

MAC5701 - Tópicos em Ciência da
Computação
Agentes Inteligentes em Jogos de Computador

Aluno: Filipe Correa Lima da Silva

filipe@ime.usp.br

Orientador: Flávio Soares Corrêa da Silva

fcs@ime.usp.br

19 de junho de 2005

Sumário

1	Introdução	5
2	Fundamentos	7
2.1	Definições	7
2.2	Tipos de Agentes	8
2.3	Propriedades de Ambientes	9
3	Agentes Inteligentes em Jogos de Computadores	11
3.1	Jogos Interativos de Computador: a "Killer Application" da IA	11
3.2	Papéis de Agentes Inteligentes em Jogos	13
3.3	Projeto Soar/Games	14
3.4	Aplicando o Teste de Turing em Jogos de Computador	17
3.5	Ensino de IA utilizando jogos de Computador	17
4	Conclusão	19
4.1	Considerações finais	19
4.2	Trabalhos futuros	20

Capítulo 1

Introdução

Nos últimos anos a IA (inteligência artificial) se tornou parte essencial dos jogos para computadores [9]. A medida em que os jogos se tornam mais complexos e os consumidores exigem personagens e oponentes controlados por computador mais sofisticados, os programadores são obrigados a colocar maior ênfase no desenvolvimento da IA de seus jogos [20].

Um dos objetivos fundamentais da IA, é entender e implementar sistemas inteligentes que apresentem todas as capacidades de um ser humano. Uma aplicação emergente onde esse objetivo pode ser perseguido, e que vem chamando a atenção de vários pesquisadores recentemente [18, 20, 15, 12, 9, 5, 27, 3], são os jogos interativos de computador.

Estamos considerando neste estudo somente jogos sob a forma de ambientes virtuais com regras e restrições específicas que possam ser populados por agentes [10, 8, 25, 4], o que exclui jogos de tabuleiro e afins.

Como o ambiente virtual onde o jogo acontece pode ser tão complexo quanto se queira, implementar comportamento inteligente nos agentes que vivem nesse ambiente torna-se um problema difícil e interessante.

Este trabalho consiste em um levantamento bibliográfico sobre o uso de jogos na pesquisa em Inteligência Artificial, buscando levantar possíveis aplicações de subáreas específicas, e os benefícios de se utilizar essa tecnologia na pesquisa e no ensino de IA.

Capítulo 2

Fundamentos

Nesse capítulo iremos nos basear no trabalho de Russel e Norvig [24], pela abordagem de caracterizar a IA como o estudo de agentes que existem em um ambiente, e têm capacidade de receber informações sensoriais e agir no mesmo, o que vem de encontro às nossas necessidades de modelagem nesse estudo.

2.1 Definições

Um *agente inteligente*¹ é qualquer entidade que possa receber informações do ambiente em que vive por meio de *sensores*, e agir nesse ambiente através de *atuadores*. Com base nesse conceito, Russel e Norvig definem que AI é o estudo de como projetar agentes que atuem no ambiente buscando ter o máximo sucesso possível, o que requer que se tenha uma forma de se definir o sucesso do agente, uma medida de performance. Tal medida não deve penalizar o agente por não ter percebido coisas que não podia, e por não ter agido quando era incapaz de fazê-lo.

Chega-se então à uma definição de *agente inteligente ideal*: *para cada possível seqüência de percepção, um agente inteligente ideal deveria tomar qualquer ação possível para maximizar a sua medida de performance se baseando nas evidências fornecidas pela seqüência de percepção e quaisquer conhecimentos prévios que o agente tenha.*

¹Também chamado de *agente racional*

Onde *seqüência de percepção* são todas as informações sensoriais que o agente recebeu desde o começo de sua existência. *Conhecimentos prévios* se refere aos conhecimentos que foram incluídos na construção do agente. Se todas as ações do agente estão baseadas nos seus conhecimentos prévios, dizemos que este agente não tem *autonomia*: o comportamento pode ser adequado porém a inteligência estará toda embutida na base de conhecimento.

O comportamento de um agente pode ser baseado na sua própria experiência e na sua base de conhecimento embutida na construção, de modo que um sistema é autônomo a medida em que seu comportamento é baseado em suas próprias experiências.

