※ 본 아티클은 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

지능적인 실수: AI 코드에 우둔함을 통합하는 방법
(Intelligent Mistakes: How to Incorporate Stupidity Into Your AI Code)

Mick West

2009. 3. 18

http://www.gamasutra.com/view/feature/3947/intelligent_mistakes_how_to_.php

[Neversoft의 공동 설립자인 West는 원래 Game Developer 매거진을 위해 작성한 글에서 "지능형 실수"를 사용하여 AI 게임 상대에 대한 신뢰감이 향상되도록 한다는 여러모로 생각하게 만드는 견해를 제시하고 있다.]

20년 전 나는 첫 번째 상업용 게임(*Steve Davis World Snooker)*을 만들고 있었다. 이 게임은 최초의 스누커/풀 게임 중의 하나로, AI 상대와 플레이를 즐기는 제품이었다.내가 만든 AI 캐릭터는 매우 간단했다. 즉, 포켓에 들어갈 수 있는 최고 가치의 공을 고른 다음 포켓에 넣으면 되었다.

모든 공의 정확한 위치를 알고 있으므로 매번 볼을 포켓에 넣는 것은 아주 쉬웠다. 이것은 최고 레벨 난이도면에서는 좋았지만 쉬운 모드를 위해 AI에 샷에 대한 임의의 각 편차를 부여했다.

프로젝트가 끝날 무렵 클라이언트로부터 AI가 "플레이를 너무 잘 한다"라는 피드백을 받았다. 이 피드백에 당혹해 하며 클라이언트가 전문가 모드가 덜 정확하기를 바라고 있다고 생각했다. 그래서 원하는 대로 변경했다. 그러나 저하된 정확도 때문에 불평을 들었고 다시 AI는 예전의 너무 좋은 플레이 상태로 변경되었다.

결국 클라이언트는 우리 사무실을 방문해서 그들이 의도하는 바를 직접 설명했다. 그들은 "쉬운" 모드로 인해 문제가 발생한다고 생각하고 있음을 알게 되었다.

클라이언트는 컴퓨터가 많은 샷을 놓치는 것은 긍정적으로 보았으나 포지셔널 플레이가 너

무 좋은 편이라고 생각했다. 컴퓨터는 안전하게 플레이하거나 다른 공을 정렬하기 위해 샷 후 백색 공을 편리한 위치에 남겨두는 것처럼 보였다. 클라이언트는 이러한 사항을 변경하기를 원했다.

문제는 이 게임에는 어떤 포지셔널 플레이도 존재하지 않는다는 점이었다. 결국 백색 공은 임의의 위치에 놓이게 되었다. AI는 오브젝트 볼이 포켓 속으로 들어가도록 하기 위해 큐 볼이 오브젝트 볼을 쳐야 하는 위치만 계산했다.

그래서 이동해야 하는 거리에 비례하는 속도로 해당 지점을 향해 큐 볼을 무턱대고 쳤다.

백색 볼이 굴러가는 위치를 계산하지 못했지만 꽤 자주 포켓 속으로 들어갔다.

왜 그것이 문제인가? AI가 임의로 공을 쳤을 때 클라이언트는 왜 “너무 플레이를 잘한다”고 생각했을까?

사람들은 AI 상대를 사람처럼 생각하는 경향이 있다. 사람들은 컴퓨터가 유사한 상황에서 사람들과 동일한 방식으로 사고한다고 생각한다.

볼이 유리한 위치에 놓이게 된 것을 보면 컴퓨터가 그렇게 의도한 것임에 틀림없다고 생각한다.

이러한 생각은 컴퓨터는 모든 위치에서 볼을 넣을 수 있는 능력이 있으므로 컴퓨터의 경우 모든 위치가 다 플레이에 유리할 수 있다는 것으로 과장된다.

따라서 포지셔널 플레이에 대해 걱정할 필요 없이 컴퓨터는 볼을 계속 포켓에 넣을 수 있다는 것이다. 사람의 경우 단일 샷마다 볼을 넣는 것은 불가능하므로 플레이어는 컴퓨터가 포지셔널 플레이를 활용하고 있다고 생각한다.

디자인 또는 코드?

이것은 디자인 문제인가 또는 코드 문제인가? 이상의 문제는 어느 정도까지는 게임의 유형에 따라 달라지고 AI로 컨트롤되는 상대는 플레이어로서 동일한 상황에 처한 사람을 대표하

도록 설정되어 있다.

풀, 체스 또는 포커 등 일대일 게임에서 AI 결정은 대개 퓨어 코드 레벨에서 결정된다. FPS와 같은 일대다 게임에서는 게임 상대가 플레이어 보다 약한 예외적인 경우도 있다.

