

지능형 물리엔진을 이용한 주행 시스템

김대현, 신동일, 신동규

세종대학교 컴퓨터공학과

e-mail : crowlid82@gce.sejong.ac.kr, {dshin, shindk}@sejong.ac.kr

Driving Agent System which uses Intelligent Physics Engine

Tae-hyun Kim, Dongil Shin, Dongkyoo Shin

*Dept. of Computer Engineering, Sejong University

요 약

본 논문에서는 물리엔진 기반으로 지능형 주행 시스템을 구현하여 사용자에게 쾌적한 주행 환경을 제공하여 사용자가 즐거움을 느낄 수 있도록 하는 것에 목적을 두고 있다. 이를 위해 시스템 환경 내에서 사용자의 위치 및 회전각을 추출한 다음 유한 상태 머신과 베이스 이론에 입각하여 구현된 모듈에 입력하여서 분석을 한다. 그리고 모듈에서 나온 분석 결과를 장애물이나 커브 구역에서 제동 함으로서 사용자가 현 위치에 따른 환경을 인지하여 좀 더 나은 주행을 할 수 있도록 하였다. 또한 시스템 내부적으로 분석된 자료가 주행 시스템에 피드백 과정을 통해 입력되어 자동적으로 환경에 적응하여 주행 시간을 단축하는 시스템을 설계 하였고, 일부 기능에 대한 구현을 완료하였다.

1. 서론

근래 다양한 하드웨어 및 소프트웨어 기술의 발달로 인해 컴퓨터 상에서 사용자에게 뛰어난 가상 현실을 제공하는 시뮬레이션 프로그램이나 가상 환경이 많이 선보이고 있으며, 특히 과거와 달리 흔히 물리엔진과 인공지능 엔진 및 그래픽 엔진 등을 결합한 프로그램을 이용하여 제작되고 있다. 이러한 엔진을 사용해서 많은 프로그램들이 실생활에서는 경험 해 볼 수 없는 것을 가상공간을 통해 구성하는 것이 가능해졌고, 그로 인해 가상 공간에서의 경험에 대해 매력을 느끼고 즐기는 사용자들이 점점 증가하는 추세이다. 그러나 앞서 말한 시뮬레이션이나 가상 환경 등의 높은 난이도나 조작인터페이스의 사용은 그런 환경이 익숙치 않은 유저에게는 여전히 높은 진입장벽을 가지고 있으며 그런 유저를 위해 필요한 것이 프로그램 내의 상황에 능동적으로 대처하며 변화하며 조작 및 동작을 할 수 있도록 구현 된 지능형 물리엔진이다. 본 논문에서는 물리엔진으로 구현 한 시뮬레이션 환경을 인지하고 수집한 수치를 이용, 학습하여 사용자가 컨트롤하는 가상 환경 내의 물체가 최적화된 움직임을 보일 수 있도록 하는 데 목표를 두고 있다. 이를 위해 기본적으로 주행 시스템을 구현하여 사용자가 방향 컨트롤에 어려움을 겪을 수 있는 위치에서 제반 서비스를 기반으로 좀 더 쉽게 사용자가 박진감 넘치는 운전조작을 할 수 있도록 조절하는 데 목표를 두고 있다. 또한 몰입도를 높이기 위해 가상 환경 내부적으로도 인공지능 엔진을 구현하여 시간이 지남에 따라 운전 컨트롤이 자동적으로 향상되는 느낌을 가질 수 있도록 할 것이다.

2. 관련 연구

2.1 물리엔진

가상 환경 내의 오브젝트를 대상으로, 질량·속도·마찰·공기 저항 등의 수치를 이용하여 뉴턴 역학 모델을 시뮬레이트하는 프로그램을 통칭한다^[1] '물리엔진(physics engine)'이라는 명칭을 바탕으로 생각해보면, 가상 환경하에 있어 물리 엔진이란 곧 '자연계의 물리 현상을 가상 환경 내에서 흉내 내는(simulate) 프로그램'일 것이다. 그렇다면, 물리 엔진을 이용하여 어떤 종류의 물리 현상을 시뮬레이트하는 것일까? CPU 의 연산 능력에는 한계가 있으므로, 모든 물리 현상을 프로그램 내에서 완벽하게 구현하거나 범지구적인 거대한 자연계의 현상까지 시뮬레이트하는 것은 물론 불가능하지만 뉴턴역학 같은 일정한 법칙을 따르는 현상은 구현이 가능하다. 하나의 시뮬레이션 프로그램 내에는 논리, 렌더링 엔진(rendering engine), 네트워킹, 물리 엔진, 사용자 입력 처리 등의 수많은 프로그램이 결합되어 있다. 물리 엔진은 이중 프로그램 내부에 논리적으로 뉴턴 역학 공간을 구성하고, 이 공간 안에서 일어나는 물리 현상을 시뮬레이트하는 일을 하게 된다.