2.2 Tipos de Agentes

Os agentes são divididos em 4 classes de acordo com o nível de inteligência embutida.

1. *Agentes Estímulo-Resposta*: são os agentes mais simples, providos com uma base de conhecimento formada por regras *Se-Então*², sendo que seu comportamento está totalmente codificado nessas regras.
2. *Agentes com memória*: são agentes Estímulo-Resposta que guardam estados do ambiente e que sabem como o ambiente evolui em função do tempo e em função de suas ações. Portanto as regras de produção podem se basear tanto na seqüência de percepção quanto no estado do ambiente para decidir o que agente deve fazer.
3. *Agentes guiados por objetivos*: esse tipo de agente tenta resolver o problema de se chegar a um estado específico³. Isso pode ser simples para os casos em que apenas uma ação é suficiente para se chegar a tal estado. Se o agente tem que executar uma seqüência específica de ações para chegar ao estado desejado, o problema pode ficar bastante complexo, exigindo técnicas de planejamento e busca para calcular tal seqüência de ações.

²ou regras de produção

³que pode ser uma combinação entre estado interno do agente e o estado do ambiente

4. *Agentes guiados pela função de utilidade*: pode acontecer de existirem várias seqüências distintas de ações que levem o agente a atingir seu objetivo. Por vezes é interessante desenvolver um método para avaliar quais dessas seqüências é a melhor. Por exemplo suponha que exista um número de ações que levem o agente a ficar mais perto do seu objetivo. O agente então se pergunta qual dos estados gerados por essas ações será o mais útil⁴, e então vai preferir a seqüência de ações que contenha essa ação em detrimento das outras seqüências.

2.3 Propriedades de Ambientes

Existem diversas propriedades de um ambiente que podem influenciar no projeto do agente. Uma dessas propriedades é a *acessibilidade*. Dizemos que o ambiente é *acessível* ao agente se ele pode obter o estado completo do ambiente através de seus sensores. Também dizemos que o ambiente é *efetivamente acessível* se todas as informações do ambiente necessárias para se tomar uma decisão ótima estão disponíveis aos sensores do agente.

Um ambiente pode ser *determinístico* ou *não-determinístico*. Um ambiente é determinístico se o próximo estado depende apenas do estado atual e das ações escolhidas pelos agentes no estado atual.

Dizemos que um ambiente é *estático* se não muda enquanto o agente está pensando, ou seja, o agente não precisa se preocupar com a passagem do tempo enquanto está deliberando. Se o ambiente muda com a passagem do tempo, então dizemos que o ambiente é *dinâmico*. Se o ambiente não muda com a passagem do tempo, porém a performance do agente é avaliada em função do tempo que ele demorou para deliberar, dizemos que o ambiente é semi-estático, ou semi-dinâmico.

Se pudermos dividir a experiência de um agente em episódios, cada um consistindo em uma seqüência de percepção e ação do agente, sendo que a qualidade da ação dependa somente do episódio atual, dizemos que o ambiente é episódico. Em outras palavras, o agente não precisa pensar no que poderá acontecer em

⁴ou, de acordo com Russel e Norvig [24], "em qual estado o agente estará mais feliz".

um episódio futuro se ele tomar uma determinada ação no episódio atual. Se o sucesso de uma ação depender de episódios anteriores, então dizemos que o ambiente é não-episódico.

Se existe um número limitado de informações sensoriais e ações, dizemos que o ambiente é *discreto*. Caso contrário, o ambiente é considerado *contínuo*. Podemos perceber pelas definições que o caso mais difícil é o ambiente *inacessível, não-episódico, dinâmico, contínuo e não-determinístico*. No próximo capítulo veremos alguns trabalhos que utilizam jogos como ambiente de teste e pesquisa de agentes inteligentes.

Capítulo 3

Agentes Inteligentes em Jogos de Computadores

Como vimos no capítulo 2, a inteligência artificial consiste no estudo de agentes que existem em um ambiente e que podem agir e receber estímulos desse ambiente [24]. Com jogos de computador temos efetivamente um meio onde se pode criar esses ambientes e populá-los com agentes que podem receber informações sensoriais e agir.

