



※ 본 아티클은 CMP MEDIA LLC와의
라이선스 계약에 의해 국문으로
제공됩니다

게임 메커니즘 분석 (Anatomy of a Game Mechanic)

테일러 시그만(Tyler Sigman)

가마수트라 등록일(2009. 07. 29)

http://www.gamasutra.com/view/feature/4091/anatomy_of_a_game_mechanic.php?print=1

[게임 설계 분야의 전문가 시그만(Sigman)은 게임 메커니즘이 시각적으로 어떻게 표현되는지 그리고 이러한 매혹적인 그래프를 통해 비디오 게임이 훌륭하게 개선되는 방법과 관련해 우리가 배울 점은 무엇인지 깊이 있게 분석한다.]

디자인 팀 간에 게임의 특성에 대해 논의하거나 이러한 특성을 프로그래머에게 전달하는 것과 상관없이 적절한 전문 용어가 부족하면 메시지의 내용이 불명확하게 된다. (“불명확”이라는 단어를 사용하는 방법이 메시지를 불명확하게 할 수 있다.)

컴퓨터 프로그래머는 기술적인 배경 지식을 가지고 있고 이들 대부분은 어려운 대학 수학 강의를 수강한 경험이 있다. 그렇기 때문에, 게임 설계자가 원하는 메커니즘을 설명할 때 잘못된 용어를 사용하게 되면 전달하려는 메시지 내용이 전달되지 못할 수 있다.

수학의 기초는 대부분의 게임 메커니즘을 이해하기 위한 편리한 토대가 된다. 그렇기 때문에, 정립된 수학적 용어 및 기호를 사용해 게임 메커니즘을 설명하는 것은 일반적으로 당연한 일이다.

본 글의 첫 부분은 게임 메커니즘의 시각적 표현에 대한 짧은 소개이고, 이러한 표현을 설명하기 위해 적절한 용어를 사용한다. 후반부는 설명에 도움이 되기 위해 더욱 상세하게 게임 메커니즘에 대한 몇 가지 예를 들어 논하도록 하겠다.

그래프의 시각적 가치

백문이 불여일견이다.

그렇지 않다. 확실히 그렇지 않다. 나는 시각적으로 게임 메커니즘에 관해 논하는 가치를 표현하기 위해 더욱 간결한 방법으로 위트있게 생각하려고 했지만 사회적인 지식만 머릿속에 맴돌 뿐이었다.

시각적인 방법을 사용하는데 있어서의 장애는 이를 설명하기 위해 여전히 방법이 필요하다는 것이다. 그렇기에, 결국 수 천 단어로 이루어진 설명이 필요하다. 그래서, 몇 가지 기본 항목을 통해 깊이 생각해보고 난 후 재미있는 소재에 대해 논해보자.

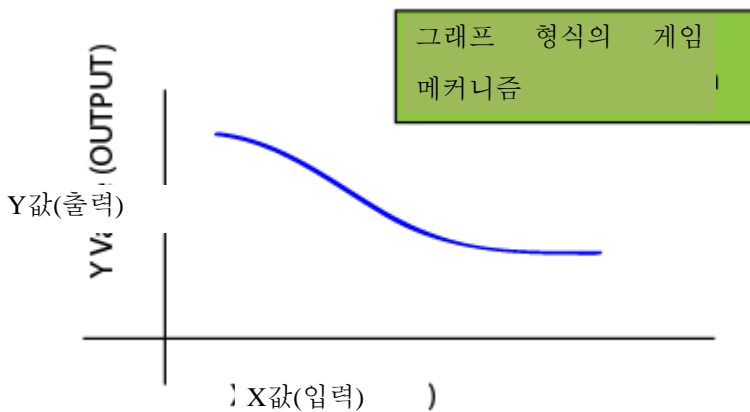
용어: 함수

수학적으로 게임 메커니즘은 보통 함수로 말할 수 있다. 함수는 수학적으로 “블랙 박스”와 같다. 특정 값을 입력하면 블랙 박스(게임 메커니즘)는 값을 출력한다.



