

Ambiente de simulação baseado no jogo Super Bomberman para ensino e pesquisa em Inteligência Artificial

Igor V. de Souza¹, Fábio M. Pereira¹

¹Departamento de Ciências Exatas e Tecnológicas – Universidade Estadual do Sudoeste da Bahia (UESB)
Vitória da Conquista, BA – Brasil

Abstract. *There are several problems involving the process of teaching and learning in the field of Artificial Intelligence, such as those involving the superficial understanding of contents by students or the absence of environments that allow the application of content covered in the classroom. In view of this, in this paper, we propose the creation of programmable game called BomberAgent, in order to present an environment of simulation and tests that can be exploited as a tool for study, analysis and application of techniques in Artificial Intelligence.*

Resumo. *Existem diversos problemas envolvendo o processo de ensino-aprendizagem na área da Inteligência Artificial, como os que envolvem a compreensão superficial dos conteúdos pelos discentes ou a ausência de ambientes que permitam a aplicação de conteúdos vistos em sala de aula. Diante disso, neste trabalho, propomos a criação do jogo programável BomberAgent a fim de apresentar um ambiente de simulação e testes que possa ser explorado como uma ferramenta para estudo, análise e aplicação de técnicas na área de Inteligência Artificial.*

1. Introdução

A Inteligência Artificial (IA) corresponde a uma área da computação que segundo Russell e Norvig (2013) “tenta não só compreender, mas também criar entidades inteligentes”. Tais entidades inteligentes visam simular a execução de tarefas que envolvem o raciocínio e a reflexão que os seres humanos e os animais são capazes de realizar. Entre essas tarefas, estão atividades que para nós, seres humanos, são consideradas triviais e que, até mesmo, lidamos diariamente com muita facilidade. Porém, para os computadores são extremamente complexas e desafiadoras de serem realizadas como, por exemplo, atividades que visam reconhecer faces familiares ou planejar atividades.

Segundo Battistella et. al. (2016) os conteúdos da área de Computação são geralmente extensos e apresentados através de aulas expositivas, o que não é adequado quando se deseja atingir níveis mais elevados de aprendizagem, com a aplicação do conhecimento em situações práticas. Isto geralmente ocorre no processo de ensino e aprendizagem na área de IA: os discentes possuem uma compreensão apenas abstrata dos temas abordados em sala de aula, devido a falta de aplicação do conhecimento em situações práticas. Além disso, a inexistência de um ambiente padronizado no qual os discentes possam focar apenas na implementação de algoritmos, sem se preocupar em implementar a infraestrutura necessária para executá-los, contribui para dificultar o processo de aprendizagem.

Este trabalho é resultado parcial do projeto intitulado *Laboratório Virtual de*

Sistemas Inteligentes (LVSI), cujo objetivo é o desenvolvimento de um ambiente para ensino, competições e experimentos controlados de pesquisa empírica em Inteligência Artificial. Além de disponibilizar material didático para dar suporte às disciplinas ligadas à área da IA, o LVSI deverá prover um ambiente onde discentes poderão desenvolver, testar e comparar algoritmos pra resolução de diferentes tipos de problemas na área.

O desenvolvimento do *BomberAgent* buscou a criação de um sistema que permitisse aos discentes realizar experimentos práticos com os temas abordados em sala de aula, assim como fornecer um ambiente de fácil programação e entendimento que apresente as ferramentas necessárias para que o discente apenas se preocupe com a implementação de seus algoritmos e técnicas de IA. Outro objetivo do seu desenvolvimento foi o de tornar a experiência prática do discente mais agradável e atraente. Tendo isto em vista, buscamos criá-lo nos moldes de um Jogo Sérioso [Machado, 2009]. Desta forma, já que os jogos possuem apetrechos visuais e dinâmicos, através deste ambiente nós conseguimos atender tanto as questões de ensino e aprendizagem quanto as envolvendo incentivo.

2. BomberAgent

O *BomberAgent* é um jogo de código aberto, projetado com a finalidade de prover um ambiente de simulação e testes de algoritmos relacionados às disciplinas de IA, principalmente, as envolvendo Agentes Inteligentes. Para tornar isso possível, este ambiente fornece mecanismos aos desenvolvedores que os permitam criar agentes para competir entre si em uma partida, possibilitando que estes agentes possam ser gradualmente melhorados conforme o jogador adquire novos conhecimentos em sala de aula. O jogo faz parte do projeto LVSI, cujo principal objetivo é a construção de uma arquitetura para o desenvolvimento de simuladores de problemas de IA que envolvam representação do conhecimento e mecanismos de raciocínio baseados em lógica e orientação a objetos.

