

인공지능 엔진

이지형* · 윤태복

(성균관대학교 정보통신공학부)

1. 서 론

포괄적인 개념에서, 게임에서 인공지능이란 PC (Player Character, 사용자에게 의해서 조정되는 게임 오브젝트)를 제외한 모든 것들을 제어하기 위해 사용되는 일련의 기술이라고 정의할 수 있다. 예를 들면, PC를 둘러 싸고 있는 환경, PC를 대적하거나 경쟁관계에 있는 NPC(Non-Player Character), PC에 조언을 해주는 조인자 NPC 등을 제어하고 사용자가 느끼기에 지능적이면서 실제적인 것으로 보이도록 하여 게임의 흥미를 더하기 위한 기술들이다. 따라서 이러한 관점에서 학술적인 인공지능과 약간 다른 특성을 갖는다고 할 수 있으나, 이것은 학술적인 것이 산업에 응용되면서 갖게 되는 특성이라고 이해해야 할 것이다.

이러한 이유로 게임의 인공지능 기법에서 주로 사용되는 것들은 학술적인 관점에서 인공지능의 연구분야가 아니거나, 이미 많은 연구가 이루어진 것이 많이 있다. 그러나 근본적으로 학술적으로 연구되는 인공지능 기법이나 게임에서 사용되는 인공지능 기법은 모두 기계가 지능을 갖도록 혹은 사용자가 보기에 지능을 가진 것처럼 보이게 하기 위한 것이라는 관점에서는 모두 동일한 목표를 갖고 있다고 할 수 있다.

최근에 게임에서 인공지능 기법에 많은 관심을 갖기 시작하고 있다. 게임의 태동기라 할 수 있는 1970년대에는 게임 속에서 매우 간단한 형태의 인공지능 기법들이 사용되었고, 1980년부터 1990년대 초반까지도 인공지능은 큰 관심의 대상은 아니었다. 그 이유는, 그 동안 게임의 기술적인 측면에서 주요 관심은 그래픽이었으며, 현실적으로도

CPU 사용의 대부분이 그래픽에 할당되었기 때문에 게임에서 인공지능에 사용될 자원이 많지 않았던 것도 원인이었다. 그러나 1990년대 중반 이후 그래픽 전용 프로세서가 개발되면서 그래픽 작업의 대부분을 감당하여 그래픽 이외의 작업이 자원을 더 사용할 수 있게 되었다. 또한 게임의 발달로 좀 더 지능적이고 현실적인 NPC의 등장 필요로 해지면서, 게임에서 인공지능의 중요성을 점차 커지게 되었다. 이러한 현상은 게임개발회사의 인적 구성과 게임에서 인공지능을 구현하기 위해서 사용되는 CPU의 사용량을 살펴보아도 알 수 있다. 1997년 인공지능 프로그래머를 채용하고 있는 회사가 미국의 경우 대략 24% 정도였는데, 2000년에는 80%가 넘었으며, 인공지능에 사용되는 CPU 시간도 1997년에 5%이던 것이 2000년에는 30%를 넘게 되었다.

최근의 인공지능으로 유명한 게임들은 The Sims, Black & White, Quake, Half-Life, Unreal 등 있다. 이러한 게임들은 인공지능을 구현하기 위하여 대부분 자신들이 개발한 인공지능 엔진을 사용하고 있다.

다음 절에서는 이러한 게임에서 사용되는 인공지능의 요소 기술에 대해서 간략하게 살펴보고, 3절에서는 현재 이용할 수 있는 일반적인 인공지능 엔진과 상용 게임 인공지능 엔진 및 미들웨어(Middleware)에 대해서 정리하고, 4절에서는 게임 인공지능 엔진의 국외 및 국내 동향에 대해서 살펴본 후, 결론을 맺는다.

2. 인공지능 엔진의 요소 기술

게임에서 사용되고 있는 인공지능 기법은 매우 다양하다. 대표적인 것을 나열하면, 탐색(Path

*E-mail: jhlcc@ece.skku.ac.kr

Finding)¹⁾, 유한상태기계(Finite State Machine)³⁾, 스크립트(Script)³⁾, 플로킹(Flocking)³⁾, 퍼지집합(Fuzzy Set) 및 퍼지상태기계(Fuzzy State Machine)¹⁾, 결정트리(Decision Tree)²⁾, 인공신경망(Neural Network)²⁾, 유전자알고리즘(Genetic Algorithm)²⁾ 등이 있다. 이러한 기법 중에서 현재 게임에서 제일 많이 사용되고 있는 것은 탐색과 유한상태기계이다.