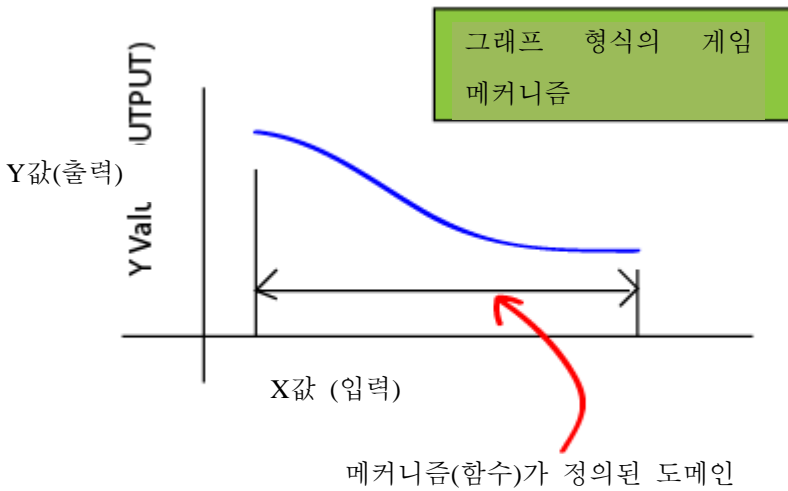
게임 메커니즘은 함수(블랙 박스)다.

그래프상에서 함수는 X-Y 좌표에서 직선 또는 곡선으로 나타난다.



함수

용어: 도메인



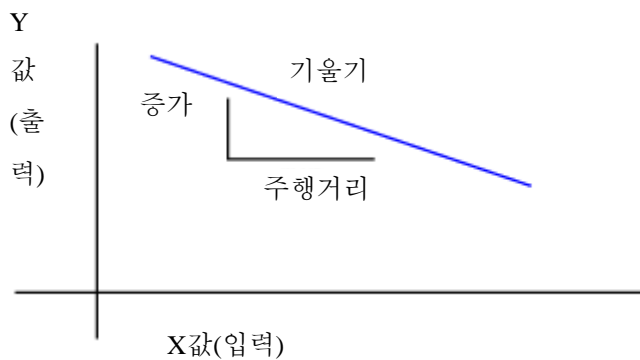
함수 도메인

게임 메커니즘의 도메인은 게임 메커니즘이 활성화될 때의 값 범위를 나타낸다. 그래프상에서 이는 X 축(“횡좌표”)으로 표현된다.

게임 메커니즘: "소총 스킬에 대한 함수로서의 사격 정확도"의 예를 들어보자. 여기서 도메인은 소총 스킬의 범위이다. 1-10 또는 1-100 또는 게임 시스템에 따라 무엇이든 값이 될 수 있다.

또 다른 예는 게임 메커니즘 "트랜스미션 기어에 대한 함수로서의 가속도"가 있다. 이 경우, 도메인은 차량의 기어 범위를 나타낸다. 1 ~ 5 가 될 수 있다.

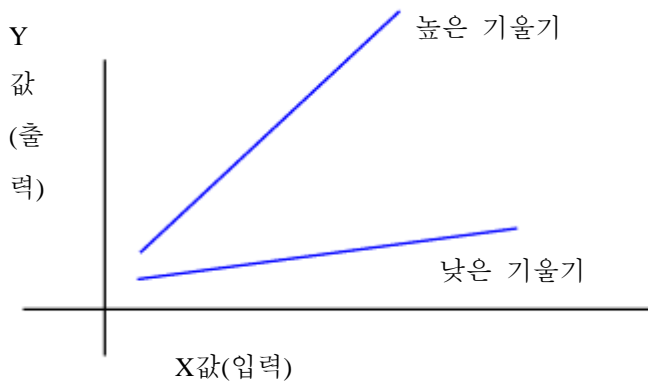
용어: 기울기



기울기

기울기는 그래프 메커니즘의 선형 또는 곡선의 각도를 의미한다. 기술적으로, 기울기는 X 값의 주어진 변화에 대한 Y 값의 변화인 “주행거리에 대한 증가치”로 설명할 수 있다. 마이너스 기울기는 아래로 향하고, 플러스 기울기는 위로 향한다.