일반적으로 플레이어는 셀 수 없이 많은 적의 무리에 대항하는 상황에 놓이게 된다. 다른 게임 장르 특히 레이싱 게임에서는 플레이어가 동등한 상대와 겨루게 한다. 여기서는 실제 AI에 대해 체스 AI와 FPS AI의 중간 정도의 성능을 발휘할 수 있기를 바란다.

컴퓨터가 사람 플레이어의 특징을 흉내내야 하는 경우 그 만큼 더 프로그래머의 작업이 많아진다. 체스 게임에서는 대부분의 AI 작업을 프로그래머가 처리한다. 게임 디자이너는 게임 표현에 더 초점을 맞춘다.

FPS에서 기본 코드는 체스 AI보다 매우 간단하다. 상태 변화, 목표 및 기본 행동 등 경로 찾기가 존재한다.

대부분의 행동 콘텐츠는 어떤 형태의 스크립팅과 함께 게임 디자이너를 통해 제공된다. 디자이너는 사람의 행동 특징을 모방한 액션, 목표 및 반응의 코딩에 대한 책임을 갖고 있다. 헤즈 업!

체스와 풀 등 헤즈업 게임에서 컴퓨터는 플레이어 보다 상당히 유리하다. 또한 Fritz와 같은 현대 체스 프로그램에서는 사람 플레이어 보다 상당히 강력하다.

풀 및 스누커 게임에서 컴퓨터가 샷을 놓치지 않도록 프로그래밍할 수 있다. 그러나 사람들은 자신의 스킬에 잘 매치되는 상대와 플레이하기를 원하므로 게임에는 플레이어가 선택할 수 있는 AI 레벨이 존재한다.

AI에 우둔함을 도입하는 가장 간단한 방법은 수행하도록 허용한 컴퓨터의 계산량을 줄이는 것이다. 일반적으로 체스 AI는 이동 방향을 결정할 때 수십억 개의 계산을 수행한다.

더 많은 계산을 수행하면(또한 더 많은 시간을 할애하면) 그 만큼 컴퓨터는 좋은 플레이를 펼칠 수 있다. 수행하는 계산량을 줄이는 경우 컴퓨터는 좋은 플레이를 펼치지 못한다.

이러한 접근법의 문제점은 AI 플레이어의 사실성이 떨어진다는 점이다. 컴퓨터 계산량을 축소하는 경우 AI는 사람들이 결코 저지르지 않는 바보스러운 실수를 하기 시작한다. 게임의 인위적인 특성을 알아차리게 되면 실제 상대와 플레이를 하고 있다는 환상이 깨진다.

개발자는 플레이어가 게임에서 즐거운 경험을 갖기를 바라고 있으며 이를 위해 노력하고 있음을 기억하라. 플레이어가 게임에서 이겼을 때 성취감을 맛보도록 하기 위해 게임 종류와 관계 없이 힘든 도전과제를 해결해냈다고 생각하기를 바란다. 즉, 플레이어를 이기려고 최선을 다하는 상대와 게임을 하였다고 느끼기를 원한다.

컴퓨터 계산량을 축소하여, 이기려고 노력은 하지만 무능하여 현실적이지 못한 게임플레이를 펼치는 AI 상대를 만들어낸다. 그러나 플레이어는 게임의 이면에서 실제로 무슨 일이 일어나고 있는지 관심을 갖고 있는가? AI를 무능하게 만들지 않고 강력한 플레이를 펼치도록 하는 대신 고의로 게임을 내팽개치도록 하면 어떻게 될까?

게임 포기

스포츠에서 게임을 포기하는 것은 자연스럽게 보이지만 고의로 실수를 저질러 게임에 지게 되는 것을 의미한다. 동등한 플레이 상대 또는 이기려고 열심히 노력하는 두 명의 상대 간에 경기를 치러야 한다는 일반적인 생각에 따라 이러한 행동은 플레이어 및 팬으로부터 비난 받을 수 있다.

그러나 컴퓨터 게임에서 동등한 경기를 하는 것은 불가능하다. 이 경기는 사람 대 기계 간의 경쟁이다. 한 쪽은 초당 십억 개의 계산을 수행할 수 있는 이점을 갖고 있고 다른 한 쪽은 강력한 인간 두뇌를 갖고 있다.

대등한 플레이란 착각, 즉 지능적인 실수와 인위적인 우둔함을 도입하여 이것을 개선하고 유지하려는 것에 불과할 뿐이다.