3.1 Jogos Interativos de Computador: a "Killer Application" da IA

Nos últimos 30 anos, a IA foi se fragmentando em campos mais especializados, enfocando problemas mais específicos e utilizando algoritmos mais e mais especializados para resolvê-los, de acordo com o trabalho de Laird e Lent em [18], enquanto tem-se feito pouco progresso em direção à construção de sistemas que se aproximem da inteligência humana, ou utilizando o termo inglês, "Human-Level AI".

Os autores definem sistemas de IA "Human-Level", como aqueles com os quais nós sonhamos quando vimos por exemplo os robôs C3PO e R2D2 no filme *Star Wars*, ou HAL em *2001, A space Odyssey*. Eles apresentam todas as caracterís-

ticas de inteligência humana como resposta em tempo-real, robustez, interação inteligente autônoma com o ambiente, planejamento, comunicação em linguagem natural, raciocínio senso comum, criatividade e aprendizagem.

Laird e Lent argumentam que jogos interativos de computador são a "Killer Application" para a pesquisa em IA, primeiro porque os jogos precisarão de IA "Human-Level" no futuro, e segundo porque os jogos fornecem os ambientes para pesquisar nos tipos específicos de problemas que levam ao tipo de pesquisa de integração e incremental necessária para obter "Human-Level" IA.

Uma lista de razões para os pesquisadores em IA levarem a indústria de jogos de computadores a sério, é fornecida pelos autores.

Primeiro, os desenvolvedores de jogos estão começando a reconhecer a necessidade de se construir personagens mais inteligentes. É interessante notar que o trabalho de Laird e Lent foi feito em 2001. A indústria *já* percebeu a necessidade e está ativamente trabalhando para construir personagens cada vez mais inteligentes [2].

Segundo, a indústria de jogos é altamente competitiva e um componente forte dessa competição é a tecnologia. Uma das tecnologias mencionadas como diferencial de sucesso para os futuros jogos é a IA.

Terceiro, *Programador de IA* já é um cargo comum na indústria [6].

Quarto, em termos de receita bruta, a indústria de jogos é maior do que a indústria do cinema.

Quinto, a tendência de mover o processo de renderização para placas gráficas libera a cpu, o que significa que podemos esperar mais processamento para os algoritmos de IA a medida que o hardware evolui.

Sexto, a indústria de jogos precisa da IA acadêmica. A ênfase atual na IA dos jogos é dar a ilusão de comportamento humano para situações limitadas, sendo que a maioria dessas técnicas não escala. A medida em que os jogos ficam mais realísticos, em termos de física e gráficos, e é de se esperar que a construção de personagens mais inteligentes seja o próximo passo em direção ao realismo. Como pesquisadores, podemos utilizar estes ambientes cada vez mais realísticos para avançar na construção de agentes cada vez mais inteligentes.

Finalmente, cabe acrescentar que o ambiente onde o jogo acontece é sim virtual, porém não é uma simulação do domínio do problema: ele é o próprio domínio do problema [1].

3.2 Papéis de Agentes Inteligentes em Jogos

Existem diferentes gêneros de jogos, cada um suportando diversos tipos de personagens que populam o ambiente desses jogos. Alguns exemplos são jogos de ação, *Role Playing Games* ou RPGs, jogos de aventura, jogos de estratégia, *God games*, esportes individuais e de equipe.

Vamos explorar alguns papéis que podem surgir no gênero dos RPGs¹. Em um jogo de RPG a idéia é imergir o jogador em um mundo imaginário onde ele tenha que interpretar papéis. Em geral o jogador tem a possibilidade de escolher entre vários tipos de personagens possíveis, como um guerreiro ou um mago. O personagem do jogador se envolve em lutas com monstros e aventuras em troca de recompensas, enquanto vai construindo seu personagem. *World of Warcraft*, *Dungeons and Dragons Online* e o brasileiro *Erinia* são alguns exemplos de jogos de RPG onde vários jogadores podem interagir em um mesmo mundo em tempo real e são batizados de MMORPG (Massively Multiplayer Online Role Playing Game).

