

De Volta para o Futuro: Uma Técnica para Sincronizar Vistas de Jogos Multi-Jogador para Celular via Bluetooth

Lauro E. Kozovits Alexandre Sztajnberg Esteban G. Clua*

Universidade Estadual do Rio de Janeiro, DICC-IME, Brazil
Fluminense, IC-UFF, Brazil

*Universidade Federal

Abstract

The scene view synchronization in different workstations both in simulations and in multiplayer games is a research topic of great relevance in both areas. This work presents an alternative and yet simple synchronization technique for Bluetooth connected mobile multiplayer games.

Keywords: mobile games, multiplayer games, bluetooth games, dead reckoning

Authors' contact:

{lauro, alexszt}@ime.uerj.br
**esteban@ic.uff.br

1. Introdução

Jogos multi-jogador constituem uma sub-área de ambientes virtuais ligados por rede (net-VEs) que têm despertado um grande interesse de pesquisas tanto por parte da indústria quanto pela comunidade científica. Trabalhos relativos a otimização e tratamento conveniente de mensagens trocadas nestes ambientes demonstram a relevância da pesquisa nesta área [Kozovits 2004].

A interatividade e a sincronização da visualização de cenas por parte de cada participante em jogos multi-jogador e simulações é um tópico de pesquisa tanto nas áreas de simulação quanto de jogos. Em um jogo multi-jogador torna-se particularmente importante que um momento de pontuação seja claramente acusável por todos os participantes ou o jogo perderá a credibilidade e o interesse.

O problema crítico de consistência visual num jogo multi-jogador ocorre quando a cena vista por um jogador é diferente daquela vista por outro (ou outros), jogador (jogadores). Por exemplo, é possível que uma bola ultrapasse um goleiro em uma workstation enquanto em outra a bola ainda está chegando de modo que o defensor ainda tenha a chance de capturá-la.

O problema relatado é freqüente em função do fato de que há atrasos no envio de mensagens entre pontos de uma rede seja ela Internet, intranet ou mesmo Bluetooth como tratada no presente trabalho.

Jogos multi-jogador para celular fazem parte de um mercado em franca expansão [Kozovits 2008] e podem

apresentar características especiais, eventualmente demandando soluções um pouco diferentes daquelas usualmente adotadas para ambientes PC, devido à diferença capacidade de processamento de suas CPUs, memória disponível para os programas, etc.

Neste trabalho apresentamos uma técnica simples que possibilita alcançar uma maior consistência visual entre dispositivos executando jogos multi-jogador para celular via Bluetooth. Na Seção 2 apresentamos os trabalhos relacionados. Na Seção 3 a técnica proposta é apresentada. Na Seção 4 seus resultados são avaliados e na Seção 5 a conclusão é descrita.

2. Trabalhos Relacionados

Em ambientes PC e em workstations em geral é comum o uso da técnica de dead-reckoning [IEEE Standard 1278, 1993], [Singhal e Cheriton] e [Singhal e Zyda] para possibilitar a previsão de trajetória de forma mais suave de objetos em movimento evitando-se saltos ou descontinuidades entre uma mensagem de atualização e outra, bem como evitar o envio de um número elevado dessas mensagens por unidade de tempo.

Por exemplo, se um avião voando durante uma simulação fizer um desvio de sua trajetória e uma ou mais mensagens forem perdidas é possível que numa workstation ele apareça num vôo direto e em outra ele seja mostrado com a trajetória contemplando o desvio (figura 1a).

O algoritmo de dead-reckoning calcula uma trajetória suave (T2) até o ponto de convergência correspondente à mensagem P3 na figura. Assim, equações não lineares são geralmente empregadas para ajustar as coordenadas recebidas, resultando num visual mais crível e, portanto, mais compatível com um vôo real.

Entretanto, se por um lado o uso de dead-reckoning implica numa taxa menor de mensagens enviadas numa simulação ou jogo se houver um obstáculo na trajetória T2 há um problema evidente para a simulação, pois uma colisão pode ser mostrada numa workstation, mas não em outra ou mesmo uma situação irreal de colisão sem dano (problema de superposição) pode ocorrer comprometendo a qualidade do jogo ou da simulação.

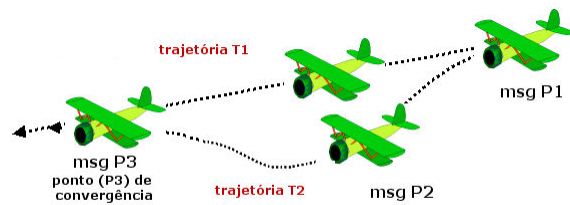


Figura 1a: Não-consistência visual entre duas trajetórias mostradas em duas workstations (T1 e T2) em função da perda de mensagem de posição P2. A chegada da mensagem P3 e o algoritmo de dead-reckoning permitem uma trajetória de compensação suave.