개념적으로, 높은 기울기는 **빠르게 변화하는** 메커니즘이고, 낮은 기울기는 **느리게 변화하는** 메커니즘이다.

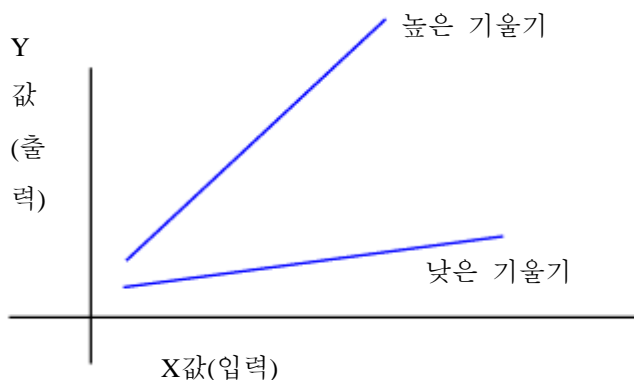


높고 낮은 기울기

이 메커니즘은 선형/비선형(항후 참조) 여부와 상관없이 국부적인 기울기를 여전히 정의할 수 있다. 선형 메커니즘의 경우 기울기는 일정하지만, 비선형 메커니즘의 경우는 일정하지 않다.

용어: 선형

가장 유용한 게임 메커니즘(소위 말해, 함수)의 기술적인 특성 중 하나는 선형/비선형의 여부이다. 선형은 말 그대로 선과 같다("선과 같은 유형"). 수학적으로 풀어보면 게임 메커니즘의 **기울기**는 **지속적**이다. 실질적으로 게임 메커니즘은 주어진 속도로 지속적으로 증가하거나 감소한다.



선형 메커니즘/함수

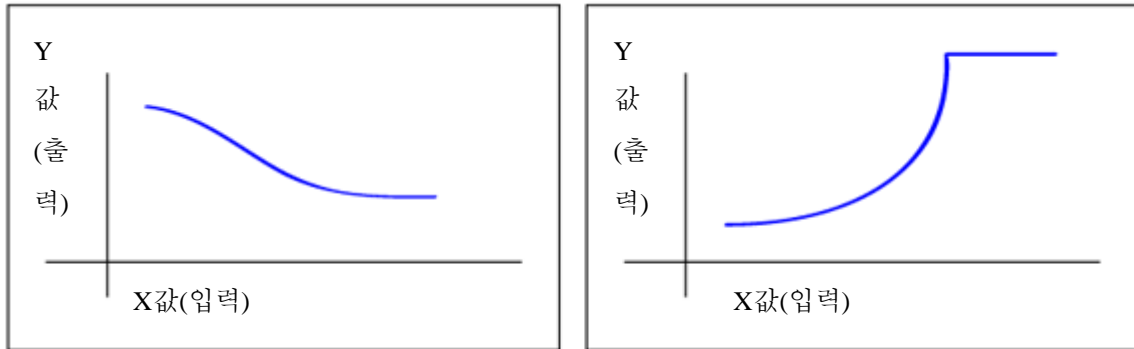
선형 메커니즘의 예

- ✧ **단순한 조향:** 자동차 핸들의 위치는 선형으로 선회율(turn rate)에 해당한다.
- ✧ **충전 시간에 대한 함수로서의 단순한 무기 피해:** 2.0 초 동안 충전하는 경우 무기 피해는 1.0 초 동안 충전하는 경우에 비해 2 배 많다.
- ✧ **무기 스킬에 대한 함수로서의 단순한 공격 기회:** 게임 플레이어가 무기 스킬을 50 만큼 가지고 있으면 25 를 가지고 있는 게임 플레이어보다 공격할 수 있는 기회가 2 배 더 많다.
- ✧ **몬스터 레벨에 대한 함수로서의 단순한 XP 보상:** 레벨 4 몬스터를 패배시키면 레벨 2 몬스터를 패배시키는 것보다 2 배 더 많은 경험을 할 수 있다.

