3

AI를 이용한 게임 밸런스 방법

㈜엔플루토 ▌오 별

1. 개 요

1.1 게임 밸런스와 개요

서로 다른 게임 내 캐릭터의 PvP(Player Vs. Player)에 대한 능력의 형평성을, 대전에 있어서 게임 밸런스라고 말할 수 있다. 이 글은 인공지능(AI : Artificial Intelligence)과 2007버전 엑셀의 통계도구를 이용하여, 반복적 자동테스트를 통해 게임 밸런스를 맞추는 방법에 대해 설명하고자 한다.

1.2 문제 제기

게임 내 대상물을 테스트하고자 할 때, 변수는 기획에 의해 설정된 변수와 상황변수, 플레이어에 의한 변수 등으로 구분할 수 있다. 기획자에 의해 설정된 변수만 고려한다고 해도 통상적으로 10여 가지를 넘어서는 것이 보통이다. 수십 가지의 변수 효과를 모두 검증하기 위해서 수행해야 하는 테스트는 많은 수의 사람(테스터)과 시간이라는 비용을 동반한다. 이 비용은 상당히 크며, 때로는 실행 불가능한 테스트가 발생하기도하고(너무 많아서), 때로는 정량화되지 못한 결과를 도출할 수도 있다. CBT, OBT 등의 대규모 테스트에서도 통제되지 못한 변수들의 변화에 의해, 부정확한 데이터를 얻거나 해석하지 못하는 경우도 다반사이다. 기획의 의도가 충분히 반영되었는지 확인하지 못하고 게임이 출시될 수 있는 위험에 노출되어 있는 것이다.

이렇듯 우리는 밸런스 테스트를 수행하는데 있어

1) 시간적 제약
2) 비용적 제약
3) 통제 불능에 의한 부정확한 데이터 획득

등의 어려움에 처해 있다.

통제 불능에 의한 부정확한 데이터 획득의 예로써, 대전에 영향을 주는 변수가 게임변수 외에 사람자체가 변수라는 면을 말할 수 있다. 대전에 영향을 주는 변수들을 다음과 같이 2가지 카테고리로 구분해보면,

1) 게임 플레이 환경의 차이 : 게임 내 특정 변수 값, 또는 상수 값들이 다를 경우
2) 서로 다른 실력의 플레이어들에 의한 차이

대전 결과에는 이들 모두가 한데 섞여 나타나기 때문에, 플레이어의 실력을 동일하게 설정하고 대전하는 것은 불가능하다. 그렇게 되면 결과가 게임변수에 의한 것인지, 사람에 의한 것인지를 구분하지 못하고, 올바른 사후조치 또한 어려워진다.

위와 같은 밸런스 테스트의 어려움을 최소화시키면서, 기획의 반영상태를 확인할 수 있는 방법이 필요하다.

1.3 해결책 제안

AI를 이용하여, 개발한 게임에서 AI끼리 자동 대전이 가능하도록 만들면, 게임 플레이에 영향을 주는 변수중 사람에 의한 변수를 제외시킬 수 있다. 그렇게 되면, 기획자에 의해 설정된 게임변수들만의 영향을 명확히 관찰할 수 있다. 또한, 테스트에 소요되는 비용을 절감할 수 있다. 이 방법의 목적은 다음과 같다.

1) 밸런스 테스트에 소요되는 비용의 절감
2) 정량화시킨 결과에 의한 정확한 해석 - 정확한 기획검증
3) 기획 보조 툴 제공
4) 서비스 이전 살펴볼 데이터 항목에 대해 선 점검

1), 2)의 목적은 위에서 언급한 내용과 합치한다. 3), 4)의 목적은 1), 2)가 완수될 경우 부수적으로 발생할 수 있는데, 추가로 기획되는 내용에 대해 사전 검증이 가능해지기 때문이다. 또한, 비슷한 장르의 게임에 대해서는 3)의 역할이 더욱 강력해질 것이다.