Este ambiente de simulação, baseia-se nos jogos da franquia *Super Bomberman*¹, de onde deriva seu nome. Deste modo, o design desse ambiente foi feito com o intuito de assemelhar-se ao máximo aos jogos da franquia original, diferenciando-se apenas pelo fato de o jogador ter que programar o personagem em vez de controlá-lo por meio de um *joystick*, mantendo, dessa forma, a interatividade e a dinamicidade dos jogos da franquia durante a simulação; o que permite que os jogadores sejam desafiados de forma dinâmica e inteligente. Afinal, o jogador além de preparar os seus agentes para reconhecer o cenário e suas mecânicas, deve também programá-los para utilizá-los ao seu favor, já que em cada partida podem existir diferentes formas de se alcançar a vitória devido à dinamicidade do ambiente e variedade de itens/possibilidades.

Segundo a taxonomia de Russel e Norvig (2013), o *BomberAgent* é um ambiente de simulação que pode ser classificado como multiagente, acessível, não determinista, dinâmico, não episódico e discreto. Nele, o jogador possui uma visão completa do cenário, onde pode visualizar todos os obstáculos e personagens.

2.1. O Jogo

O objetivo principal do *BomberAgent* é que o jogador consiga programar seu personagem, o *Bomberman*, para sobreviver e eliminar os outros personagens do cenário antes que se esgote o tempo da partida. Para conseguir isto, o personagem pode colocar

1 https://pt.wikipedia.org/wiki/Super_Bomberman

bombas para explodir obstáculos e inimigos e, ainda, movimentar-se para qualquer direção onde não haja obstáculos em seu caminho. Um exemplo de como o jogador visualiza o cenário pode ser visto na Figura 1.



Figura 1. Exemplo do início de uma partida.

A Figura 1 exemplifica o início de uma partida onde o cenário ainda possui todos os seus obstáculos e personagens. Conforme pode ser visto, existem cinco personagens que podem ser programados pelo jogador. Por padrão, cada um desses personagens inicia a simulação em uma localização específica do cenário e possui uma única bomba. As bombas neste jogo são peças fundamentais para o desenrolar da partida porque elas permitem aos personagens explorar o cenário através da destruição de obstáculos e eliminar os adversários. Além disso, elas podem ser utilizadas como obstáculos para impedir a passagem dos personagens, possibilitando que os agentes elaborem vários tipos de jogadas.

Outra característica importante sobre as bombas são suas explosões. Normalmente, uma bomba após ser colocada, leva alguns segundos até explodir naturalmente, o que possibilita ao personagem fugir do alcance da explosão, evitando, assim, morrer. Entretanto, elas também podem ser explodidas antes do tempo, caso a explosão de uma outra bomba a atinja, como pode ser visto na Figura 2. Um detalhe sobre essas explosões é que elas possuem um alcance pré-definido que pode ser aumentado conforme o personagem adquire melhorias durante a partida. Outro detalhe é que essas explosões não são capazes de atravessar nenhuma entidade contida no cenário, com exceção do *Bomberman*, impossibilitando que após colidir com alguma entidade, a chama da explosão continue a propagar na mesma direção.



Figura 2. Explosão das duas bombas alinhadas.

Além das paredes que cercam o cenário e das próprias bombas, existem outras três entidades que impossibilitam a passagem das chamas, são elas: os blocos rachados, que são destrutíveis; os blocos lisos, que são indestrutíveis assim como as paredes do cenário e os itens, que são destruídos após a colisão com as chamas. Estes itens, surgem aleatoriamente no cenário após a destruição das paredes rachadas e funcionam como melhorias para o personagem. Estas melhorias variam de acordo com o tipo de item que o personagem pegar, sendo que dentre os itens disponíveis estão: a bomba, que aumenta o número de bombas que o personagem pode explodir simultaneamente; o chuta bomba, que permite ao personagem chutar as bombas em alguma direção; os patins, que aumentam a velocidade do personagem e a chama, que aumenta o alcance da explosão das bombas. Estes itens podem ser visualizados na Figura 3.