2.1 탐색

언급한 것과 같이 탐색은 현재 게임에서 가장 많이 사용되는 기법 중의 하나이다. 게임 중에서 NPC가 PC가 위치한 곳까지 혹은 특정 목표지점까지 찾아 가야 할 필요가 있다. 이러한 탐색 기법은 어느 장르의 게임에서도 필요한 것이다. 예를 들면, PC와 전투를 하는 NPC의 경우, 전투를 위해서는 PC가 위치한 곳까지 찾아 가야 하는데, 지도를 참고해서 가장 빠른 길로 PC가 위치한 곳에 다다라야 한다. 이때 보통 많이 사용되는 방법은 A*(A star) 탐색이라 불리는 휴리스틱(Heuristic) 탐색 기법이다. A* 탐색은 주어진 지도에 존재하는 길을 그래프로 모델링하고, 어느 특정 위치에서 목적지까지 거리 혹은 비용을 예측할 수 있는 휴리스틱 함수를 이용하여 가장 효율적인 길을 찾게 된다. 이때 사용되는 휴리스틱 함수는 사전에 사용자가 정의한다.

그러나 일반적으로 가장 효율적인 길을 찾는 것은 매우 큰 비용을 필요하게 되므로, 실시간으로 진행되는 게임에서 매 순간마다 A* 탐색을 이용하여 효율적인 길을 찾는 것은 비효율적일 수 있다. 따라서 길찾기를 효율적으로 하기 위해서 지도에 있는 길을 단순화 시키거나 어느 몇 지점 사이의 이동은 고정된 길을 이용하는 방법을 이용하기도 한다.

2.2 유한상태기계

NPC가 게임 중에서 취하게 되는 행동은 여러가지로 구성된다. 예를 들면, PC와 전투를 하는 NPC는 자신의 에너지가 전투에 충분하고 PC가 근처에 있으면, '접근'을 하고, PC와 만나게 되면 '전투'를 하며, 전투 중 피해가 커지면 '도망'을 하고, 전투에서 승리하면 다른 PC를 찾아 '탐색'을 한다. NPC는 자신의 상태와 주변 상황에 따라서

이러한 행동을 적절히 실행하여야 하는데, 이것을 구현하는데 주로 사용되는 기법이 유한상태기계이다. 유한상태기계란 사전에 정의된 몇가지 상태와 각 상태 사이의 전이 조건으로 구성된다.

유한상태기계가 많이 사용되는 이유는, 일반적으로 게임에서 사용되는 NPC의 행동 반응 패턴이 유한상태기계로 잘 표현될 수 있기 때문이다. 또한 이것은 구현하기에 간단하고, 이해하기 쉬우며 또한 게임 개발 중에 디버깅(debugging)하기에도 용이하기 때문이다. 이러한 이유로 유한상태기계는 게임의 NPC들의 지능적 혹은 지능적으로 보이는 행동패턴을 구현하기 위해서 가장 많이 사용되는 기법이다.

그러나 이 기법의 단점은 어느 상태에서 어느 특정 조건만 만족되면 항상 특정 상태로의 전이가 결정되어 있다는 것이다. 즉, 이러한 패턴에 사용자가 익숙해지면 더 이상 NPC의 행동은 지능적으로 보이지 않게 되고, 게임의 흥미는 줄어들 수 밖에 없다.

2.3 스크립트

스크립트는 일종의 프로그래밍 언어로써 특정목적을 위해서 간단하고 사용하기 쉽도록 개발된 것이다. 일반적으로 게임의 흥미로운 진행을 위해서는 많은 사항이 구현 시험의 많은 반복을 통하여 결정되는 경우가 많다. 특히 이러한 구현 시험은 프로그램의 전문가들이 아닌 게임 디자이너들이나 시험 사용자 그룹이 되는 경우가 많이 있다. 따라서 이러한 구체적인 사항을 프로그래머가 아닌 사람들도 구현하고 테스트 해 볼 수 있도록 사용하기 쉽고, 구현 시험의 반복이 용이하도록 스크립트 언어를 이용한다. 대표적인 스크립트 언어로는 Quake의 QuakeC와 Unreal의 UnrealScript 등이 있다.

이러한 스크립트 언어의 기능은 게임마다 다양하여, 어떤 것은 게임의 단순한 구성 변경 정도만 할 수 있는 것부터, 게임의 모든 진행을 변경할 수 있는 것까지 있다.