2. 본 문

2.1 2D 슈팅 게임에 적용한 사례

예제로 사용된 게임은 "실용적 예제로 본 게임인공지능 프로그램 하기, Programming Game AI by Example/Mat Buckland/사이텍미디어" 책의 Raven 게임

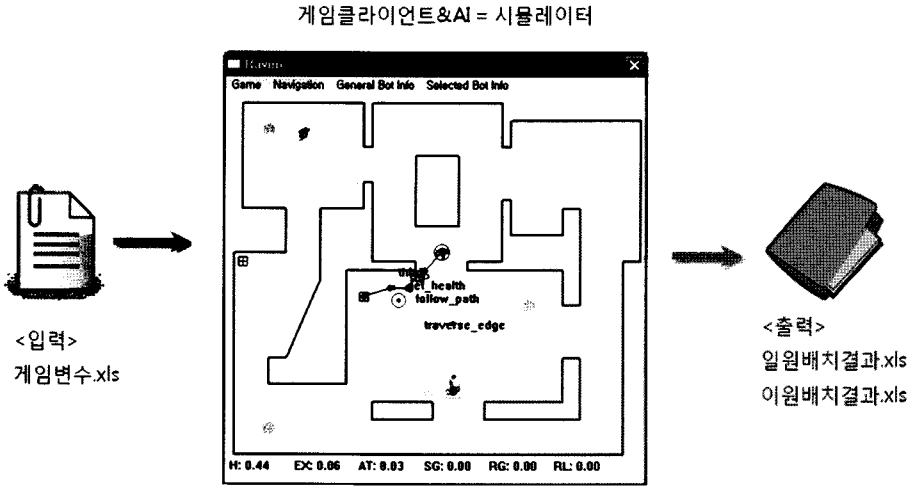


그림 1 게임클라이언트와 AI를 이용한 밸런스 방법 개략도. 실험계획법 적용

이다. 이 책의 예제는 모두 http://www.wordware.com/files/ai에서 다운받을 수 있다.

Raven은 4가지 무기를 사용하는 2D 슈팅게임이다. AI가 게임의 종류에 따라 달라질 것이며, 최대한 플레이어를 닮은 AI를 만든다고 해도 완전한 수준에서 플레이어와 같지는 않지만, 여기에서는 게임의 여러 변수들의 변화가 결과에 얼마만큼의 영향을 미치는지 알아보는 방법을 소개하겠다.

Raven 예제의 Params.lua에 있는 80여 개의 변수를 모두 엑셀파일 npc.xls로 옮기고, 값을 변화시켜, normal bot[1]과 special bot[2]의 속성값을 설정할 수 있도록 되어있다. npc.xls의 특정 변수 행의 check난에 "1"을 표시하면 일원 배치 실험법, 2 이상의 숫자를 2개씩 사용하면 2개씩 짝지어 이원 배치 실험법이 실행된다. 즉, 2, 2라고 두 변수가 함께 check되었거나, 3, 3이라고 두 변수가 함께 check난에 기재되었다면 숫자는 다르지만 각각 짝을 지어 이원 배치 실험법으로 실행된다. 수준수는 경계 값을 포함한 실험 수준을 의미한다. 기본은 2수준이며, 3으로 쓰면 경계값과 등 간격의 사이 값 하나를 포함하여 총 3가지 수준에 대해 반복한다. 반복수는 각 수준에서 반복 실험수이다. 따라서, 수준수가 4이고, 반복수가 30인 일원 배치의 경우, 총 실험수는 120번이며, 이원 배치의 경우에는 480번이다.

npc.xls을 설정하여 저장한 후 Raven 게임을 실행시키면, 설정된 총 반복수만큼 normal bot과 special bot이 대전을 한다. 한번 대전은 상대의 HP[3]가 0이하가 될 때 종료되며, 남은 HP를 측정값으로 했다. 남은 HP를 측정값으로 채택한 이유는 분산분석을 위해 정규성이 있는 측정값을 구하기 위해서이다. 승률 등은 절대값을 사용하기 힘들기 때문에, 절대값으로 사용할 수 있고, 정규성이 있을 것으로 예상되는 "잔여 HP"가 측정값이다. 반복수는 정규성을 가지도록 테스트를 해보면서 조정해보면 좋다. 이 예제에서는 50회는 넘어야 정규분포 비슷하게 나오고, 특정게임에 적용할 때는 반복수를 테스트해보면서 결정하는 것이 좋다.