용어: 비선형

메커니즘이 선형이 아닌 경우 이는 *비선형*이다.

매 선형 메커니즘에 대해 비선형 메커니즘도 존재할 수 있다. 그렇게 많지는 않지만 중요한 점은 비선형 메커니즘은 그래도 *매우* 일반적이라는 것이다. 종종 선형 메커니즘이 정확하게 당신의 디자인 의도를 시뮬레이션 하거나 구현하지 않기 때문에 비선형 메커니즘을 사용해야 한다.



비선형 메커니즘/함수

비선형 게임 메커니즘의 예

- ❖ 지금까지 생성된 대부분의 RPG 캐릭터 레벨 진화 XP 테이블(이에 대해 **故** 개리 가이랙스(Gary Gygax) 또는 데이브 아네슨(Dave Arneson)에 감사해야 한다.): 레벨 1 에서 레벨 2 로 올라가기 위해서는 오직 1,000XP 가 필요할 수 있지만 레벨 20 에서 21 로 올라가기 위해서는 1,000,000 XP 가 필요할 수 있다.
- ❖ **문명 사회에서의 인구 증가율**: 멸망하는 사회가 아니라면 500 년 된 사회는 일반적으로 100 년 된 사회의 인구에 비해 5 배 이상 많다.
- ❖ **드라이빙 게임에서 “도로에서의 초”에 대한 함수로서의 차량 가속도**: 실제 차량은 낮은 기어에서 더 높은 가속도를 내고 높은 기어에서는 더 낮은 가속도를 낸다. 6 초 동안 0-60 으로 주행할 수 있지만 그 다음 6 초부터는 또 다른 60MPH 를 얻는다는 예상을 할 수 없다.
- ❖ **기하학 전쟁(Geometry Wars)에서 시간에 대한 함수로서의 스코어**: 20 분 동안 게임을 했다면 10 분 동안 게임을 한 사람과 비교해 스코어보다 2 배 이상 더 많게 된다.

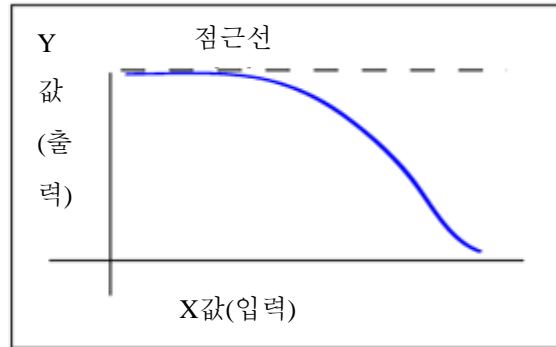
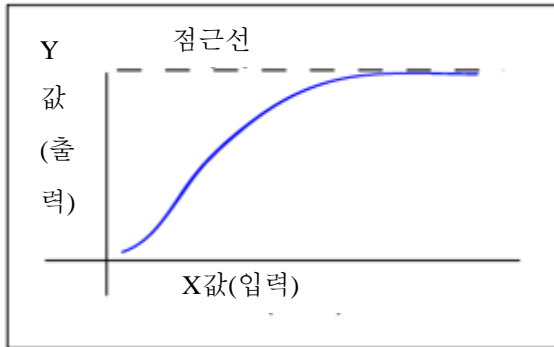
비선형 메커니즘은 우리와 매우 친근하며, 이를 설명하는 또 다른 구어적인 방법이 있다. 예를 들면, 다음과 같다.