컴퓨터는 즐거운 플레이를 위해 게임을 포기해야 한다. 컴퓨터를 이기는 경우 그것은 착각일 뿐이다. 컴퓨터가 이기게 해 준 것이다. 개발자들은 플레이어가 기분 좋게 이기도록 게임이 진행되기를 바란다.

AI 프로그래머는 이러한 아이디어에 익숙해야 한다. 인위적인 우둔함과 위장된 무능을 조장하면서 게임을 조정하고 있다. 그러나 게임 결과를 미리 결정해 놓지는 않는다.

AI가 게임에 지도록 고의로 설정하지는 않으며 플레이어에게 이길 수 있는 적절한 기회를 제공한다. 플레이어가 플레이를 잘 치르지 못하는 경우 AI는 승리하지만 이 플레이어는 적어도 강한 상대를 만나 이길 뻔했다고 느끼고 한 번 더 싸워야겠다는 생각을 하게 된다.

숨겨진 핸디캡

컴퓨터 체스 전문가인 Steven Lopez(리소스 참고)는 사람 대 사람 간의 체스 게임에서 게임 시작 전 체스판에서 말의 일부를 제거하여 게임 시작 시 초급 플레이어가 유리한 입장에 서게 하는 것을 상급 플레이어가 어느 정도까지 허용할 수 있는지에 관해 기술하고 있다.

게임을 시작할 때 상급 플레이어와 초급 플레이어는 능력의 고저에 따라 플레이를 진행하고 게임은 균형을 잃지 않게 된다. 상급 플레이어는 초급 플레이어에게 기회를 주기 위해 "우둔하게" 플레이를 치를 필요가 없다.

그러나 컴퓨터와 플레이하는 사람은 이상과 같은 방식으로 상대에게 우선권을 주는 것을 원치 않는다. 동일한 스킬 레벨의 AI 상대인 경우 체스판에 말 전체를 놓고서 플레이하는 것을 더 선호한다.

Fritz의 프로그래머는 이러한 문제에 대한 해결책을 생각해냈다. 즉, 사람이 말 또는 위치상의 유리한 입장을 갖게 되는 상황을 플레이어가 활용하도록 의도적으로 설정하는 방식을 발견했다. 사람 플레이어가 유리한 입장을 갖게 되면 AI는 게임에 이기려고 노력한다.

득점이 없는 경우 AI는 상대에 맞추어 플레이를 쉽게 치른다. 실제로는 복잡하고 엄청난 양의 컴퓨팅 작업이 진행되고 있다.

AI의 목표는 "게임에서 이긴다"에서 "게임에 이기려고 노력하는 것처럼 행동하나 사람에게 유리한 입장에서 플레이할 수 있도록 하여 이기려고 노력하게끔 한다"로 변경되었다. AI는 덜 지능적인 것처럼 보이기 위해 보다 지능적이어야 한다.

포커 AI

Left Field의 World Series of Poker에 대한 AI를 프로그래밍할 때 AI 계산은 기본적으로 각 난이도 레벨에서 동일하다.

컴퓨터는 알려진 카드를 바탕으로 이길 확률을 계산하고 베팅 히스토리를 기반으로 상대방의 핸드 스트렝스(hand strength) 추정치를 계산한다. 이길 확률을 이용하여 상대와 같은 패를 낼 비율을 계산하고 이를 통해 그만둘지, 상대방의 패를 보자고 요구할지 또는 더 많이 걸지 여부를 결정한다.

특별한 경우에 대한 규칙과 예외가 다수 존재하지만 이것은 기본 사항이다. AI 플레이어는

동일하고 광범위한 계산을 수행하고 상대와 같은 패를 낼 비율을 계산하기 위해 이밸류에이터(evaluator)를 통해 수 만개의 시뮬레이션 패를 실행한다.

이상의 계산을 수행한 후에만 차별화를 수행한다. 그 시점에서 능숙한 플레이어는 최상의 플레이를 연출하고 서투른 AI 플레이어는 지능적인 실수를 저지른다.

서투른 포커 AI의 경우 지능적인 실수를 저질러야 할 것을 계산한 다음 그렇게 하지는 않는다. 또한 그렇게 하지 않는 한 우둔해 보이지는 않게 된다.

예를 들어 사람 플레이어가 큰 돈을 거는 경우 플레이어의 패가 최상의 패가 될 확률은 75%이고 이 때 지능적인 실수는 카드를 그만두는 것이 될 것이다. AI가 이길 확률이 존재하고 서투른 플레이어를 시뮬레이션하고 있는 중인 경우 플레이어는 이길 확률이 분명하지 않으면 큰 돈을 걸지 않는다.

반대로 플레이어는 기회가 불충분하다고 생각하면 상대방의 패를 보자고 요구한다. 이렇게 하는 것은 당연한 일이고 저능해 보이지 않으면서 AI 플레이어의 강점을 축소할 수 있다.