special bot의 HP를 기준으로, special bot이 이기면 그 때 남은 special bot의 HP를 양수로 기록한다. special bot이 지면, normal bot의 HP를 음수로 기록한다. 만일 같은 종류의 bot이라면, 많은 대전을 통해 얻은 측정값이 정규분포를 이룰 것이라고 기대했다(후반에 살펴보겠지만 실제로 그렇게 된다). "잔여 HP"를 측정해보면, 우리는 특정변수(이동속도, 무기의 공격력, 선택 알고리즘의 변화, 반응속도, 조준정확도, 탄환속도 등)들이 HP로 환산했을 때, 얼마만큼의 효과가 있는지 살펴볼 수 있다.

2.2 결과분석

모든 대전이 종료되면, bot이 멈춘 상태로 Raven 게임에는 더 이상 bot들의 움직임이 관찰되지 않는다. 결과물은 excel 폴더에 result_1.xls, result_2.xls로 출력되며, 각각 일원 배치와 이원 배치의 결과이다. 결과가 나오면 다음의 순서에 따라 분석한다.

엑셀을 이용하려면, 통계도구가 있는 2007 버전부터 가능하다. 엑셀 2007의 데이터>>데이터 분석도구 메뉴를 이용한다. 보통은 분석도구가 등록되어 있지 않기 때문에 사용자가 추가로 지정해야 할 것이다.

1) narmal bot : 기준 속성값을 가지는 AI
2) special bot : 테스트하고자 하는 속성값을 바꾼 AI
3) HP : Health Point. 게임에서 캐릭터가 가지고 있는 생명력

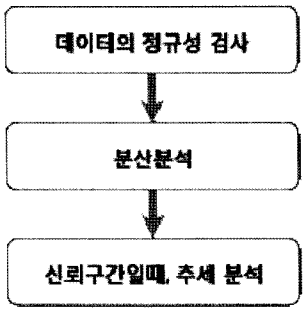
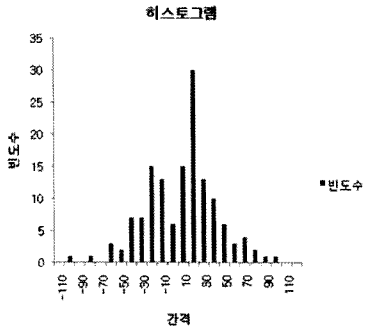


그림 2 엑셀을 이용한 분석 순서

그림 2와 같이 3단계로 나누어 분석하는데, 측정값인 잔여 HP가 실제로 정규성을 가지는지(정규분포 형태인지) 검증하고, 정규성이 인정되면, 분산분석을 통해 평균에 차이가 있다고 말할 수 있는지 살펴본다. 설정한 유의확률에 대해 평균이 다르다고 인정되면, 평균값들의 변화에 대해 추세분석을 하고 사이 값들의 결과를 예측한다(interpolation).

2.2.1 측정값에 대한 정규성 검사

엑셀의 데이터 ›› 데이터 분석도구에서 히스토그램과 기술통계법을 이용하여, 측정값 140개에 대해 아래와 같이 분석하였다.



기술통계법	
평균	5.235714
표준 오차	2.864221
중앙값	10.5
최빈값	20
표준 편차	33.88992
분산	1148.527
첨도	0.382792
왜도	-0.20159
범위	192
최소값	-101
최대값	91
합	733
관측수	140
신뢰 수준(95.0%)	5.663074

그림 3 측정값에 대한 정규성 검사. 히스토그램과 기술통계법 결과

표 1 엑셀에서 일원배치 결과의 예. P값이 0.05보다 작으면 평균의 차이가 의미 있다

분산 분석: 일원 배치법

요약표

인자의 수준	관측수	합	평균	분산
Column 1	30	-199	-6.63333	1110.24
Column 2	30	485	16.16667	978.1437
Column 3	30	850	28.33333	1668.782
Column 4	30	795	30.57692	642.9738