Figura 3. Itens contidos no cenário.

2.2. A Arquitetura

A arquitetura do *BomberAgent* foi projetada a fim de torná-lo bastante intuitivo e flexível, buscando, com isto, facilitar ao usuário a compreensão do sistema e incentivá-lo a participar do desenvolvimento dele, para, assim, construir uma comunidade ativa de desenvolvedores. Pensando nisso, o desenvolvimento do *BomberAgent* foi feito utilizando a linguagem de programação Java que além de ser multiplataforma e orientada a objetos, também possui uma ampla comunidade de usuários, tornando-a uma linguagem de fácil obtenção de material de estudos e informações. Outro ponto importante da arquitetura, é que ela é dividida em três subsistemas, sendo cada um deles responsável por lidar com um tipo de tarefa específica, enviando mensagens aos outros, caso seja necessário, através de suas interfaces. O objetivo dessa arquitetura foi não só facilitar a compreensão do sistema e tornar mais simples a depuração e testes como também criar níveis de encapsulamento que permitam que as atualizações de métodos e

estruturas de um subsistema não afetem diretamente os outros. Os três subsistemas são: o motor, o ambiente e o agente, como pode ser visto no diagrama de classes representado na Figura 4.

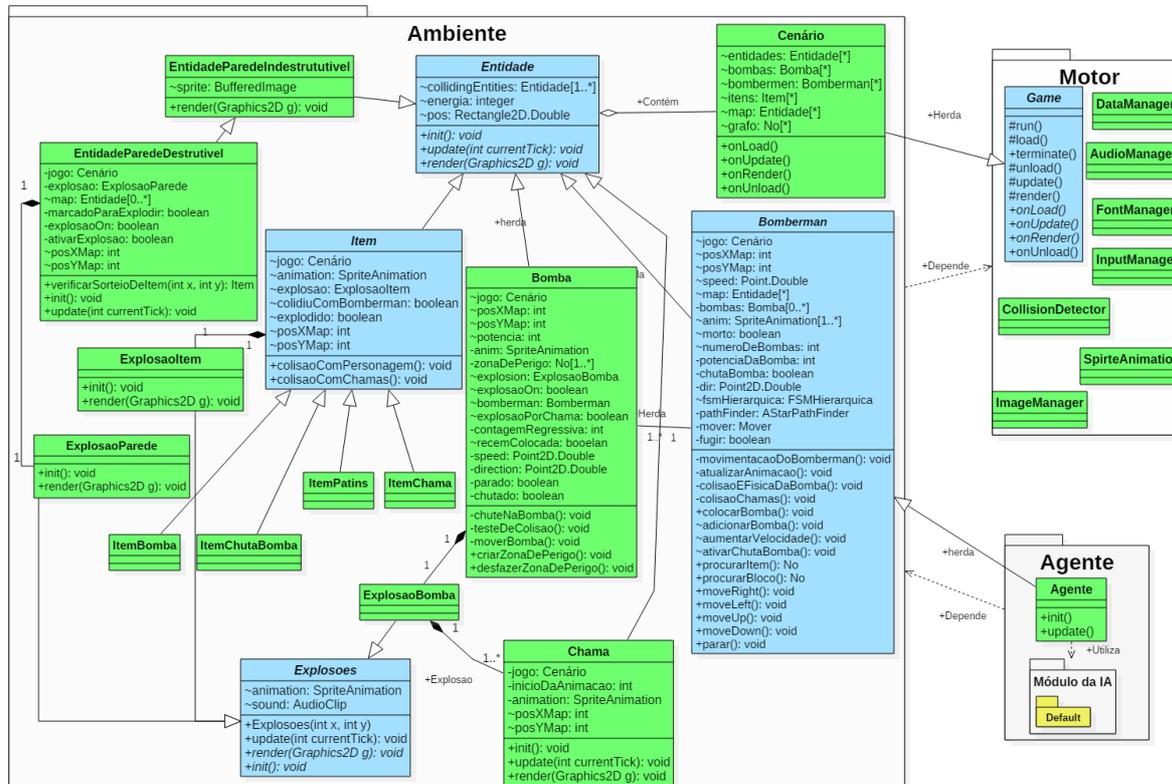


Figura 4. Diagrama de classes do sistema.