2.4 플로킹

플로킹은 어떤 집단의 행동을 시뮬레이션 하는데 사용되는 기법을 의미한다. 예를 들면 게임에서 한 무리의 전사(戰士)들이 이동하는 모습이나 NPC 몬스터(Monster)들이 무리를 지어 PC를 찾아 다니는 모습 등을 표현할 때 이용된다. 플로킹은 이미 계

임 분야에 앞서 광고제작과 같은 애니메이션 부분에서 양떼의 이동 모습, 하늘을 나르는 새떼의 모습 등의 표현에 성공적으로 사용되고 있다.

플로킹 기법이 사용되기 이전에는 어떤 집단의 이동 모습을 표현하기 위해서 그 집단을 구성하는 각 개체가 어떤 경로로 움직여야 하는지를 모두 기술하였다. 따라서 집단을 구성하는 개체가 많아지면 집단의 이동 모습을 표현하는 것은 매우 힘든 작업이 되고, 따라서 자연스러운 이동 모습의 생성도 용이하지 않았다.

이에 반해서 플로킹은 어느 집단에 속하는 개체가 간단한 몇 개의 규칙에 따라서 무리를 따라서 이동한다고 가정한다. 예를 들면 개체 주변에 있는 다른 개체의 위치와 간단한 물리 법칙에 따라서 방향을 어떻게 결정한다 등에 관한 것이다. 이러한 가정과 규칙을 단순 반복적으로 각 개체에 적용함으로써 개체의 자연스러운 이동 모습이 표현된다.

2.5 퍼지집합 및 퍼지상태기계

퍼지집합은 애매한 개념, 예를 들면 “빠르다”, “느리다”, “많다”, “적다” 등의 개념을 쉽게 표현하고 다룰 수 있게 한다.

그리고, 일반적으로 인간이 갖고 있는 지식이나 경험은 명확한 값으로 표현되기 보다는 이러한 애매한 개념으로 표현되는 경우가 많다. 예를 들면, “PC의 에너지가 높으면, 빨라 도망하라” 등과 같이 ‘높다’라는 것이 무엇인지, ‘빨리’라는 것이 무엇인지에 대한 명확한 언급 없이도 사람들은 이 규칙을 이해하고 실제로 적용한다.

이러한 이유로 퍼지집합을 이용하면 인간의 지식을 쉽게 표현할 수 있어, 인간 지식의 모델링이 필요한 경우에 많이 이용된다. 따라서 게임 속에서 NPC의 행동 패턴이나 기타 상황 변화에 게임 디자이너들이 갖고 있는 규칙을 게임에 반영하는데 유용하게 사용될 수 있다.

퍼지상태기계는 퍼지집합이론을 유한상태기계와 결합한 것이다. 일반적인 유한상태기계에서 어느 한 상태에서 다른 상태로의 전환은 정해진 간단한 규칙에 의해서, 예를 들면 “에너지가 50 이상”과 같은 단순한 규칙들에 의해서 이루어진다. 이미 언급했듯이 이러한 단순성은 유한상태기계의 단점이 될 수가 있는데, 이를 보완하기 위해서 퍼지집합 개념

을 결합한 것이다. 즉, 상태 전이 조건으로 “에너지가 높으면”과 같이 인간이 느끼는 것과 유사하게 설정함으로써 다양한 변화를 추구할 수 있다.

2.6 결정트리

결정트리는 입력과 출력이 이산(discrete) 값을 갖는 함수를 근사화하는데 적절한 방법이다. 이 방법은 함수를 관찰하여 얻어진 입출력 데이터를 바탕으로 학습하여 관찰되지 않은 입력에 대한 출력 값을 예측할 수 있게 한다. 이 방법은 학습이 비교적 빠르고 쉬우며, 관찰한 데이터에 오류가 있거나 데이터의 차원(dimension)이 높아도 잘 동작하는 특징을 갖는다.

이 방법은 주어진 입출력 데이터로부터 출력값을 가장 잘 구분해 낼 수 있는 입력변수가 무엇인지를 하나씩 식별해 내고 그것을 트리(Tree) 구조로 구조화함으로써 학습이 이루어진다. 학습후에 관찰되지 않은 입력값을 주면 생성된 트리구조를 따라 내려가면 예측되는 출력값을 얻을 수 있다. 이 방법을 이용한 대표적인 게임으로는 Black & White가 있다.

