação resulta em um evento do jogo. Seus parâmetros são informações do jogador obtidas pela simulação do jogo quando a ação é enviada, e os dados específicos da ação. Tais dados indicam a ação escolhida e qual opção foi selecionada, caso seja necessário. Assim que a operação é chamada pela simulação do jogo, o serviço chama a operação `decodeAction` de `ActionManager` para decodificar a ação e obter o evento de jogo relacionado. O `ActionManager`, por sua vez, apenas cria um novo `GameEventDefault` - implementação padrão da classe abstrata `GameEvent`, que é o evento padrão de comunicação entre o servidor do jogo e clientes estacionários - com os dados do jogador, da ação e da opção selecionadas pelo jogador. Este evento é então retornado à simulação de jogo, que o processa normalmente como um evento de jogo qualquer.

Por meio da operação `getNewState` (Figura 9) do Serviço de Adaptação de Interação, a simulação do jogo é capaz de obter o novo estado do avatar de um jogador quando ele muda de contexto de execução. Para tanto, precisa informar

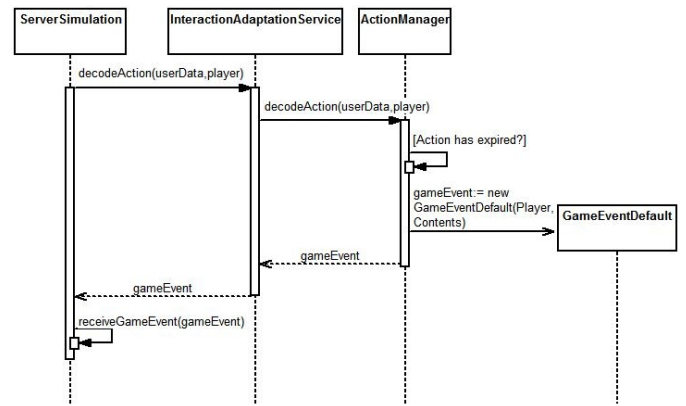


Figura 8: Diagrama de seqüência - Enviar ação de um jogador pelo Serviço de Adaptação de Interação

o estado atual do avatar e o novo contexto assumido pelo jogador.

Logo que a operação é chamada pela simulação do jogo, é delegada a lógica da operação para um objeto `StatesLevelMapper`, que possui todos os mapeamentos entre cada estado e os estados de outro contexto que lhe são correspondentes. É obtido, então, o objeto `StateLevelMapping` relativo ao estado atual, que contém todos os estados correspondentes a este estado. Finalmente, sua operação `getCorrespondingState` permite obter o estado desejado passando como parâmetro o novo contexto do jogador.

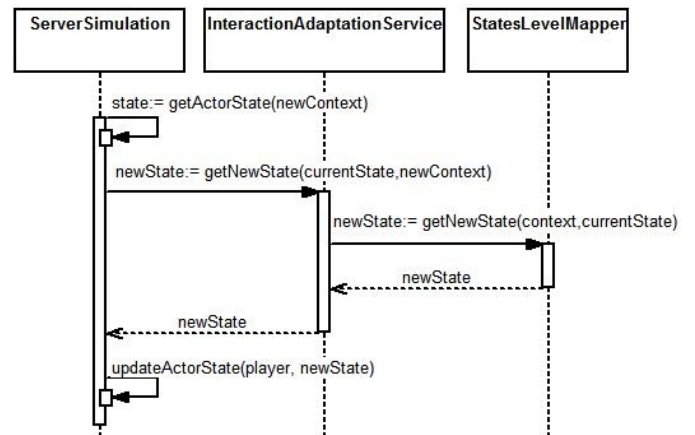


Figura 9: Diagrama de seqüência da operação `getNewState` do Serviço de Adaptação de Interação

4. CENÁRIOS IMPLEMENTADOS

Ao longo da implementação dos dois serviços tratados na seção anterior, foram criados cenários de jogos para validá-los. Tendo sido implementados tanto para computadores pessoais, para dispositivos móveis com interface gráfica e telefones celulares de interface textual, garantem a correção e eficiência das operações realizadas por ambos os serviços em diferentes contextos de execução.

4.1 Implementação

Para implementar os cenários foi preciso realizar algumas etapas, também necessárias à criação de qualquer jogo a ser utilizado sobre o middleware PM2G. A primeira delas foi a criação de entidades do jogo a partir das entidades abstratas já definidas no *framework* PM2G, referentes aos objetos do jogo (tanto itens virtuais como os personagens), aos estados que cada avatar pode assumir e também às ações definidas para os jogadores móveis. Além destas classes, foram criadas classes concretas de infra-estrutura necessárias para a execução do jogo no servidor e no cliente, e também classes referentes a outros serviços.