O motor é a base do *BomberAgent*. O seu funcionamento assemelha-se aos motores feitos para jogos e, assim como eles, é responsável por realizar tarefas relacionadas ao processamento de gráficos, sons, detecção de colisões entre entidades, animação, gerência de arquivos e de entrada de dados. É a este que os outros subsistemas enviarão uma mensagem caso uma entidade precise ter representada na tela a sua movimentação ou caso algum evento acione algum efeito sonoro. Desta forma, pode-se dizer que o motor é responsável por realizar a ligação entre os eventos que ocorrem no ambiente de simulação, com as interfaces do computador.

O ambiente é o subsistema responsável pela construção da estrutura e da lógica da simulação, é nele que as regras como as condições de vitória ou derrota estão contidas. É também nele que são criadas e definidas as entidades que farão parte da simulação, como, por exemplo, os personagens, os obstáculos, os itens e o próprio cenário. Por isso, é este subsistema que fornece aos jogadores todos os métodos e funções que seus personagens necessitam para interagir com o ambiente de simulação.

Finalmente temos o subsistema do agente que é encarregado de lidar com o mecanismo de raciocínio dos personagens e é através dele que o jogador se comunicará com o ambiente. Agente foi o nome que atribuímos a todas as classes que implementam a classe abstrata *Bomberman*, já que é a partir dela que é feita a interação do personagem com o ambiente. Essas classes ficarão a cargo do discente desenvolver e possuem alguns métodos herdados do subsistema do ambiente que são obrigatoriamente adicionados aos seus escopos. Isto é feito para permitir que o jogador não precise lidar

com o gerenciamento de animações, movimentos e detecção de colisões, tendo, assim, somente que se preocupar com o desenvolvimento do mecanismo de raciocínio do seu personagem.

Assim como foi mostrado na Figura 4, dentro do subsistema do agente existe um pacote chamado de módulo da IA. Este módulo é responsável por guardar as classes e estruturas lógicas utilizadas pelo jogador para construir o mecanismo de representação do conhecimento e de raciocínio do agente, permitindo que o jogador possa desenvolver algoritmos mais arrojados e abrangentes que, devido a sua organização e complexidade, não podem ser desenvolvidos dentro do escopo da classe do agente. Para construção desses mecanismos de raciocínio e representação do conhecimento, o jogador possui uma biblioteca de métodos fornecidos pelo ambiente. Utilizando essa biblioteca, o jogador pode obter, por exemplo, informações sobre a quantidade de inimigos vivos, os objetos contidos no cenário, a quantidade de bombas que o personagem possui, a posição dos personagens no cenário e os métodos responsáveis pelas movimentações e interações dos personagens (ações). Um outro ponto importante, é que além de fornecer essa biblioteca, há dentro do módulo da IA um pacote chamado *Default* que contém um mecanismo de raciocínio já implementado nos agentes, a fim de demonstrar como essa biblioteca pode ser utilizada pelos jogadores.

3. Trabalhos Relacionados

Atualmente, os jogos digitais vão muito além de somente proporcionar diversão e entretenimento aos jogadores. É cada vez mais comum, vermos jogos que simulam situações práticas do dia a dia com o objetivo de proporcionar treinamento de profissionais ou de conscientizar uma comunidade. Nestes tipos de jogos, o entretenimento transforma-se em uma ferramenta para tornar a passagem do conhecimento de uma determinada área mais atraente, pois ao mesmo tempo em que é feita a construção de conceitos também são realizadas estimulações psicomotoras [Machado, 2009]. Isto possibilita que estes jogos, também conhecidos como *Jogos Sérios*, possam ser empregados nas mais diversas áreas do conhecimento com diferentes finalidades, dentre elas a área de IA.

Para a área de IA, os jogos apresentam-se como uma excelente plataforma para teste e validação de novas metodologias e algoritmos, pois possuem a riqueza e a complexidade de ambientes sofisticados [Tatai, 2003]. Afinal, eles proporcionam ambientes controlados que podem simular situações adaptadas do mundo real, tornando possível testar procedimentos que talvez não estejam prontos para um ambiente real. Na área de ensino e pesquisa em IA, por exemplo, já existem diversos jogos sendo aplicados como ambientes de simulação. Dentre eles, alguns dos mais conhecidos são: O Mundo do Wumpus, Robocode e a Robocup.