- ❖ "수익 체감의 법칙"
- ❖ "기하학적인 증가"

비선형 메커니즘의 유형

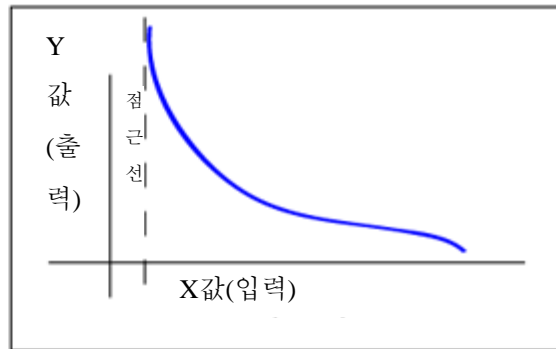
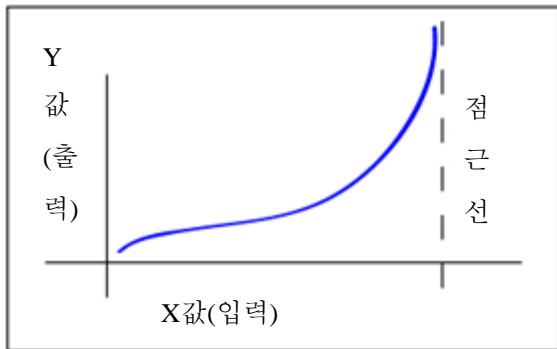
선형 메커니즘은 기울기뿐 아니라 여러 측면에서 비선형 메커니즘과 다르다. 다음과 같이 비선형 메커니즘을 더욱 세분화할 필요가 있다.

- 값에 대한 점근:** 이러한 메커니즘은 일정한 결과(y 값)에 접근할 때 점점 “평평해 지는” 경향이 있다. 한 점에서 X 에 대한 큰 변화는 Y 값에 대한 매우 작은(하찮은) 변화를 야기한다. 다른 말로 표현하면, 일정 지점을 지나면 0 에 가까워 진다는 것이다.
 - 메커니즘 예:** 무기 스킬을 기반으로 공격할 가능성: 게임 플레이어의 무기 스킬이 증가할 때 공격을 위한 게임 플레이어의 가능성을 결과적으로 절정에 이르도록 하는 전투 시스템을 설계할 수 있다. 한 지점을 지난 후 무기 스킬이 추가적으로 증가하는 것은 영향을 거의 미치지 않는다.



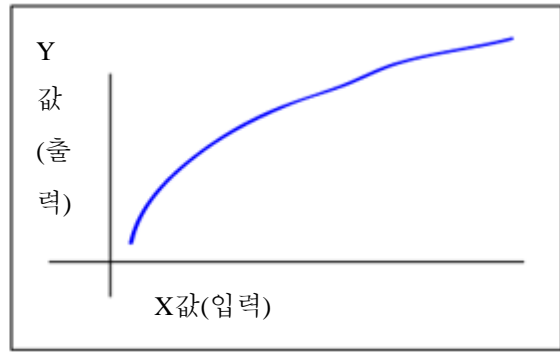
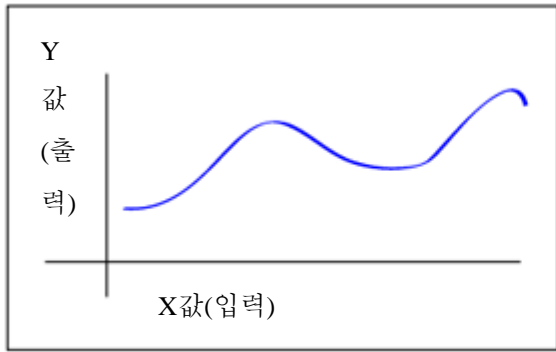
값에 대한 점근

- 무한대로의 점근:** 이러한 메커니즘은 X 가 증가하거나 감소할 때 무한대(Y)로 접근하는 경향이 있다.
 - 예:** 인구 증가율(기하학적인 증가). 기술적으로 일부 한계는 결과적으로 한계 자체를 벗어나려고 한다. 그러나, 그 한계까지 메커니즘은 점근법에서 무한대로의 관계를 따를 경우도 있다.