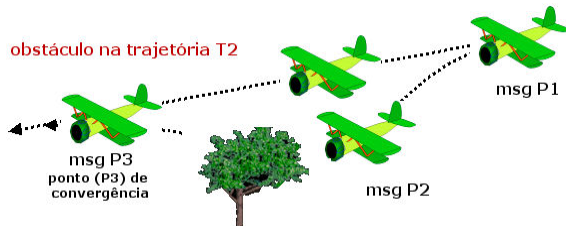


Figura 1b: Se houver um obstáculo na trajetória T2 um problema pode ser criado para a simulação.

O uso de técnicas de dead-reckoning é um padrão em jogos de ação via Internet ou mesmo em simulações militares [UNITED STATES DEPARTMENT OF DEFENCE 2002].

Uma outra abordagem é o uso de agentes inteligentes que tomam decisões baseados em metas a serem alcançadas [Szwarcman et. al. 2000]. Neste trabalho não há, na prática, uma consistência visual a cada fotograma, mas sim de resultados finais.

Por exemplo, de acordo com o que ilustra a figura 2, se um personagem deve sair de uma sala A em primeiro plano e efetivamente chegar a uma sala B ao fundo (esta é a meta) então não faz diferença neste tipo de simulação qual o caminho tomado para se chegar àquela situação final (personagem na sala B). Se houver duas portas (digamos uma à direita e outra à esquerda) que possam conduzir o personagem da sala A até a sala B é possível que o personagem orientado a metas possa, na workstation AZUL (lado esquerdo da figura 2), escolher sair pela porta da esquerda.

Na workstation ROXA (lado direito da figura 2), em função de um atraso na rede, um objeto em movimento (por ex. um outro personagem) pode estar momentaneamente obstruindo a trajetória para a porta da esquerda nesta workstation ROXA (o que não ocorre na workstation AZUL). O tipo de simulação (baseada em agentes reativos orientados a metas) conduzida na workstation ROXA irá, naquele momento, e, em função daquela sua situação peculiar, optar por chegar ao mesmo destino final (sala B) saindo entretanto pela porta da direita.

Portanto, apenas a situação final é visualmente consistente: nas workstations AZUL e ROXA o personagem foi da sala A para a sala B. Situações intermediárias visualmente inconsistentes são irrelevantes para o objetivo da simulação como um todo. É importante observar que isto pode ser aplicável para um dado jogo de estratégia por exemplo, mas nem sempre é aplicável a um jogo de ação ou arcade para celular.

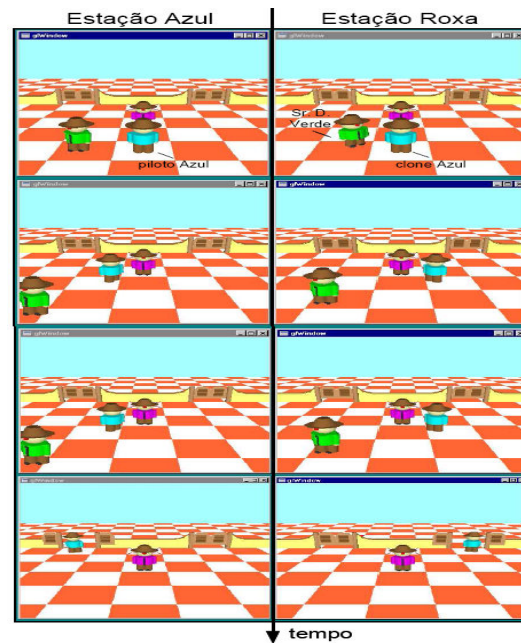


Figura 2: ilustração do mecanismo de personagens reativos orientados por metas extraído de [Szwarcman et. al. 2000]

3. A Técnica Proposta

Considere-se um jogo multi-jogador [Showball 2008], como na Figura 3a, em que dois celulares, um atuando como servidor e outro como cliente utilizam a interface Bluetooth para comunicação. Observa-se que, é necessário um ajuste para compatibilizar o visual de objetos em movimento rápido visto na tela de dois celulares.

Nossa abordagem explora a constância no atraso de uma rede Bluetooth. Além disso, não se observou nenhum problema de *lag* que são comuns em redes não locais como a Internet por exemplo. Assim, torna-se possível estimar a posição de um objeto em movimento tal como será visto pelo outro ponto.

Considere-se, por exemplo, que no início de uma interação o servidor gera a coordenada do objeto em movimento, digamos $(X_{t=initial}, Y_{t=initial})$ e a envia para o cliente. No momento em que o cliente recebe esta coordenada e a exibe, o servidor já estaria mostrando o objeto numa coordenada futura $(X_{t=futuro}, Y_{t=futuro})$ gerando uma discrepância entre os dois visuais.