지능적인 실수는 확률적인 방식으로 이행할 수 있다. 저능한 것처럼 가장한 AI는 플레이어가 패가 센 것처럼 허세부리는 경우 게임을 그만두겠다고 카드를 뒤집지는 않는다.

이러한 상황은 높은 무작위성 포커 게임에서 잘 드러난다. AI가 실제로 실수를 저지르는 경우 플레이어는 개별 상황에 대해 대응하지 않는다.

AI는 수백 개 계산을 수행하고 있으므로 결코 인간과는 달리 우둔한 실수를 저지르지는 않는다. 하지만 AI의 인위적인 저능함을 통해 컴퓨터와 플레이어 간에 동등한 게임이 이루어지며 서투르고 평균 실력을 갖춘 플레이어에게 즐거운 게임을 제공하게 된다.

인위적인 부정확성

풀 및 슈터에서 컴퓨터 AI는 최상의 정확성을 자랑한다. AI 슈터는 귀하가 위치하고 있는 1인치에 대한 10억 분의 1까지 다 파악하고 있으며 모자를 쏘아 벗긴 다음 5마일 정도 날려

보낼 수 있다. 유사하게 풀에서 AI는 모든 볼의 위치를 알고 있고 샷을 수행하기 전 모든 볼이 어디로 굴러갈 지 계산할 수 있다.

스누커 AI를 구현했을 때 이 AI는 투 쿠션으로 모든 볼을 포켓에 넣을 수 있었고 플레이할 때마다 브레이크 샷을 147개나 완벽하게 수행했다(포지셔널 플레이를 수행하지 못해 백색 볼을 넣은 경우는 제외).

AI는 플레이하기에 유쾌한 상대는 아니므로 최고 레벨에서도 정확성을 줄여야 하고 스누커에서 벗어나도록 쿠션 샷을 제한해야 한다.

단순히 AI의 정확성을 줄이는 것만이 게임 플레이를 개선하는 최상의 방책은 아니다. 스누커에서 "포지셔널 플레이"를 발견했을 때 컴퓨터에 유리하게 작용할 임의의 결과가 의도적으로 감지되었다. 볼이 좋은 위치에 놓이거나 포커 AI가 럭키 콜을 수행하여 이긴 경우 불공정하고 속임수를 쓴 게임으로 생각될 수 있다.

AI의 정확성을 줄이는 대신 체스에서 정확성을 증가시키는 것을 제안하고자 한다. 흥미롭고 역동적인 게임을 제공하려면 AI는 게임플레이를 조정하여 플레이어가 이용할 수 있는 상황을 만들어야 한다.

이상은 풀에서 무턱대고 샷을 만들고 큐 볼이 놓이게 될 위치를 고려하지 않는 것이 아니라 AI는 의도적으로 볼을 포켓에 넣지 못하도록 하고 플레이어가 멋진 샷을 만들 수 있는 위치에 큐 볼이 놓여지도록 해야 함을 의미한다.

슈터게임에서 적 외계인은 잠복처에서 임의로 빠져 나와서는 안되고 플레이어가 가까이에 다가가 그들을 치는 경우 때때로 잠복처에서 빠져 나와야 한다. 또한 "우연히" 스스로 사선으로 뛰어들어 게임을 보다 흥미진진하게 만들어야 한다.

럭 오브 드로우(Luck of the Draw)

완벽한 적과 플레이하는 것은 흥미롭지 않다. 그러나 무능한 적과 플레이하는 것도 재미없다. 보다 흥미로운 게임을 만들려면 인위적인 저능함과 지능적인 실수 개념을 도입해야 한다.

지능적인 실수는 AI 부분의 실패로 보여질 수 있으나 실제로는 플레이어에게 흥미로운 게임을 선사하기 위해 신중히 계산된 방식이다. 이러한 지능적인 실수는 플레이어가 여전히 특정 레벨의 스킬을 갖추려고 할 때 도전적인 면을 감안한다.

프로그래머 입장에서 보면 지능적인 실수를 추가하는 것은 AI의 정확성을 줄이는 것보다 더욱 복잡하지만 플레이어에게 보다 재미있는 게임을 제공한다.

리소스

Liden, Lars. "Artificial Stupidity: The Art of Intentional Mistakes," in AI Game Programming Wisdom 2, Charles River Media, 2004.
http://lars.liden.cc/Publications/Downloads/2003_AIWisdom.pdf
Lopez, Steven. "Intelligent Mistakes," Chessbase News, 2005.
http://www.chessbase.com/newsdetail.asp?newsid=2579