2.7 인공신경망

인공신경망은 인공지능 분야에서 학습을 위해서 사용되는 대표적인 기법으로, 그 구조는 인간의 두뇌와 유사하게 간단한 계산을 수행하는 노드(node)들과 각 노드를 서로 연결하는 링크(link)로 이루어져 있다. 일반적으로 학습에 많이 사용되는 인공신경망 구조는 다층 퍼셉트론(Multilayer Perceptron)이며, 대표적인 학습방법은 오차역전파방법(Error Back-Propagation)이다.

인공신경망은 주어진 입출력 데이터를 학습하여, 데이터에 내재된 패턴을 습득하게 된다. 이러한 학습 기능을 이용하면 추론, 분류, 예측 등의 수행할 수 있다. 예를 들면, 게임 중 PC의 행동 패턴에 대한 데이터를 수집하여 그것을 학습하면 PC가 어느 상황에서 어떤 행동 패턴을 보이는가를 습득할 수 있다. 이를 바탕으로 PC와 유사한 행동 패턴을 보이는 NPC를 생성해 낼 수도 있고, 어떤 특정 상황에서 PC가 어떤 행동을 할 것인가를 추론 또는 예측할 수도 있다.

그러나, 인공신경망은 학습에 많은 시간이 걸리

고, 좋은 학습 결과를 얻기 위해서는 전문가의 도움이 필요하다는 문제점을 갖고 있어, 현재 몇몇 게임에서 부분적으로 이용되고 있다.

2.8 유전자알고리즘

유전자알고리즘은 생명체의 진화를 모방하여 최적화 문제를 해결하는 기법이다. 진화과정이 g환경에 잘 적응한 생명체들의 DNA가 서로 섞이는 교배(crossover)와 돌연변이(mutation)을 통해서 지속적으로 환경에 적응하는 생명체가 탄생하는 과정이라는 것에 착안하여, 최적화 문제의 해답을 0과 1의 스트링으로 표현하여 가상 DNA를 만들고, 이것에 적절한 교배와 돌연변이 연산을 적용하여 더욱 최적화된 해답을 찾아가는 방법이다.

유전자알고리즘은 수학적 분석이 어렵거나 비선형적인 문제에서도 좋은 성능을 보여, 과학, 공학, 경영학 등에 널리 이용되고 있다. 그러나 아직 게임에서의 응용은 그리 활발하지는 않는데, 길찾기, 최적의 조합 찾기, PC에 효과적으로 대응하는 NPC 생성 등에 사용될 수 있을 것이다.

3. 게임 AI 엔진

정보 산업의 급격한 발전과 더불어 자동화 기술과 지능적인 기능 구현을 위해 사용되는 인공지능의 필요성은 증대 되었으며, 이런 요구사항에 발맞춰 여러 분야에서 사용할 수 있는 인공지능 미들웨어가 개발되어 사용되고 있다. 현재 주로 사용되고 있는 인공지능 미들웨어에는 게임과 영화 같은 엔터테인먼트(entertainment) 산업에서 공용으로 사용되는 범용적 부류와 웹마이닝(Web Mining), 생명정보학(Bioinformatics) 등의 특정한 분야에서 사용될 수 있는 특정적 부류, 그리고 퍼지(Fuzzy)와 유전자 알고리즘(Genetic Algorithm)과 같은 하나의 인공지능 기술에 대해 라이브러리 형태로 제공해주는 부분적 부류로 나눌 수 있다. 다음은 인공지능 미들웨어에 대해 좀 더 자세히 살펴본다.

3.1 AI Implant

캐나다의 BioGraphic Technologies에 의해 개발되었다. 실시간 3D AI SDK이며, 캐릭터의 길

찾기(Pathfinding)또는 자연스런 행동 연출이나 지능적인 행동을 표현하기에 적합하다. 3D Max나 Maya와 같은 애니메이션 툴(Animation tool)이나 렌더링 툴(Rendering tool)과 연동하여 사용하기 쉽고 Unix, MS Window, Sony PlayStation, Nintendo GameCube, MS XBox와 같은 다양한 플랫폼을 지원한다. 사용 범위는 게임, 애니메이션, 영화, 시뮬레이션에서 주로 사용된다⁴⁾.