Isto torna-se visualmente mais relevante quanto maior for a velocidade do objeto. Na Figura 3a isto é ilustrado através de dois celulares, lado a lado, conectados por Bluetooth. Observa-se que a posição da bola no servidor (celular da esquerda) está ligeiramente adiantada em relação ao cliente (celular da direita).

Observe ainda que o jogo Showball foi feito de modo que um jogador possa se posicionar de frente um para o outro. A barrinha inferior em preto representa o jogador local enquanto a barrinha em vermelho representa o jogador remoto.

Na técnica proposta, o cliente retransmite para o servidor a coordenada que ele (cliente) recebeu do próprio servidor (Figura 3b). O servidor irá então recebê-la num tempo $t=atual$ quando o objeto (no servidor) já deveria estar na posição $(X_{t=atual}, Y_{t=atual})$.

Assumindo-se que o atraso na rede bluetooth seja semelhante nos dois sentidos, torna-se possível esperar que a informação de posição vista pelo cliente seja aproximadamente o valor médio entre a posição atual calculada pelo servidor $(X_{t=atual}, Y_{t=atual})$ e a posição informada pelo cliente $(X_{t=inicial}, Y_{t=inicial})$.

No tempo $t=atual$, o servidor irá enviar como nova mensagem de atualização para o cliente a posição atual, mas irá desenhar o objeto numa posição média numa tentativa de compensar os atrasos de transmissão nos dois sentidos e reproduzir o que é desenhado no cliente no mesmo momento:

$$(X_{\text{exibido servidor}}, Y_{\text{exibido servidor}}) = [(X_{t=atual}, Y_{t=atual}) + (X_{t=inicial}, Y_{t=inicial})] / 2.$$

A posição simulada pelo servidor é a posição futura que será desenhada pelo cliente. Entretanto, este mesmo servidor irá, para efeito apenas de visualização, retornar a uma posição anterior. Assim, usamos a expressão “de volta para o futuro” para denominar esta técnica. Na figuras 3a mostramos a posição simulada na cor preta, na figura 3b a posição anterior (a que retorna) em vermelho e, finalmente, em 3c a média das posições (em verde). Esta média corresponde, aproximadamente àquela que é vista pelo cliente (lado direito da figura) gerando uma maior consistência visual.



Figura 3a: jogo Showball executando no simulador e exibindo a bola em movimento (em preto) gerada no celular Bluetooth executando como servidor (celular da esquerda). A

posição com ligeiro atraso de transmissão (em verde) é mostrada no celular cliente Bluetooth à direita.



Figura 3b: jogo ShowBall executando no simulador e exibindo a bola em movimento (em vermelho) no servidor tal como informada pelo cliente.

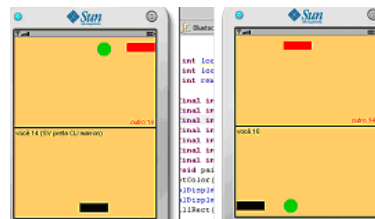


Figura 3c: jogo ShowBall executando no simulador e exibindo a bola em movimento (em verde) no servidor (à esquerda) em posição semelhante aquela que é vista no cliente (à direita) compensando, aproximadamente, os atrasos da rede Bluetooth (de ida e de volta) e gerando uma maior consistência visual.

4. Análise de Resultados

Foi implementado o programa Showball [2008] usando-se Java ME para demonstrar a técnica proposta. A técnica apresentada assume que o atraso entre o que é enviado do servidor ao cliente é semelhante aquele no sentido contrário. Isto pode não ser verdade o que poderia resultar numa discrepância entre a posição vista no cliente e a posição estimada no servidor com a técnica proposta.

Outro fato que cabe relatar é que a técnica proposta efetivamente ajusta o visual entre os dois pontos, mas não elimina acelerações ou retardos na velocidade do da bola que foram observados afetando a fluidez de seu movimento em situações esporádicas. Entretanto, o protocolo JSR 82 [2002] não nos permite ter o controle das camadas mais inferiores deste protocolo para investigar o que estava efetivamente ocorrendo nos testes realizados.

Ainda que possa haver variações entre os atrasos foi observado empiricamente que mesmo o uso da média aritmética entre as duas posições produz resultados melhores nos telefones testados do que o seu não uso. Sendo uma média simples, o benefício, na avaliação dos autores supera o seu não uso.

A técnica formal de dead reckoning [Singhal e Cheriton 1994] poderia, eventualmente, trazer melhores resultados ao custo de uma complexidade bem maior do jogo para celular o que pode ser um fator limitante de seu uso.