A IA é utilizada nesses tipos de jogos para controlar inimigos, parceiros e personagens de suporte. Os inimigos podem ser por exemplo os monstros que o personagem encontra ao longo de sua aventura, ou podem ser personagens semelhantes ao do jogador porém controlados pela IA. Construir uma inteligência "acreditável" dos agentes inimigos é uma tarefa bastante complexa. Os inimigos devem ser autônomos e precisam interagir com ambientes dinâmicos e complexos, o que requer comportamento reativo, planejamento e senso comum. Para serem acreditáveis, eles também precisam ter as mesmas informações sensoriais que o jogador. Eles precisam navegar pelo extenso mundo virtual, necessitando de *path-finding*, raciocínio espacial e raciocínio temporal. Agentes avançados podem ter mecanismos para se adaptar as estratégias de seus oponentes, e podem inclusive

¹ou "jogos de RPG" como são conhecidos no Brasil

aprender.

Parceiros ou aliados constituem um problema de pesquisa ainda mais difícil que os inimigos, porque enquanto o jogador provavelmente vai ver seu inimigo uma vez em todo o jogo², o aliado estará lado a lado com o jogador durante boa parte do jogo ou durante todo o jogo. O desafio é fazer com que um agente aliado tenha comportamento humano. O jogador pode por exemplo querer conversar com o aliado, que terá então que se lembrar das conversas. O aliado pode por exemplo ficar magoado com o jogador caso este o maltrate. Ele pode jurar aliança ao jogador caso este o trate muito bem durante um tempo determinado. Estamos querendo simular emoções humanas no agente. O RPG *NeverWinter Nights*[10], é um exemplo de jogo que contém aliados persistentes. O jogador desenvolve um relacionamento com seus aliados a medida que a história do jogo se desenvolve. É interessante acrescentar que o jogo tem finais diferentes de acordo com a interação do jogador com os aliados.

Personagens de suporte são por exemplo o dono da taverna, o guarda do portão de entrada da cidade, o Rei ou o mercador mesquinho. São personagens que o jogador pode ver várias vezes durante o jogo, mas não com a mesma frequência que um aliado. Ainda assim os mesmos requisitos de inteligência necessários para a emulação de inteligência que se aplica para os aliados se aplicam também aos personagens de suporte.

3.3 Projeto Soar/Games

John E. Laird foi um dos principais responsáveis pela construção da arquitetura SOAR [19], juntamente com Allen Newell. O projeto SOAR/Games [16], liderado por Laird, utiliza jogos para a pesquisa em IA.

O objetivo do projeto SOAR/Games é aplicar técnicas do estado da arte de IA a jogos de computadores por meio do desenvolvimento de agentes inteligentes [21]. Utilizando a arquitetura SOAR, fruto de 15 anos de pesquisa em diversas universidades, foram desenvolvidos agentes que planejam e aprendem

²Em nome da jogabilidade, o jogador sempre vai vencer os combates em uma ou mais tentativas. Por isso o tempo de vida de um inimigo não é dos maiores.

e foram desenvolvidos ambientes que servem como base para testar resultados de pesquisa em aprendizagem de máquina, arquiteturas inteligentes e projetos de interface.

Um aspecto central do projeto, foi o de construir um *motor de IA*³ que pudesse ser reutilizada em diversos jogos, reduzindo assim o tempo de desenvolvimento, porque o desenvolvedor precisaria apenas adaptar o motor de IA às particularidades de cada jogo. No motor de IA reside toda a inteligência utilizada pelos agentes no jogo. O motor de IA deve suportar vários tipos de agentes: agentes reativos, específicos ao contexto, flexíveis e realísticos. Como requisito adicional, um motor de IA deve facilitar o desenvolvimento de agentes.

Laird divide o motor de IA em três componentes: a máquina de inferência, a base de conhecimento, e a interface com o jogo. Laird utiliza a arquitetura SOAR como sendo a máquina de inferência. O objetivo da máquina de inferência é aplicar o conhecimento do agente à situação atual. A situação atual do agente é representada por estruturas de dados e informação contextual. A máquina de inferência está constantemente operando em um ciclo de decisão: perceber, pensar e agir.

A base de conhecimento da SOAR consiste em uma hierarquia de operadores, sendo que cada operador consiste em um conjunto de regras de produção. A cada ciclo de decisão, a SOAR decide quais operadores estarão ativados em todos os níveis da hierarquia. A SOAR tem uma memória interna capaz de persistir tanto os operadores como informações sensoriais disponibilizadas pela interface.