2.2 인공지능 엔진

시뮬레이션 프로그램의 개발 목적 중 하나가 '즐거움 또는 재미'가 되는 순간부터 AI 는 필연적으로 발생했다. 그 이유는 가상 공간 하에서 '재미'를 느끼는 것 자체가 자신 외의 객체와의 연동에 기반하고

있기 때문이다. 여기서 객체라 함은 자신 이외의 플레이어(사람일수도 있고 AI 일수도 있다), 가상공간의 세계, 가상공간 속에서 인정되는 유효사물(가상 환경 속에서 플레이에 영향을 미칠 수 있는 모든 사물)을 포괄하는 개념이다. 사용자는 가상공간 내에서 유효사물을 이용하여 가상공간 세계가 정의하는 룰에 따라서 객체와의 커뮤니케이션을 통하여 유발될 수 있는 다양한 ‘재미’를 경험하게 되는 것이다. 이렇듯 가상 환경 하에서의 인공지능 객체의 역할은 여러 가지로 중요하다고 할 수 있는데, 그 역할 중 하나로 AI는 프로그램의 보조 역할을 수행한다. 가상 환경 내에서 플레이어의 원활한 조작 및 수행의 보조 역할을 할 수 있도록 하는 것도 AI가 담당한다. 본 논문에서는 사용자에게 물리엔진으로 구성 된 가상 환경 하에서 ‘재미’를 느낄 수 있도록 함과 동시에 가상 환경의 인식을 통해 수집 된 데이터를 이용, 스스로 진화하는 능력을 보여주는 모듈을 구성 함으로써 인공지능의 게임분야에의 응용을 도모 하였다.

2.3 유한상태머신 (FSM: Finite State Machine)

FSM은 유한한 개수의 상태(state)들로 구성된 기계를 말하며, 가장 널리 사용되는 인공지능 처리 방식이다. 유한 상태 머신(FSM)은 장치나 어떤 기술의 순차 로직 혹은 제어 기능에 단순하면서도 정확한 설계를 가능하게 한다. FSM에 대한 아이디어는 유한한 상태(state)들을 가진 전자 제어 기계들로부터 얻어졌다. 여기서 상태(state)라는 말은 하나의 조건이라고 생각하면 된다. FSM의 알고리즘은 간단하지만, 거의 모든 시스템을 상태로 표현할 수 있다. 그렇기 때문에 FSM은 시스템을 제어하는 알고리즘으로 간단하면서 가장 많이 사용된다. FSM은 하나의 입력(일반적으로 조건이 변화하는 것을 의미)을 받고 그에 의거해서 현재 상태에서부터 다른 어떤 상태로 전이(transition)되는 방식으로 이루어져 있다. FSM의 프로그램 알고리즘을 보면 실제로 단순히 switch() 구문 혹은 if 구문을 사용하여 프로그래밍되는 것이 전부이며 그래서 FSM은 특별한 수준의 인공지능 기능을 요구하지 않는 가상 환경에서 많이 사용하며 만약 설계/구현 과정에서 상태의 수가 늘어나게 되면 switch() 구문을 사용하는 대신 배열을 사용하여 원하는 루틴에 빠르게 접근한다.^[2] 본 논문에서는 시뮬레이션 환경 내에서 주어지는 사고나 역주행 같은 다양한 상황에 맞춰서 사용자에게 현 상황에 대해 경고를 주거나 위험한 상태가 감지 됐을 때 최악의 상황을 면할 수 있게끔 사용자의 조작에 인공적으로 제한을 두는 방식으로 제어를 할 수 있도록 할 것이다.