Outro passo foi configurar os serviços na inicialização do jogo, para que pudessem ser utilizados ao longo da simulação. Isto é necessário porque, embora concebidos para serem o mais genérico possível, tanto o Serviço de Adaptação de Conteúdo como o Serviço de Adaptação de Interação precisam de informações específicas do jogo para seu correto funcionamento. O primeiro precisa ter armazenadas todas as estratégias de adaptação que serão utilizadas no jogo, baseado nos dispositivos para o qual é projetado. Já para o Serviço de Adaptação de Interação é necessário que o jogo forneça duas informações: uma associação entre estados correspondentes para diferentes níveis de abstração e um mapeamento entre os estados e as possíveis ações que o avatar pode realizar a partir deles.

Finalmente, para permitir a utilização dos cenários, foram criadas operações básicas para o jogador que lhe permitem conectar-se e desconectar-se do jogo, bem como definir seu contexto de execução.

4.2 Descrição dos cenários

Os cenários baseiam-se em situações típicas de um jogo massivo, no estilo RPG (do inglês, *Role Playing Game*) e incluem interações do jogador com avatares de outros jogadores e com itens no mundo do jogo. São eles:

- **Obter estado do jogo:** informa ao jogador o conteúdo de sua área de interesse, em uma periodicidade e nível de detalhamento que depende do seu contexto. Jogadores estacionários são atualizados constantemente com estas informações, enquanto jogadores móveis têm que requisitar o estado do jogo sempre que desejarem obter informações da sua área de interesse. A quantidade de informações recebidas varia para cada contexto de execução dos jogadores. Os estacionários recebem todas as informações possíveis, sejam elas a respeito dos personagens dos outros jogadores, dos NPCs ou dos itens virtuais. Jogadores móveis, por sua vez, não recebem informações acerca dos NPCs. Além disso, para cada contexto móvel o nível de detalhamento das informações é diferente: jogadores que utilizam dispositivos móveis com interface gráfica têm uma noção da posição de cada personagem na área de interesse, enquanto jogadores com aparelhos celulares de interface textual não têm este tipo de informação. A Figura 10 mostra as visões de jogadores em cada um dos contextos para um mesmo instante do jogo.
- **Movimentação do avatar:** permite ao jogador mover seu avatar livremente pelo mundo virtual, dentro

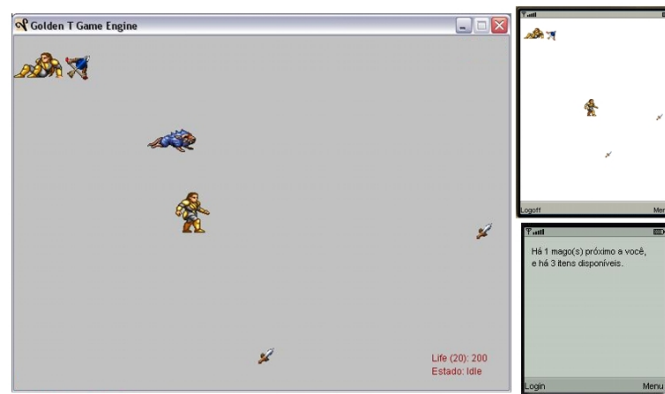


Figura 10: Diferentes visões de um estado do jogo, de acordo com o dispositivo do jogador

de sua área de interesse e também entre diferentes áreas de interesse. Isto pode ser utilizado para um objetivo específico, como um ataque ou coleta de item virtual, mas pode também ser realizado apenas com o propósito de locomover o personagem. Neste caso, a simples movimentação do avatar foi definida como uma interação de baixo nível, pois exige do jogador um controle minucioso da posição atual do seu personagem. Desta forma, foi implementado apenas para o contexto de computadores pessoais. O jogador interage utilizando o *mouse*, clicando na posição do mundo virtual para a qual deseja que seu personagem se dirija.

- **Atacar adversário:** este cenário é um exemplo de interação entre dois jogadores, na qual seus dois avatares lutam entre si. Para o jogador estacionário, não existe um comando explícito para atacar; ao invés disso, o jogador tem que controlar seu personagem e guiá-lo até o adversário. Já o jogador móvel, por não controlar diretamente a movimentação do seu personagem, simplesmente realiza um comando de ataque especificando o adversário. Seu avatar, então, se locomoverá de maneira automática até o adversário e os dois começarão a lutar entre si.
- **Coletar item virtual:** de forma semelhante ao ataque a um adversário, a coleta de um item virtual por um jogador estacionário ocorre quando ele movimenta seu personagem até o item desejado. Já o jogador móvel dispõe de um comando explícito, caso haja itens virtuais visíveis na sua área de interesse. Quando a ação é enviada ao servidor, a simulação do jogo movimenta o personagem até o item virtual, até que haja uma colisão e o item seja coletado. A Figura 11 apresenta uma visão das ações de coleta para um jogador móvel.