A interface, o segundo componente do motor de IA é o canal de comunicação da máquina de inferência com o ambiente virtual. Ela tem a responsabilidade de extrair informações sensoriais do ambiente e alimentar a máquina de inferência, e comunicar ao ambiente as ações escolhidas pela máquina de inferência. Como os jogos podem variar bastante em termos do ambiente virtual em que se inserem, cada motor de IA vai precisar de uma interface diferente. A interface deveria disponibilizar para a máquina de inferência exatamente as mesmas informações sensoriais que um jogador humano recebe. Laird diz que a interface deve acessar a estrutura de dados de jogo diretamente, evitando assim todos os problemas de visão computacional envolvidos.

³do inglês *AI engine*

O último elemento de um motor de IA é a base de conhecimento. Como cada jogo é diferente, é impossível pensar em reaproveitar totalmente a base de conhecimento. Mas um motor de IA pode disponibilizar uma base de conhecimento geral para um gênero específico de jogos. Ela seria composta de objetivos, táticas e comportamentos independentes do jogo.

Experimentos com 2 jogos comerciais foram realizados, *Quake2* e *Descent3*. Para cada jogo foi construída uma interface e uma base de conhecimento específica. No jogo *Quake2*, foi implementado um agente, chamado quakebot, que tem por objetivo jogar quake tão bem quanto um humano. O quakebot derrota facilmente iniciantes e é um forte desafio para jogadores mais experientes. O agente implementado para o jogo *Descent3* consegue explorar os níveis do jogo derrotando os monstros existentes. Adicionalmente, foram feitos experimentos utilizando o sistema KnoMic (*Knowledge Mimic*) [26], onde o agente quakebot foi capaz de aprender assistindo um jogador experiente jogar *Quake2*. Em [14], a capacidade de antecipação embutida no quakebot é descrita.

Laird cita algumas áreas que se beneficiam diretamente do projeto Soar/Games: modelagem de oponentes, coordenação de agentes, processamento de linguagem natural e planejamento. O autor ainda acrescenta em [20], que os jogos são facilmente entendíveis, orientados a ação e têm um forte apelo visual, e que esses fatores levam a uma acessível e emocionante demonstração de pesquisa aplicada em inteligência artificial.

Outro trabalho recente do grupo de pesquisa de Laird é desenvolvimento de um jogo de aventura onde agentes inteligentes fazem realmente diferença. O jogo está sendo implementado como um mod do jogo *Unreal Tournament* e é chamado de *Haunt2*. O jogador controla um fantasma, trazido de sua dimensão por um cientista malvado que o aprisiona em uma casa em que habitam um número de personagens que são agentes inteligentes. O foco da aventura está na interação do jogador com os agentes. O objetivo do jogo é conseguir uma forma de voltar para a dimensão, usando para isso poderes especiais para influenciar os agentes e tendo como desafio contornar as limitações impostas por um corpo etereal [17, 5, 15, 16].

3.4 Aplicando o Teste de Turing em Jogos de Computador

Astrid Glende propõe uma forma de se aplicar o teste de Turing em jogos de computador. Em [9], Glende discute a obtenção dos critérios necessários para emular comportamento humano, em outras palavras passar no teste de Turing, no domínio dos jogos.

O jogador precisa acreditar que as ações tomadas pelos agentes sejam inteligentes para que o agente possa passar no teste de Turing. Para isso o jogador não pode saber se o personagem que ele está vendo é controlado por computador ou não porque isso iria tornar tendencioso o julgamento. Por isso Glende assume que o jogo aconteça em rede porque assim o jogador não sabe quem está controlando o personagem.

Glende apresenta um conjunto de características comportamentais que um agente controlado por um humano deve apresentar: imprevisibilidade, criatividade na resolução de problemas, personalidade, objetivos, autonomia, improvisação, planejamento e aprendizagem. Para que o agente passe no teste de Turing, essas características devem ser mapeadas no agente inteligente.

Glende conclui dizendo que a complexidade dos jogos modernos de computador se deve à sua natureza de tempo-real, a sua dinâmica, o conhecimento incompleto do ambiente e a restrição de recursos, problemas que, como vimos no capítulo 2, aparecem nos tipos de ambientes mais complicados.