2.4 베이지안 네트워크

베이지안 네트워크는 베이지 규칙을 기반으로 노드로 표현되는 각 변수의 의존 관계를 통해 확률 값을 추론하여 신뢰 값을 구하기 위한 DAG (Directed Acyclic Graph) 모델이다.^[3] 노드들 간의 관계는 방향을 가진 에지를 통해 표현되며, 원인이 되는 부모노드와

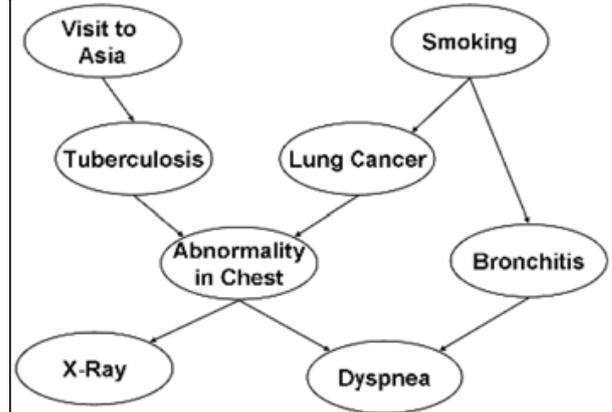
결과가 되는 자식노드로 나뉘어진다. 각 노드는 여러 개의 속성을 가질 수 있고 이들의 합은 1이 된다. 부모를 가진 자식노드들은 의존 관계를 나타내는 확률 테이블을 가지고 있고 부모가 없는 노드들은 초기 확률 값을 갖는다. 부모가 한 개일 때 자식 노드의 확률 값은 다음의 식을 통해서 얻어진다.

$$P(A) = \sum_i P(A|Bi)P(Bi) \tag{1}$$

식에서 B는 A 노드의 부모 노드이고 i는 부모 노드의 상태 수이다. A 노드의 확률 값은 부모 노드가 가진 상태의 확률 값과 자식 노드가 가지고 있는 조건부 확률 값의 곱으로 표현된다. 일반적인 노드들 사이의 확률 분포 $P(x_1, x_2, \dots, x_n)$ 은 다음과 같이 나타난다.

$$P(x_1, x_2, \dots, x_n) = \prod_i P(x_i | Parents(X_i)) \tag{2}$$

그림 1은 베이지안 네트워크의 예를 보여준다.



(그림 1) 아시아 네트워크, Lauritzen, 1988

베이지안 네트워크에서 추론은 크게 예측 방향과 진단 방향으로 나뉘어져 확률 값의 전파로 이루어진다. (그림 1)에서 결핵(Tuberculosis)이나 폐암이 있을 때 흉부에 이상이 올 것을 추론하여 X-레이 결과나 호흡 곤란(Dyspnea) 증상을 예측하는 것이 전자에 속하고, X-레이 결과나 호흡 곤란 증상을 관측함으로써 결핵이나 폐암을 원인으로 추론하는 것이 후자에 속한다. 베이지안 네트워크에서 추론 속도는 얼마나 노드들이 연결되어 있는지, 루프의 수, 증거와 질의 노드 간의 위치 등과 같은 구조적인 요인에 영향을 받는데, 노드 간의 경로가 여러 개 존재할 경우엔 계산이 어려워지기 때문에 근사값을 사용하는 알고리즘을 적용한다. 단일 경로 구조를 갖는 베이지안 네트워크에서는 조건부 독립 조건과 베이지 규칙을 적용하여 추론이 이루어진다.^[3] 두 노드 X, Y의 관계가 X→Y와 같을 때 X가 증거노드이면 Y의 확률값 추론은 확률 테이블을 통해서 이루어지고 Y가 증거노드인 경우엔 X의 확률 값 추론이 베이지 규칙을 적용해서 이루어지는데 베이지 규칙은 다음과 같다.