점근에서 무한대

비점근: 비선형 메커니즘은 “평평해 지지” 않고 점근에서 무한대로 나가지 않는다.

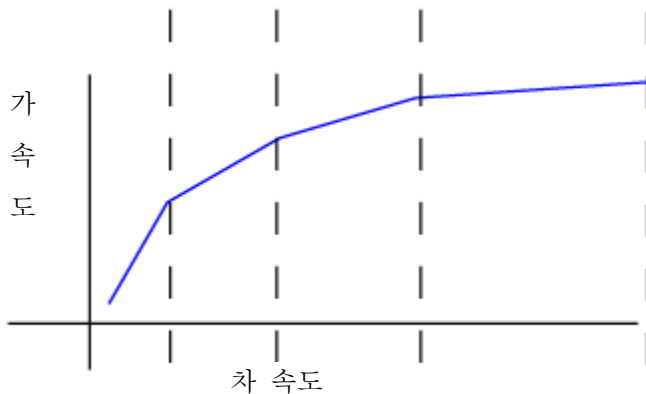


비점근

분할된 선형

분할된 선형 메커니즘은 기술적으로 비선형이지만 서로 다른 기울기로 이루어진 두 개 이상의 부분으로 구성된 선형 메커니즘처럼 생각할 수 있다.

분할된 선형 메커니즘의 예는 실질적으로 단순한 차량의 변속기 메커니즘이다. 이러한 메커니즘에서 속도 함수로서의 가속도와 토크를 모형화한다. 각 기어는 서로 다른 값을 가질 수 있다. 결과 그래프는 여러 상이한 선형 “기어 대역”으로 이루어졌고, 각각의 기울기는 서로 다르다.



분할된 선형 메커니즘: 독특한 선형 분할 통보

분할된 선형 메커니즘은 상대적으로 생성되기 쉽고 복잡한 비선형 메커니즘을 시뮬레이션 하기에 아주 좋은 방법이다. 이 메커니즘이 다양한 핵심 포인트간에는 선형이라고 가정하면 가기에 좋은 방법이고 조절하기도 매우 쉽다.

선형 및 비선형 메커니즘의 혼재

당신이 초콜릿을 좋아하고 바닐라도 좋아한다면 이 둘을 합쳐서 두 개 모두 가질 수 있다. 인앤아웃버거(In and Out Burger)는 이런 문제로 당신이 고민하지 않도록 한다. 게임 메커니즘을 설계할 때 도메인의 일부가 선형이고 다른 일부는 비선형이도록 할 수 없는 이유는 존재하지 않는다.

그러나, 매우 좋은 이유가 있지 않는 한 이러한 복잡함을 피하려고 한다(다음 페이지 참조).

올바른 메커니즘을 선택할 때 주의할 사항: 선형 또는 비선형 또는 이 모두?

게임 메커니즘을 설계하는 것은 과학이고 예술이다. 일반적으로 단순함과 정교함의 균형을 잘 잡아야 한다. 모든 요인은 설계에 대한 결정에 중점을 두어야 한다. 원하는 시뮬레이션의 정확도는 어떠한가? 모형화하려는 메커니즘에 대해 얼마나 잘 알고 있는가? 조정하고 재조정하고 균형을 맞추기 위해 얼마나 단순화해야 하는가? 이를 작업하기 위해 다른 팀원들은 이를 잘 이해하고 있는가? 최종 사용자(게임 플레이어)가 이를 잘 이해하고 있는가? 등이 있다.

시그만(Sigman) 설계 법칙 41a: 당신의 설계 목표를 달성하는 가장 단순한 메커니즘을 선택하라.

제발 간단 명료하게 하라. AI 프로그래밍에 대한 매우 간편한 이치와 유사하다. 당신의 목표는 가장 명확하고/진보적이고/혁신적인 시스템을 만드는 것이 아니라 최소한의 필요한 작업을 통해 설계 목표를 달성하는 것이다. 다른 말로 표현하면, 당신의 게임 메커니즘은 필요한 만큼만 기술적이면 된다. 선과 같지만 사실이다.