3.2 DirectIA

DirectIA(Direct Intelligent Adaptation)는 프랑스의 Mathematiques(MASA)에 의해 개발 되었다. 실제 인간과 유사한 행동 모델링 구현과 시스템 자원을 효율적으로 사용하는 것이 특징이다. C++과 유사한 스크립트 언어를 지원하며 다양한 플랫폼에서 사용 가능하다. 경영·군사 시뮬레이션 및 게임에서 주로 사용된다⁵⁾.

3.3 Renderware AI

Renderware Platform(RenderWare Studio, Graphics, Audio, AI, Physics) 중에 인공지능 기능을 담당하는 부분이며, 프랑스의 인공지능 도구개발업체인 키노곤(Kenogon)에 의해 개발되었다. 기능별로 C++의 클래스 구조로 되어 있어 사용하기 용이 하다. 주로 지각능력(Perception), 3D 길찾기(PathFinding), 행동처리(Behaviors)에 대해 효과적으로 사용할 수 있도록 제작되었다⁶⁾.

3.4 SimBionic

캘리포니아의 Stotter Henke 사에 의해 개발되었고, 게임 인공지능 미들웨어 전문 도구이다. GUI 환경의 비주얼한 제작툴을 제공하며, 복잡한 행동처리에 대하여 실시간으로 확인할 수 있는 기능을 제공한다^{7,8)}.

3.5 Biomind

생명공학 분야에서 전문적으로 사용되는 DNA 데이터 분석을 위한 인공지능 미들웨어이다. Pattern Mining, Clustering Categorization, Genetic Network Inference, BiomindDB Browsing 기능을 사용하여 KEGG(Kyoto Encyclopedia of Genes and Genomes), SGD(Sacharomyces Genome Database

or stanford)와 같은 외부 데이터베이스의 지식 기반 아래 유전자를 분석하고 추론한다. 또한 텍스트 마이닝(Text Mining)에도 사용되며, J2EE와 XML 기반 환경과의 인터페이스도 지원한다⁹⁾.

3.6 BBE(Brainiac Behavior Engine)

게임에서 상대 플레이어에 대한 동적이고 지적인 모습을 연출하며 오브젝트의 상태 전이에 관련된 구현이 용이하도록 개발되었다. 엔진의 사용에 유연성이 높아, 프로그래밍 기술이 없이도 사용 가능한 것이 특징이다¹⁰⁾.

3.7 Spark!

실시간 퍼지논리 에디터(Editor) 및 뷰어(Viewer)이며, 공개 라이브러리인 FFLL(Free Fuzzy Logic Library)을 이용하여 개발되었다. 퍼지 멤버십 함수의 추가가 용이하고, 룰 생성이 쉬우며, 실시간 디버깅이 가능하고, 다른 엔진과 통합이 간단한 한 것이 특징이다¹¹⁾.

3.8 Quake engine

id Software 에서 자사의 FPS(First Person Shooting)게임 장르의 인공지능 기능 지원을 위해, Michigan 대학에서 개발한 규칙 기반의 SOAR엔진을 이용하여 개발하였다.

QuakeC이라는 스크립트를 이용하여 NPC(Non-Player Characters)를 제어하며, 플레이어의 패턴에 대한 기억능력이 있어, 지능적인 반응 모습을 보여준다. 서버와 클라이언트에서 모두 적용 가능한 것이 특징이다^{16,17)}.

3.9 기타

그 외에도 범용적인 AI 기술을 이용하여 인지기술, 지식 관리, 지식 처리기술을 제공해 주는 WebMind^{14,15)}가 있으며, 원하는 지점간의 적합한 경로를 찾아 제공해 주는 기술에 사용되는 Path-Engine¹⁸⁾과 유전자 알고리즘 기능을 지원하는 GALib¹³⁾, 퍼지 기능을 제공하는 FFLL¹¹⁾ 등이 라이브러리 형태로 제공되고 있다.

범용적으로 개발된 인공지능 미들웨어의 경우 원하는 기능을 구현하기 위해서는 엔진 및 툴에 대한 적응시간이 필요하다는 단점이 있다. 하지만 사용자

의 요구가 반영되고 기능이 향상되면서 사용하기 쉽게 변하고 있는 추세이다.

4. 국외 및 국내 동향

게임에서 인공지능의 사용은 여러 이유로 점점 늘고 있는 것이 세계적인 추세이며, 많은 게임 개발 회사들이 인공지능에 많은 관심을 갖고 있다. 그러나 게임 개발에 상용 인공지능 엔진이나 미들웨어가 사용되기 보다는 게임 개발과 함께 필요한 인공지능 기법을 직접 구현하는 경우가 많다.