$$Bel(X) = P(X|Y) = \frac{P(X|Y)P(X)}{P(Y)} = \alpha P(x)\lambda(x) \tag{3}$$

$$\alpha = \frac{1}{P(Y)}, \lambda(x) = P(Y|X)$$

이 때

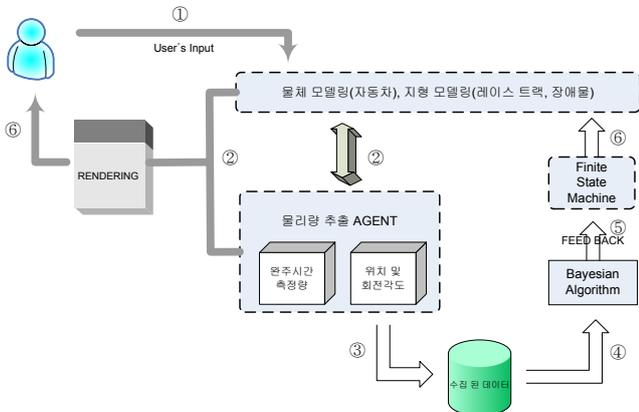
이러한 베イズ 규칙을 사용하는 베이지안 망의 구조 학습으로 널리 쓰이는 것이 K2 알고리즘이다. 이 방법은 결측치가 없는 데이터에서 베이지안 망을 쉽고 간단하게 학습해낸다. 이 방법은 기본적으로 노드의 순서에 의존한다. 노드의 순서란 인과적 순서라고도 하는데 인과적으로 선행하는 사건의 노드가 후행하는 사건의 노드 앞에 놓이는 것을 의미한다. 예를 들어 흡연 여부는 폐암의 원인이 될 가능성이 있지만 폐암은 흡연의 원인이 될 수 없다. 이 경우 흡연을 나타내는 노드는 폐암을 나타내는 노드보다 순서 상으로 앞서게 된다. 이런식으로 베이지안 망의 각 노드(특징을 나타냄)들의 인과적 순서가 주어지면 구조 학습은 훨씬 간단해지게 된다.^[4] 본 논문에서 제안하는 자동차 시뮬레이션을 처리하는 모듈은 이러한 단순한 과정을 반복적으로 거치는 알고리즘을 적용하여 효율적으로 각 위치에 따른 레이싱 트랙에 최적화 된 회전각도를 구해서 제공하도록 한다.

3. 지능형 물리엔진을 사용한 주행 시스템 구조

이 실험에서는 자동차와 같은 물체모델링과 레이싱 트랙 및 장애물 같은 지형모델링을 구현, 각자의 성질에 맞는 속성을 제공한다. 또한 시뮬레이션 환경 내에서 측정 된 정보를 데이터베이스화 시켜서 베이지안 알고리즘을 적용시켜 가장 최적화 된 이동정보를 사용자에게 제공한다.

1) 주행 시스템 실험 과정

우선 실험에서 제공되는 다양한 수치들을 수집하는 모듈을 구성하기 전에 중요한 것은 데이터의 시작과 끝의 범위를 정하는 것이다. 그리고 한 번의 트랙 완주 시의 각각의 데이터를 이용해서 분석하는 부분과 실시간으로 현재까지 수집 된 모든 데이터를 이용해야 하는 부분이 있으므로 데이터를 용도별로 나눠서 저장해야 한다. 다음 그림은 주행 시스템의 구성을 나타내고 있다.



(그림 2) 주행 시스템 실험 과정

① 먼저 응용 프로그램을 통하여 사용자의 힘을 입력 받는다. 이 때 게임에 설정 된 가상 환경에 따라 장애물 등의 물체와 지형의 물리량을 측정하여 현재

자동차의 위치 및 회전각 등의 다양한 값들이 생성된다.

② 생성된 값을 토대로 물체와 지형모델은 실시간으로 변화된 위치와 힘을 렌더링 엔진에 제공하는 한편, 물리량 추출 하는 모듈에서 요구하는 완주시간 및 위치와 회전각도를 전달한다.

③ 물리량 추출 AGENT 에선 수집 된 정보를 데이터베이스에 넘겨준다.

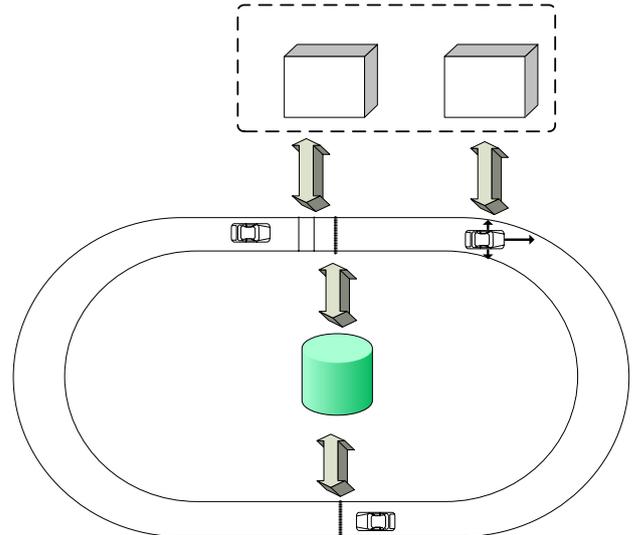
④ 데이터베이스에 수집 된 개인 사용자 별 완주시간과 각 위치에 따른 회전각도 정보를 베이지안 알고리즘을 적용 시켜 현재 사용자의 위치에서 가장 최적화 된 회전각도를 구한다.