분산 분석

변동의 요인	제곱합	자유도	제곱 평균	F 비	P-값	F 기각치
처리	25408.36	3	8469.454	7.58668	0.000115	2.685643
잔차	125032.1	112	1116.358			
계	150440.5	115				

표 2 엑셀에서 이원배치 결과의 예. P값이 0.05보다 작으면 평균의 차이가 의미 있다

분산 분석 : 반복있는 이원 배치법

변동의 요인	제곱합	자유도	제곱 평균	F 비	P-값	F 기각치
인자 A(행)	183576.9	2	91788.45	123.2883	7.05E-34	3.048833
인자 B(열)	2390.633	2	1195.317	1.605524	0.203796	3.048833
교호작용	1888.267	4	472.0667	0.63407	0.638859	2.424502
잔차	127310	171	744.5026			
계	315165.8	179				

2.2.2 분산분석

정규성 검사에 의해 정규분포를 따를 것이라고 판단되면, 분산분석을 이용하여 정량적 통계분석을 한다. 즉, 해당변수의 차이에 의해 HP의 결과차이가 나타나는가를 알아본다.

2.2.3 추세분석

분산분석에 의해 유의확률 0.05에 대해 평균값의 차이가 있다라고 결론이 나면, 자동 테스트에 의해 확인하지 않은 영역에 대해서는 추세분석으로 유추할 수 있다. 변수의 변화에 의한 HP의 차이를 추정하는 것이다.

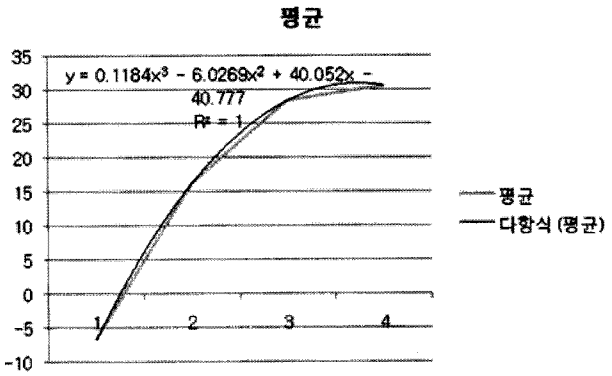


그림 4 추세분석의 예. 분산분석의 유의확률 P값이 0.05보다 작으면 수준별 평균으로 추세를 분석한다

3. 결론과 향후 과제

3.1 요약

지금까지 간단한 2D 슈팅게임 예제를 이용하여 실험계획법과 AI를 이용한 밸런스 테스트 자동화 방법에 대해서 살펴보았다. 게임의 AI가 사람과 유사하게 플레이 한다고 가정할 경우, AI를 이용하여 사람간 실력 차에 의한 변수를 제외시킨 상태에서, 게임변수들만의 차이로 인한 대전 결과차이를 HP라는 정량적인 수로 검증해볼 수 있었다.

또한, 1번 대전에 1분 이하의 시간이 소요되기 때문에, 수 백 번의 대전에 수시간만 소요된다. 이와 같은 방식으로 반복 테스트에 소요되는 비용과 시간을 절감할 수 있었다.

결론적으로 게임개발에 있어서 AI는 다음과 같은 장점을 가진다.

- 게임 컨텐츠로의 효용성 : 특성(또는 성능)이 다른 AI들
- 플레이어의 학습을 위한 도구 : 저 수준부터 고급 수준까지 단계적으로 배치
- 밸런스테스트를 위한 도구

3.2 향후 방향

AI를 이용한 밸런스 방법은 어디까지나 AI가 사람 수준의 플레이가 가능하다는 가정에서 출발한 것이다. 그러나, 게임마다 AI의 성능이 다를 수 있고, 게임이 복잡해질수록 AI의 성능이 사람 수준에 미치지 못할 수 있다. 보통, 게임에서 사용되는 AI의 지능은 다음과 같이 구분이 가능하다.