그렇다고 항상 단순해야 한다는 말은 아니다. 종종 설계 목표는 **복잡한 시스템을 통해 달성되기도 한다**. 그러나, 복잡함을 피할 수 있을 정도로 단순해라. 왜냐하면, 조정, 조절, 반복, 이해하는 것이 훨씬 용이하기 때문이다.

이행의 단순함을 위해

1. 선형 메커니즘
2. 단일 방정식(예를 들어, $y = x^2$)으로 정의한 비선형 메커니즘
3. 분할된 선형 메커니즘
4. 다중 방정식과 혼합된 선형/비선형 메커니즘으로 정의한 비선형 메커니즘

그러나, 구현 복잡성과 게임 플레이어가 보거나 경험하는 복잡함을 혼동하지는 말아야 한다!

시그만(Sigman) 설계 법칙 41b: 메커니즘 구현의 복잡성은 메커니즘을 이해하는 게임 플레이어의 복잡성과 항상 같지 않다.

비선형 메커니즘은 선형 메커니즘과 비교해 구현하기에 좀 더 복잡하지만 그렇다고 이해하기 어렵다는 것은 아니다. 앞에서 언급한 "수익 체감의 법칙" 같은 개념은 많은 사람들에게 친근하다. 그 이유는 일상 생활에서 수많은 상이한 메커니즘을 만나기 때문이고, 이들의 상당수는 비선형 메커니즘이기 때문이다.

기어를 변경할 때 차량의 가속도가 0-60 주행이거나 토끼가 기하급수적으로 늘거나 당신의 재정 투자금이 해당 기간에 수많은 방법으로 처리될 수 있다는 사실을 이해하기는 어렵지 않다. 사람들은 매일 그들 주위에서 벌어지고 있는 메커니즘에 대해 생각조차 하지 않아도 일상에서 벌어지는 메커니즘을 관찰하고 있다.

당신이 시뮬레이션을 만들거나 아케이드 액션 게임을 만든다면 실생활의 메커니즘을 연구하는 것은 게임 설계에 아주 중요하다.

일상 생활에서의 몇 가지 선형 메커니즘

주행 속도 및 거리: 직선 도로에서 1 시간 동안 120mph 로 주행하는 경우 당신은 동일한 조건에서 60mph 로 주행하는 상황과 비교할 때 2 배 더 많이 갈 수 있다. 물론 당연한 소리다. 그러나, 이것은 당신이 적용할 수 있는 선형 메커니즘 중 하나이다!

질량에 대한 함수로서의 운동에너지: 2 파운드의 무게로 초당 50 피드 속도로 날아가는 물체는 동일한 속도지만 무게가 4 파운드인 물체의 운동에너지의 1/2 수준이다.

대류권에서의 온도: 대류권까지의 표면 온도가 서서히 줄어든다("감율").

일상 생활에서의 몇 가지 비선형 메커니즘

US 연방 세금: US 세금(세율, 단계적인 공제 폐지, 기타 요인)과 관련된 수많은 요인으로 10 만 달러를 버는 사람은 5 만 달러를 버는 사람과 비교해 세금을 2 배만큼 지불하지 않는다. 그러나, 당신의 세금 환급 범위 내에서 마지막 10 달러를 번 것에 대한 세금은 첫 번째로 10 달러를 번 것보다 더 많은 세금을 낸다(두 번째의 한계 세율에 해당함을 전제). 세율의 핵심 시스템은 분할된 선형 메커니즘이다.

웨이트 트레이닝: 매 시간마다 당신은 힘을 키우기 위한 트레이닝을 하기 때문에 동일한 수준의 힘을 얻지 않는다. 종종 빠르게 얻기도 하지만 때로는 한동안 안정된 상태에 머물기도 하다(비선형 감소).

인구 증가율: 인류의 역사를 10 년씩 분할해서 보게 되면 새로운 생명이 태어나 비선형 형태로 증가한다(비선형 증가).