O Mundo do *Wumpus* [Russel e Norvig, 2013] é um jogo de computador em que um agente deve explorar uma caverna formada por salas conectadas através de passagens. O mapa do jogo é igual a uma matriz, no qual cada quadrado (ou posição da matriz) corresponde a uma sala. O agente sempre inicia no canto inferior esquerdo do mapa e tem como objetivo explorá-lo a procura de uma barra de ouro. Uma vez que ele a encontra, seu objetivo se torna sair da caverna com a barra de ouro em mãos. Para que o agente tenha sucesso em sua missão, ele deve evitar entrar em salas que contenham buracos ou o *Wumpus* (um monstro que devora qualquer um que entre em sua sala) para não morrer.

O Robocode [Nelson & Larsen, 2001], é um jogo de programação cujo objetivo

do jogador é programar um robô tanque para batalhar e competir contra outros robôs em uma arena. As batalhas na arena ocorrem em tempo real e podem ser vistas pelos jogadores, permitindo que eles possam ver o desempenho de seus algoritmos em tempo real. Isto torna o aprendizado mais divertido, pois os jogadores competem entre si através de uma batalha para ver qual algoritmo foi melhor elaborado. Alaiba e Rotaru (2008) apresentam um mecanismo de representação do conhecimento e de raciocínio em lógica utilizando a linguagem Prolog para desenvolvimento de agentes da Robocode.

A *RoboCup* [RoboCup, 2016] é uma iniciativa internacional cuja finalidade é aumentar a pesquisa na área de robótica e IA. Esta iniciativa oferece uma plataforma para pesquisas em desenvolvimento de robôs virtuais para simulações de futebol e resgate. A simulação de futebol é conhecida como *Soccer Server* e, assim como o esporte de verdade, se trata de um ambiente multiagente e em tempo real em que para alcançar a vitória, um time deve fazer o máximo de gols que puder no time adversário, ao mesmo tempo em que tenta levar o mínimo possível. Para isto, os membros de cada equipe precisam se comportar de maneira rápida, flexível e cooperativamente, levando em consideração situações locais e globais, tornando-se, dessa forma, uma plataforma interessante para estudos voltados às áreas de IA distribuída e pesquisas multiagentes.

A Figura 5 traz um comparativo entre esses ambientes de simulação citados neste capítulo e o ambiente elaborado neste trabalho.

Características	Ambiente			
	Mundo do Wumpus	Robocode	Robocup	BomberAgent
Nível de Acessibilidade	Inacessível	Inacessível	Inacessível	Acessível
Determinismo	Determinista	Não Determinista	Não Determinista	Não Determinista
Dinamicidade	Estático	Dinâmico	Dinâmico	Dinâmico
Número de Agentes	Monoagente	Multiagente	Multiagente	Multiagente
Natureza Matemática das Grandezas	Discreto	Discreto	Contínuo	Discreto
Periodicidade	Não episódico	Não episódico	Não episódico	Não episódico
Pós	Ambiente simples de implementar e de compreender	Aprendizado divertido; possibilita visualizar o desempenho dos algoritmos em tempo real	Plataforma interessante para estudos voltados as áreas de IA distribuída e pesquisas multiagentes em tempo real	Fácil desenvolvimento do mecanismos de raciocínio do agente; variedade de terrenos, ações e percepções; cenário com elementos interativos
Contras	Simplicidade do ambiente e falta de diversidade das ações e iterações do agente com o ambiente	Falta de interação dos agentes com o cenário e poucas alternativas a nível de estratégia	Falta da diversidade de classes de percepções, ações, objetivos e terrenos, além da alta complexidade para o desenvolvimento do mecanismo de raciocínio dos agentes	Ambiente em fase de desenvolvimento e falta de diversidade de personagens

Figura 5. Comparação entre ambiente de simulação.

Como pode ser observado, o *BomberAgent* procura ser um ambiente de simulação não tão simples como o Mundo do Wumpus, mas não tão complexo como a Robocup, mas com uma maior variedade em termos de número de ações e itens existentes no ambiente do que a Robocode.