그 이유는 각 게임에서 필요로 하는 인공지능 기법에 대한 기능이 매우 다양하기 때문에, 어느 한 상용 엔진이 이를 모두 만족시키기 어렵기 때문이다. 또한 인공지능 기법들은 각 상황 별 구체적인 데이터와 자료구조에 의존하여 계산이 이루어지는데 각 게임의 상황을 표현하는 구체적인 데이터와 자료구조가 독자적인 형태로 이루어 질 수 밖에 없는데, 이러한 비표준화는 상용 인공지능 엔진의 사용을 어렵게 만드는 원인이 되고 있다. 또 다른 원인으로서는, 일반적으로 인공지능 기법들은 많은 계산량을 필요로 하는데, 현재 CPU 사용의 상당 부분을 인공지능에 할당한다 하더라도 일반적인 인공지능 기법이 사용되기에는 부족한 실정이다. 따라서 인공지능 기법을 적은 자원을 이용하여 구현하는 방법을 찾다 보니, 자연스럽게 게임에 의존적으로 최적화된 방식으로 구현되기 때문이다.

그러나 이러한 개별적인 개발의 어려움과 비효율로 인하여 많은 인공지능 개발자들이 상용엔진의 사용에 관심을 갖고 있어 새로운 상용엔진의 개발이 증가하고 있으며, 이에 따라 상용 엔진의 사용도 점차 늘 것으로 예상되고 있다.

이에 비해 국내의 경우 아직까지도 인공지능에 대한 관심은 매우 약한 편이다. 아직 우리나라의 게임산업이 외국에 비하여 전반적으로 성숙되지 않은 탓도 있지만, 대부분의 인기 게임이 MMOG(Massively Multi-user Online Game) 장르인 상황에서 인공지능 보다는 3D 그래픽과 서버/클라이언트 기술에 우선적으로 투자할 수 밖에 없는 상황이기도 하다. 그러나 장기적으로 국내에서도 더 흥미롭고 더 자연스러운 게임의 제작을 위해서 인공지능에 대한 관심이 점점 증가 할 것이며, 이에 따

라 인공지능 엔진의 개발과 사용에 관심을 갖게 될 것이다.

5. 결 론

게임분야에서 사용되는 인공지능 기법과 인공지능 엔진에 대해서 대략적인 소개를 하였다. 최근에 인공지능의 대표적인 응용분야로 게임이 주목 받고 있다. 그러나 게임이라는 분야는 게임의 요소들의 지능이 목표라기 보다는 그 지능들로 인하여 게임이 얼마나 흥미로운가가 목표가 되므로, 학술적인 인공지능과는 다른 특성과 기법이 이용되기도 한다. 그러나 기계에서 지능을 구현하고자 하는 측면에서는 동일한 목표를 갖고 있으며, 현실적으로도 학술적으로 개발된 많은 방법론이 게임에서 이용되고 있다.

학술적으로 인공지능 분야의 국내 수준이 외국에 비해서 결코 낮지 않고, 외국도 인공지능 엔진에 관심을 갖고 본격적인 투자를 시작한 것이 최근의 일이므로, 국내에서도 많은 관심을 갖고 투자한다면 좋은 결과가 있으리라 예상된다.

참고문헌

1. S. J. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, 2002.

2. T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
3. P. Sweetser, "Current AI in Games: A Review", <http://www.itee.uq.edu.au/~penny/Game%20AI%20Review.pdf>
4. <http://www.ai-implant.com>
5. <http://www.directia.com>
6. <http://www.renderware.com>
7. D. Fu, R. Houlette, "Putting AI in entertainment: an AI authoring tool for simulation and games", *Intelligent Systems, IEEE*, Vol. 17, pp. 81-84, 2002.
8. <http://www.symbionic.com>
9. <http://www.biomind.com>
10. <http://www.louderthanabomb.com>
11. <http://fill.sourceforge.net/fcl.htm>
12. <http://www.twilightminds.com>
13. <http://lancet.mit.edu/ga>
14. <http://www.stockai.com/>
15. <http://www.goertzel.org/benzine/WakingUpFromTheEconomyOfDreams.htm>
16. <http://www.planetquake.com/qca/qcstudio/>
17. <http://www.gamers.org/dEngine/quake/spec/quake-spec34/index2.htm>
18. <http://www.pathengine.com>
19. E. F. Anderson, "Playing smart-artificial intelligencce in computer games", *Proc. zfxCON03*, 2003.