5. CONCLUSÕES E TRABALHOS FUTUROS

A integração de jogos multiusuário, mobilidade e heterogeneidade de dispositivos apresenta-se como uma tendência para jogos digitais futuros. Embora trabalhos apontem as vantagens desta integração, estes falham em não apresentar um modelo claro e concreto que dê suporte a aplicações

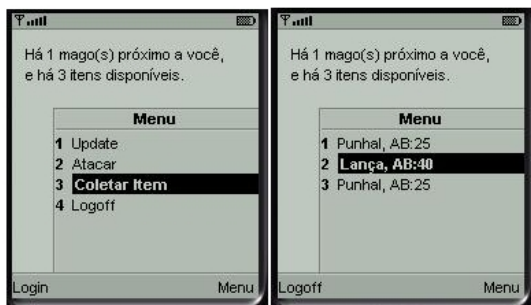


Figura 11: Realização de um ação via um dispositivo móvel

que busquem alcançar tal integração. A proposta PM2G é uma visão precisa e original da integração de diversas tendências, através de modelos que ditam como estas coexistem. Ressaltamos também que são incorporadas funcionalidades até então inéditas, como a interação direta entre jogadores móveis, cujos resultados podem ser integrados ao contexto do jogo multiusuário.

Neste documento, apresentamos a especificação de dois dos principais serviços integrantes da proposta PM2G. Através da implementação de um protótipo e da elaboração de cenários comuns a um modelo de jogos que se pretende oferecer suporte, nos permitiu validar nossas idéias. O projeto PM2G continua em evolução, onde pretendemos continuar a implementação e testes dos demais serviços, promovendo uma contínua integração entre os mesmos. Espera-se ao final oferecer aos projetistas de jogos um ambiente que desperte a criação de novos e criativos jogos à comunidade de usuários. Em verdade, os serviços propostos podem ser aproveitados como base para outros domínios de aplicações, como medicina ou educação.

6. REFERÊNCIAS

- [1] F. R. Cecin. FreeMMG: uma Arquitetura Cliente-Servidor e Par-a-Par de Suporte a Jogos Maciçamente Distribuídos. Master's thesis, Instituto de Informática. Universidade Federal do Rio Grande do Sul, Janeiro 2005.
- [2] B. D. Chen and M. Maheswaran. A cheat controlled protocol for centralized online multiplayer games. In *Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*, pages 139–143. ACM Press, 2004.
- [3] D. Fox. *Massively Multiplayer Game Development*, volume 1, chapter Small Portals: Tapping into MMP Worlds via Wireless Devices, pages 244 – 254. Charles River Media, Inc., Rockland, MA, USA, 2003.
- [4] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer Support for Massively Multiplayer Games. In *Proceedings of the 23rd Conference of the IEEE Communications Society*. IEEE Computer Society, July 2004.
- [5] E. M. Koivisto and C. Wenninger. Enhancing player experience in MMORPGs with mobile features. In *2nd International Digital Games Research Association Conference*, 2005.
- [6] Object Management Group. Unified Modelling Language, 2006. Disponível em <http://www.uml.org/>.
- [7] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17, aug 2001.
- [8] J. Smed, T. Kaukoranta, and H. Hakonen. Aspects of Networking in Multiplayer Computer Games. In L. W. Sing, W. H. Man, and W. Wai, editors, *Proceedings of International Conference on Application and Development of Computer Games in the 21st Century*, pages 74–81, Hong Kong SAR, China, Nov 2001.
- [9] Sun Microsystems. Java 2 Platform, Micro Edition, 2006. Disponível em <http://java.sun.com/j2me/>.
- [10] F. Trinta, C. Ferraz, and G. Ramalho. Middleware Services for Pervasive Multiplatform Networked Games. In *Proceedings of ACM Workshops on Network and System Support for Games'2006*. ACM Press, 2006.
- [11] F. Trinta, C. Ferraz, and G. Ramalho. Uma Proposta de Cenários e Serviços de Suporte para Jogos Multiusuário Multiplataforma Pervasivos. In *Anais do XII Simpósio Brasileiro de Sistemas Multimídia e WEB*, 2006.
- [12] G. Walton. Nine things to look for in the next generation of MMOG, 2004. Keynote presentation, 2nd International Digital Games Research Association Conference. Disponível em http://www.gdconf.com/archives/2004/walton_gordon.ppt.