Cabe finalmente observar que o problema de inconsistência visual em jogos para celulares via Bluetooth é relevante apenas em objetos com velocidades elevadas.

O jogo Showball [2008] permite que se acelere a bola rebatida de modo a evidenciar o problema do atraso tratado deste trabalho. Entretanto, observou-se que para velocidades compatíveis com a resposta humana mantendo-se a jogabilidade do produto as diferenças de posição perceptíveis geradas devido ao atraso são pequenas.

Na figura 4 mostra-se a execução do jogo Showball implementado, evidenciando-se que quanto mais rápido (par superior na figura) é o movimento de um objeto, mais discrepante visualmente é o seu posicionamento entre os celulares participantes do jogo.

No par inferior quando a velocidade é reduzida a coordenada da bola gerada originalmente no servidor (em preto) não apresenta uma diferença tão expressiva em relação a coordenada informada (de volta, em vermelho) pelo cliente e, por último, àquela calculada (em verde) com a técnica proposta.

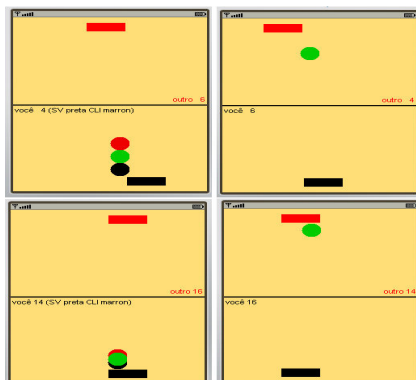


Figura 4: Showball em diferentes velocidades de execução

5. Conclusão

A técnica proposta “de volta para o futuro” é simples, de fácil implementação e eficaz o que é relevante na programação de jogos para celular. Funciona bem para jogos multi-jogador utilizando Bluetooth.

Possivelmente, a implementação de uma técnica formal de dead-reckoning poderia gerar resultados mais acurados. Entretanto, o esforço nesta implementação, além de aumentar o tamanho do código, algo crítico para um jogo para celular, ainda poderia representar uma precisão irrelevante para jogos multi-jogador via Bluetooth.

Trabalhos futuros devem estudar a escalabilidade da técnica proposta em jogos com um número maior de jogadores simultâneos.

A implementação do jogo Showball utilizando a tecnologia proposta está disponível para download e estudo em <http://www.jogos.etc.br/showball>

Agradecimentos

O autor gostaria de agradecer o programa de professores visitantes da UERJ e o apoio dado pelo Instituto de Matemática e Estatística da mesma Instituição IME-UERJ.

Referências

- BLUETOOTH 2008. *Bluetooth Core Specification v2.1* [online]. Disponível em: <http://www.bluetooth.com/Bluetooth/Technology/> [Acessado em 15 de Agosto de 2008].
- IEEE STANDARD 1278 - 1993. IEEE standard for information technology - Protocols for distributed simulation applications: entity information and interaction, IEEE Computer Society
- JSR 82, 2002. Java APIs for Bluetooth. [online]. Disponível em: <http://jcp.org/en/jsr/detail?id=82> . [Acessado em 15 de Agosto de 2008].
- KOZOVITS, L.E., 2004. *Otimização de Mensagens e Balanceamento de Jogos Multi-Jogador*. Tese de doutorado, Instituto de Informática, PUC-Rio University.
- KOZOVITS, L.E. 2008. *Mercado de Jogos para Celulares* [online] Active Tecnologia. Disponível em: <http://www.activetecnologia.com.br/j2me/MercadoJogosCelulares.pdf> [Acessado em 15 de Agosto de 2008].
- SHOLLBALL, 2008. *Showball Game* [online] Disponível em: www.jogos.etc.br/showball [Acessado em 15 de Agosto de 2008].
- SINGHAL, K.S., CHERITON, D., 1994. USING A POSITION HISTORY-BASED PROTOCOL FOR DISTRIBUTED OBJECT VISUALIZATION. TECHNICAL REPORT, DEPT. OF COMPUTER SCIENCE, STANFORD UNIVERSITY.
- SINGHAL, S., ZYDA, M., 1999. NETWORKED VIRTUAL ENVIRONMENTS: DESIGN AND IMPLEMENTATION. BOSTON:ADDISON WESLEY.
- SZWARMAN, D. FEIJÓ, B. AND COSTA, M., 2000. A framework for networked reactive characters, Proceedings of SIBGRAPI 2000, Gramado, Brasil, p. 203-210.
- UNITED STATES DEPARTMENT OF DEFENCE, 2002. Defence Modeling and Simulation Office. High Level Architecture. [online] Disponível em: <https://www.dms.mil/public/transition/hla/>. [Acessado em 6 de Agosto de 2007].