⑤ 현 위치에 최적화 된 회전각도의 값을 FSM(유한상태모델)에 전달한다.

⑥ 전달 된 최적화 값을 이용해 FSM 에선 현 상황에 맞는 상태머신을 선택, 물체 모델링에 전달한다.

⑦ FSM(유한상태모델)에서 물체 모델링에 전달 된 상태 값을 렌더링 엔진을 거쳐 화면에 표시하여 사용자에게 제공한다.

2) 지능형 물리엔진 구현 모듈



(그림 3) 지능형 물리엔진 모듈 구현도

위의 구성대로 모듈 코딩의 핵심과 단계를 설명하면 다음과 같다.

① 자동차가 시작점을 지나면서 데이터의 수집이 시작된다. 주행을 하는 동안 자동차의 포지션은 XY 축의 절대 값으로 표현되며, 로테이션(회전 각) 값은 0~360 도를 0~1 까지의 구역으로 환산해서 적용했다. 즉 3.6 도마다 0.01 정도의 값을 할당 받게 되며 이는 코딩 과정에서 직관적으로 이해하기 쉽게 만들기 위해 적용됐다.

② 자동차는 주행을 하는 동안 자동차에 설치 된 센서를 이용해 주변을 탐색하게 된다. 여기서 말하는 센서란 화면에는 보이지 않으면서 충돌체크만 할 수 있도록 기능을 제한시킨 강체를 말한다. 이 부분은 실험에 있어서 중요한데 앞으로 모든 지형의 위치 및 회전각은 사전 정보로서 모듈에 입력된 상태로 시작한다. 그리고 자동차의 이 센서의 충돌을 통해서 지

형과의 거리가 감지되고 충돌이 일어날 경우에만 지형과 자동차의 회전각을 체크하고 비교함으로써 모듈의 수집과 분석을 가능하게 하고 계산 과부하를 줄여 줄 수 있기 때문이다.

③위의 과정을 통해서 수집 된 정보는 텍스트의 형태로 파일에 쓰게 된다. 그리고 자동차가 전체 트랙 길이의 중간에 구현 된 데이터 분석 실시 체크포인트를 지나는 순간부터 모듈의 분석하는 부분에선 지금 까지 수집 된 자동차의 이동경로와 이동경로에 따른 속도의 변화 및 지형과의 회전각 차이를 구하게 되고 트랙의 절반을 지나는 데까지 걸린 측정 시간을 저장한다.

④완주가 한번 끝난 자동차는 두 번째 트랙을 주행할 때부터 자신이 완주한 트랙의 코스에 도달하기 전부터 이전에 주행 한 정보, 즉 추천하는 속도와 회전각을 눈에 잘 들어오도록 표시해 줌으로서 이전보다 더 나은 주행이 되도록 한다. 그리고 내부적으로는 커브 구역에서만 각 위치 별로 회전 각을 수집하여 완주 시간과 그 코스를 지나갈 때의 속도를 기준으로 가중치를 줌으로써 운전자가 지속적으로 더 나은 정보를 받을 수 있도록 구성한다.

⑤데이터가 많이 쌓일수록 주행표시정보는 점점 더 정확해지고 자동차 핸들 자체에도 추천 된 회전각 범위를 크게 벗어나지 않는 크기에서만 핸들링이 가능하도록 유한상태머신을 통해 조작성 함으로서 사고 발생 위험을 미리 줄일 수 있는 지능형 물리엔진의 모습을 보일 것이다.

4. 베이지안 알고리즘을 적용한 실험결과 예측

위 실험에서 제공되는 수치 중에서 회전 각을 선택하는 기준은 장애물과 회전 각이 수평이 될 때 가장 안전한 상태로 인식하며 장애물과 같은 각도일 때를 기준 점으로 삼아서 다음 표와 같은 가중치를 정한다.