A existência de uma maneira de se aplicar o teste de Turing atesta a validade de se abordar os jogos de computador como um ambiente alternativo de pesquisa em Inteligência artificial.

3.5 Ensino de IA utilizando jogos de Computador

O ensino é outra área interessante de aplicação onde os jogos podem ser utilizados. Em particular, vários trabalhos vêm sendo realizados em direção à utilização de jogos de computadores no ensino de ciência da computação.

Randolph Jones apresenta um curso de projeto e implementação de jogos de computadores em [11], defendendo que o curso provê um ambiente ideal para os estudantes integrarem uma vasta base de conhecimentos e habilidades no curso de ciência da computação apresentando a variedade de conceitos de ciência da computação necessários para o desenvolvimento de um jogo.

O trabalho de Ron Coleman et al. [7], descreve o desenvolvimento de um currículo com concentração de disciplinas voltadas para o desenvolvimento de jogos de computadores.

Lasse Natvig e Steinar Line descrevem o AoC [22], *Age of Computers*, um jogo baseado em interface web onde podem ser ensinados conceitos fundamentais de computadores para um curso com 250 pessoas. Os autores relatam que a resposta dos alunos foi bastante positiva e foi uma motivação forte para continuar com o projeto.

Dentro do ensino de computação, nosso foco está em utilizar jogos de computadores como ferramentas no ensino de inteligência artificial. A idéia central está na disponibilização de um ambiente virtual onde os alunos possam efetivamente construir e visualizar os agentes inteligentes em ação. Em particular existe um gênero de jogo chamado *jogo de programação*, onde os jogadores programam a IA de seus agentes e os colocam para competir em um ambiente virtual, que pode ser utilizado para este fim. São exemplos desses jogos *GUN-TACTYX* [4], *Robocode* [23] e *Robocup* [13].

Capítulo 4

Conclusão

4.1 Considerações finais

Neste estudo, foi realizado um levantamento bibliográfico de trabalhos na área de Inteligência Artificial que utilizam jogos de computadores como ambientes de apoio a pesquisa. Foi verificado que os jogos de computador fornecem ambientes flexíveis, robustos e complexos para explorar tanto o desenvolvimento quanto o ensino de IA.

Para a pesquisa em IA, os jogos fornecem ambientes virtuais complexos, necessários para a exploração de problemas no estado da arte da IA como planejamento, modelagem de oponentes, aprendizagem, cooperação de agentes e processamento de linguagem natural.

No ensino de IA, os jogos podem atuar como um ambiente virtual onde os alunos podem construir seus agentes e visualizar seu comportamento. Trabalhos realizados na direção de utilizar jogos no ensino de ciência da computação mostraram que a abordagem é efetiva e encontra uma excelente resposta por parte dos alunos.

4.2 Trabalhos futuros

Como extensão deste estudo, pretendemos implementar um jogo voltado para o ensino de ciência da computação, como parte da dissertação de mestrado. Em uma primeira fase, vamos nos concentrar no ensino de inteligência artificial utilizando jogos. O jogo deve ser interessante para motivar os alunos, e ao mesmo tempo fácil de modificar para poder ser utilizado com eficácia em diversos tipos de aulas e demonstrações.

Como Prolog é uma das linguagens populares na pesquisa em IA, vamos estudar maneiras de se escrever a IA dos agentes do jogo em programas Prolog separados da lógica do jogo. Para isso, o motor do jogo deverá carregar dinamicamente os programas prolog e incorporar na lógica do jogo em tempo de execução, eliminando a necessidade de recompilações sempre que se quiser modificar alguma funcionalidade. Cabe acrescentar que não encontramos na literatura a utilização de prolog para a codificação da IA de um jogo, o que pode potencialmente constituir um desafio. Nossa motivação para utilizar prolog reside no fato de que é uma linguagem mais apropriada, desenvolvida especialmente para o estudo de IA e que assim sua aplicação no domínio dos jogos pode se mostrar interessante em se tratando de ensino de IA.