표 3 게임 분류에 의해 분류된 변수들

분류기준	고려해야 하는 기본변수
solution searching	해집합의 크기 N에 대한 고찰과 반응 능력
control-based	입력단위시간 T, 유저의 반응시간 Tr, 유저의 판단시간 Ts, 반응의 정확도, 입력의 빈도, 시스템 리액션타임 등
experience/ exploring	다양한 경우에 대한 기억과 활용 능력
probability-based	확률과 기대값, 편차 등에 대한 대응 능력

모든 분류기준에 적합한 하나의 일반화된 AI를 제공할 수는 없을 것이다. 그러나, 특정 게임에 특화된 AI를 개발하여 제시된 방법론의 정확성을 높이는 것이 향후 과제이며, 특정 변수를 변화시켰을 때, AI 대전에서는 나타나지 않는 차이가 실제로 사람에 의해서는 나타날 수 있는 부분을 항상 관찰하고 보완하여야 한다.

또한, 밸런스 테스트에 사용된 AI 모델에 대한 실력 수준을 플레이에 비교하여 정량화시킬 필요가 있다. 대전 밸런스를 맞췄다고 하더라도 특정 실력 수준에만 해당하기 때문이다. 이는 사내테스트나 CBT 등에서, 래더시스템[4]을 이용하여, AI가 실제 사람과의 대전을 통해 래더점수[5]를 부여 받고, 래더점수로 AI의 실력수준을 평가할 수 있다.

그림 5와 같이 래더시스템에 의한 래더점수별 사람들의 분포는 정규분포를 이룬다. 여기에서 실험에 사

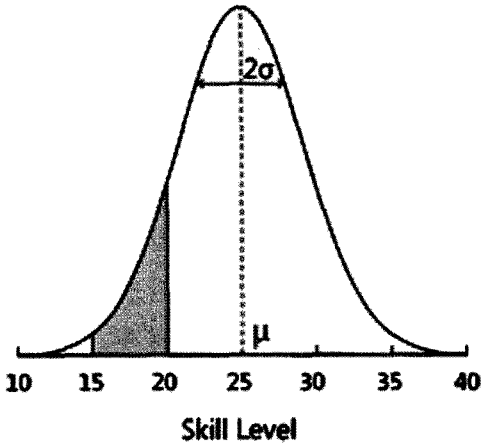


그림 5 MicroSoft사의 TrueSkill System

용한 AI가 어느 수준인가를 정량화시켜 파악한다.

사내에서 개발되고 있는 특정게임에 위와 같은 방법론을 적용하고 있으며, 사내에서 개발자들에게 AI에 대해 새로운 인식의 계기가 되고 있다고 판단한다.

감사의 글

이 글에서 사용된 예제 프로그램은 같은 팀에서 근무하는 3D 클라이언트 개발자인 김주호님이 만들어 주셨다. 그리고 실제 진행중인 프로젝트에도 같은 방식을 적용하여 가능성을 검토하였다. 김주호님의 수고에 감사드린다.

참고문헌

[1] Mat Buckland, Programming Game AI by Example, Wordware Publications, 2005.

[2] 박성현, 최병철, SPSS와 SAS 분석을 통한 실험계획법의 이해, 민영사, 2005.

[3] ELO rating system, http://en.wikipedia.org/wiki/Elo_rating_system

[4] MicroSoft TrueSkill Ranking System, http://research.microsoft.com/en-us/projects/trueskill/

[5] 정보과학회지 투고규정, http://www.kiise.or.kr/collections/soc_mt_02.asp

오 별

연세대학교 물리학 석사
㈜NHN 게임밸런스지원팀장
현 ㈜엔플루토 프로젝트분석팀장
E-mail : byouloh@npluto.com

4) 래더시스템 : 랭킹시스템의 하나로, ELO에 의해 제안된 통계적 랭킹 측정 방식

5) 래더점수 : 래더시스템에서 랭킹을 측정하는 점수