<표 1> 회전 각도에 따른 가중치 값

장애물의 각도를 기준	± 0~10°	± 10~30°	± 30~60°	± 60~90°
	0.7	0.2	0.1	0

상기의 표는 실험을 위해 기본적으로 잡은 표이며 반복적인 주행을 통해 어떤 위치에서 특별한 각도가 완주시간에 미친다고 판단되면 그 각도에 가중치를 더하고 다른 각도에서는 가중치가 줄어들 것이다. 위의 표에서는 알기 쉽게 표현했지만 실제 실험에서는 내부적으로 수없이 많은 각도가 추천될 것이며 회전시의 속도와 완주 시간표를 참고한 계산을 통해 가장 완벽에 가까운 근사치를 갖는 속도와 회전 각도 값이 전달되어 화면에 표시되게 될 것이다. 베이지안 알고리즘에 의하여 내부적으로 산출 된 가장 최적화 된 각도를 제공함으로써 사용자는 더 빠른 시간 내에 자동차 코스를 완주할 수 있게 될 것이며 이로 인해, 주행 학습결과와 이동코스의 제공으로 인해 자신에게 가장 적합한 운전습관을 파악하게 된다. 부가적으로는 레이싱 트랙 별로 저장 된 가장 빠른 완주코스를 자동주행 하는 기능을 추가해서 실제 주행에 들어가

기 전에 사용자가 학습모드를 체험할 수 있도록 할 것이다.

5. 결론

본 논문에서는 사용자가 주행 환경에 쉽게 익숙해질 수 있도록 하기 위해 주행 시스템 내에서 최적화된 학습정보를 제공하는 방법을 제안하였다. 또한 물리엔진이 무엇이고 기본적인 물리 반응을 수행하기 위한 이론을 살펴보았다. 그리고 이를 화면에 표시하기 위한 렌더링 엔진의 개요를 살펴 보았고 다음으로 베이지안 네트워크가 어떤 알고리즘을 기반으로 정보를 분석하여 최적화 된 환경을 제공할 수 있는지 살펴보았다. 이를 위해 지능형 물리엔진을 이용해 가상공간과 자동차 및 지형정보를 구성하고 각 사용자의 레이싱트랙 완주정보를 데이터베이스에 입력하는 모듈을 설계하였다. 물론 이러한 지능적 모듈 구현과 충돌 검출과 충돌 반응 등 기본적인 물리 반응의 수행을 범용 물리엔진을 사용하여 주행 시뮬레이션 환경을 충족시킨다 하더라도 실제 현실에 가까운 주행 시스템 구현을 위해서는 경주 트랙이나 차량의 특성, 비나 눈이 오거나 바람이 부는 등의 환경적인 요인 등 고려해야 할 사항은 훨씬 많다. 게다가 더욱 완벽하게 동작하는 주행 시스템을 위해 FuSM(퍼지상태머신)등의 알고리즘을 추가하거나 베이지 이론이 아닌 의사결정나무 등을 통해 주행 시스템의 속도 및 정확도 향상을 꾀할 수도 있다. 향후 그런 환경들을 포함한 여러가지 변수 요소들을 학습하고 반영하는 주행 시스템을 만들어야지만 주행을 즐기는 사용자로 하여금 흥미를 유발하고 쉽게 지루함을 느끼지 못하게 할 수 있을 것이다.

참고문헌

[1] D. H. Eberly, 유채곤 / 차미리 역, “Game physics (가상 환경 물리 바이블),” 사이텍미디어, Nov. 2004, pp. 223-444.
 [2] DeadWizardSite, “가상 환경과 AI(Artificial Intelligence),” available in <http://deadwi.jaram.org/wiki/wikka.php?wakka=CyberCollegeAI12>
 [3] Wikipedia, “Bayesian Network”, available in http://en.wikipedia.org/wiki/Belief_networks
 [4] I.H Witten & Elbe Frank, “DATA MINING”, MORGAN KAUFMANN PUBLISHERS, 2005
 [5] C. Hecker, “Rigid Body Dynamics Part 1-4,” available in <http://www.d6.com/users/checker/index.htm>.
 [6] Steve Rabin 류광 역, “AI GAME PROGRAMMING WISDOM 2”, 정보문화사, 2005. 3. 10