Referências Bibliográficas

- [1] Robert St. Amant and R. Michael Young. Links: artificial intelligence and interactive entertainment. *Intelligence*, 12(2):17–19, 2001. 13
- [2] International Game Developers Association. The 2004 report of the igda’s artificial intelligence interface standards committee. <http://www.igda.org/ai/report-2004/report-2004.html>. 12
- [3] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 47–54. ACM Press, 1995. 5
- [4] Leonardo Boselli. Gun-tactyx. <http://gameprog.it/hosted/guntactyx>. 5, 18
- [5] Mazin Assanie Alex Kerfoot Devvan Stokes Brian Magerko, John E. Laird. Ai characters and directors for interactive computer games. In *Proceedings of the 2004 Innovative Applications of Artificial Intelligence Conference*. AAAI Press, July 2004. 5, 16
- [6] Mike Brockington and Scott Greig. Gdc 2003: Neverwinter nights client/server postmortem: How i learned to stop worrying and love the magic missile. http://www.gamasutra.com/gdc2003/features/20030306-/brockington_02.htm. 12
- [7] Ron Coleman, Mary Krembs, Alan Labouseur, and Jim Weir. Game design & programming concentration within the computer science curriculum. *SIGCSE Bull.*, 37(1):545–550, 2005. 18

- [8] Blizzard Entertainment. Warcraft3: Reign of chaos. <http://www.blizzard.com/war3>. 5
- [9] Astrid Glende. Agent design to pass computer games. In *ACMSE'04: Proceedings of the 42nd annual ACM Southeast regional conference*, pages 414–415. ACM Press, 2004. 5, 17
- [10] BioWare Inc. Neverwinter nights. <http://nwn.bioware.com>. 5, 14
- [11] Randolph M. Jones. Design and implementation of computer games: a capstone course for undergraduate computer science education. In *SIGCSE '00: Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, pages 260–264, New York, NY, USA, 2000. ACM Press. 18
- [12] G. A. Kaminka, S. Schaffer, C. Sollitto, R. Adobbati, Andrew N. Marshal, Andrew S. Scholer, and S. Tejada. Gamebots: a flexible testbed for multiagent team research. *Communications of the ACM*, 45(1), January 2002. 5
- [13] Atsumi Laboratory. Robocup of atsumi laboratory. <http://www.intlab.soka.ac.jp/~matsumi/document/research/robocup/index-e.html>. 18
- [14] John E. Laird. It knows what you're going to do: adding anticipation to a quakebot. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 385–392, New York, NY, USA, 2001. ACM Press. 16
- [15] John E. Laird. Research in human-level ai using computer games. *Communications of the ACM*, 45(1):32–35, January 2002. 5, 16
- [16] John E. Laird and et al. Soar/games project. <http://winter.eecs.umich.edu/>. 14, 16
- [17] John E. Laird and et al. A test bed for developing intelligent synthetic characters. In *Spring Symposium on Artificial Intelligence and Interactive Entertainment, AAAI*, 2002. 16
- [18] John E. Laird and Michael van Lent. Human-level ai's killer application: Interactive computer games. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of*

- Artificial Intelligence*, pages 1171–1178. AAAI Press / The MIT Press, 2000. 5, 11
- [19] John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: an architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987. 14
- [20] John E. Laird and Michael van Lent. Intelligent agents in computer games. In *Proceedings of the National Conference on Artificial Intelligence*, pages 929–930. AAAI Press, July 1999. 5, 16
- [21] John E. Laird and Michael van Lent. Developing an artificial intelligence engine. In *Proceedings of the Game Developers Conference*, pages 577–588. IGDA, March 1999. 14
- [22] Lasse Natvig and Steinar Line. Age of computers: game-based teaching of computer fundamentals. In *ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, pages 107–111, New York, NY, USA, 2004. ACM Press. 18
- [23] Mathew Nelson. Robocode. <http://robocode.sourceforge.net/>. 18
- [24] Stuart J. Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*, page 1. Prentice Hall, 1995. 7, 9, 11
- [25] Id Software. Quake3 arena. <http://www.idsoftware.com/games/quake-/quake3-arena>. 5
- [26] Michael van Lent and John E. Laird. Learning procedural knowledge through observation. In *K-CAP 2001: Proceedings of the international conference on Knowledge capture*, pages 179–186, New York, NY, USA, 2001. ACM Press. 16
- [27] R. Young. An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. In *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, AAAI Press (2001). AAAI Press, 2001. 5