4. Avaliação e Testes

Para avaliarmos o *BomberAgent*, primeiramente, realizamos testes para verificar o rendimento do ambiente de simulação à implementação de diferentes algoritmos na construção do agente. Entre os algoritmos que foram utilizados para esses testes, estão o

algoritmo de busca A^* [Russel e Norvig, 2013], que foi utilizado para auxiliar os personagens a definir caminhos para alcançar um determinado ponto do cenário e os de máquinas de estados hierárquicas [Millington e Funge, 2009], usados para a tomada de decisão dos personagens. Pode-se constatar que o ambiente de simulação apresentou um rendimento satisfatório com ambos algoritmos, pois após a realização de grande número de testes foi possível constatar um bom desempenho na tomada de decisão dos agentes, o que é um bom indicativo para jogos com tomada de decisão em tempo real.

Após a realização dos testes de desempenho do sistema, o mesmo foi apresentado para professores da área de IA, a fim de que fossem realizadas avaliações por especialistas da área sobre a aplicabilidade da ferramenta. A receptividade foi positiva o que permitirá a utilização do ambiente em sala de aula para que o mesmo possa ser avaliado e comparado a outros ambientes de simulação de forma mais empírica, próxima etapa do projeto.

5. Conclusão

O objetivo por trás do desenvolvimento do ambiente de simulação *BomberAgent* é propor uma ferramenta para o estudo, análise e aplicação de técnicas de IA em jogos, que possa ser utilizada como um ambiente de ensino e aprendizagem de IA. Sabemos que uma das principais dificuldades no ensino e aprendizado nesta área, vem da falta de oportunidades em poder colocar em prática os algoritmos e técnicas de representação do conhecimento e de raciocínio que são vistos em sala de aula. Por isso, quando criamos este ambiente de simulação nos moldes de um jogo digital, esperávamos não só disponibilizar aos usuários um ambiente para estudo e aplicações práticas, mas um mecanismo que os motivassem através de um ambiente de aprendizado agradável e divertido.

O *BomberAgent* se apresentou como uma boa alternativa aos ambientes de simulação mais conhecidos, já que o mesmo não é tão simples como o Mundo do Wumpus, nem tão complexo como a Robocup e com maior variedade de elementos do ambiente e ações que a Robocode, permitindo que os alunos de cursos iniciais de IA possam experimentar diferentes formas de implementação do mecanismo de representação do conhecimento e de raciocínio dos agentes, utilizando abordagens de lógica e de orientação a objetos, de forma gradual e divertida. Testes para avaliação da melhoria do processo de aprendizagem através do uso do ambiente de simulação *BomberAgent* estão sendo realizados.

Em sua próxima versão o objetivo é a construção de um servidor de simulação para que o mesmo possa ser utilizado através de rede, permitindo que diferentes jogadores possam disputar partidas à distância.

Referências

- ALAIBA, V.; ROTARU, A. Agent architecture for building Robocode players with SWI-Prolog. Proceedings of the International Multiconference on Computer Science and Information Technology pp. 3-7, 2008.
- BATTISTELLA, P. E.; PETRI, G.; WANGENHEIM, C. G.; WANGENHEIM, A.; MARTINA, J. E. SORTIA 2.0: Um jogo de ordenação para o ensino de Estrutura de Dados. XII Simpósio Brasileiro de Sistemas de Informação, Florianópolis, SC, 2016.
- NELSON, M.; LARSEN, F. N. Robocode. IBM Advanced Technologies, 2001.

MACHADO, L.S.; MORAES, R.M.; NUNES, F.L.S (2009) Serious Games para Saúde e Treinamento Imersivo. *Book Chapter*. In: Fátima L. S. Nunes; Liliane S. Machado; Márcio S. Pinho; Cláudio Kirner. (Org.). *Abordagens Práticas de Realidade Virtual e Aumentada*. Porto Alegre: SBC, p. 31-60.

MILLINGTON, Ian. *Artificial Intelligence for games*/Ian Millington, Jhon Funge. - 2nd ed.p.cm. Includes index.

ROBOCUP. A Brief History of RoboCup. Disponível em: <http://www.robocup.org/about-robocup/a-brief-history-of-robocup/>. Acessado em: agosto/2016

RUSSEL, S., NORVING, P. *Inteligência Artificial*. 3a ed. Elsevier, 2013.

TATAI, Victor Kazuo. *Técnicas de Sistemas Inteligentes Aplicadas ao Desenvolvimento de Jogos de Computador*. Campinas, 2003.