복리: 당신이 20 살 때 1,000 달러를 5% MMA 계좌에 투자한 경우 60 살이 되면 7,040 달러가 될 것이다. 40 살 때 동일한 이율로 1,000 달러로 투자한 경우 60 살이 되면 2,653 달러가 될 것이다(37.7% 수준). 복리 계산이 선형 메커니즘이었다면 20 년 동안 투자한 금액은 40 년 동안 투자한 금액보다 정확히 50% 수준이 되었을 것이다(비선형 증가).

경험 년 수에 대한 함수로서의 연봉(게임 개발자 연봉 설문조사(Game Developer Salary Survey)를 고려): 15 년 동안 근무한 사람은 5 년 동안 근무한 사람이 경험한 것과 비교해 3 배 많이 경험했다고 볼 수 없다(비선형 감소)

중력: 두 물체간의 인력은 거리에 따라 비선형으로 감소한다. 당신이 해수면에서 200 파운드(지구의 중심으로부터 4,000 마일 이하)라면 당신의 중력은 고도가 2 배 높은 곳에서는 50%가 채 되지 않는다.

속도에 대한 함수로서의 운동에너지: 50mph 속도로 움직이는 5 파운드의 물체는 100 mph 로 움직일 때보다 운동에너지가 4 배 높다(비선형 증가).

데이터의 일반적인("중형"으로 알려진) 분산(통계 문서 참조<링크>): 상위 10% 대역의 값은 중간 10% 대역의 값보다 훨씬 적다.

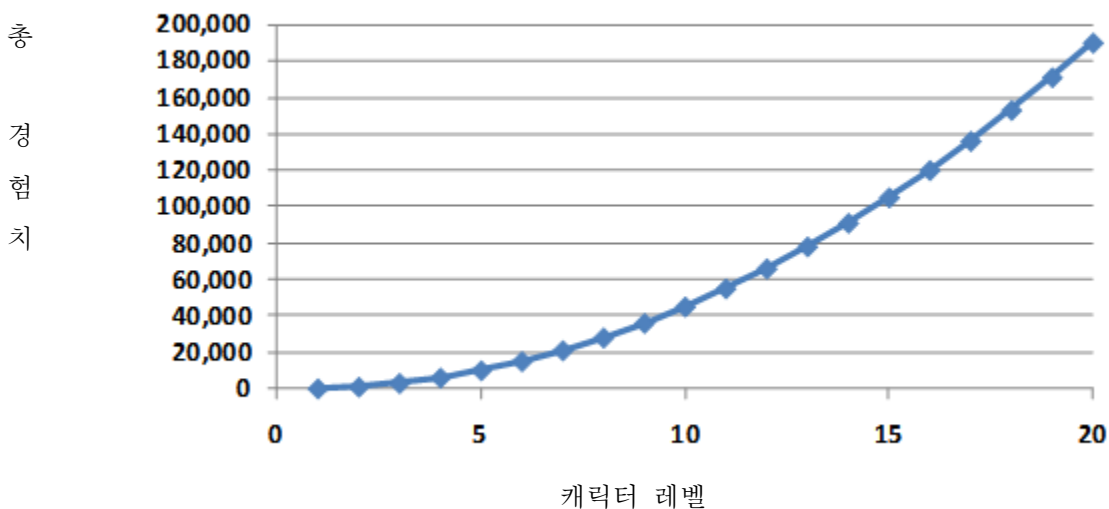
실생활에 적용되는 경우는 많다. 실제 게임 메커니즘에 대해 재빨리 몇 가지 경우를 알아보자.

게임 메커니즘 #1: 일반적인 롤 플레이 게임에서의 레벨링 테이블

빠른 설명: 캐릭터가 미리 정의한 핵심 특성 및 스킬 향상에 해당하는 “레벨”을 얻는다. 이는 레벨을 얻기 위해 다양한 태스크를 완료해서 얻는 일정한 경험치(Experience Point)를 얻게 된다. 향후 레벨은 초기 레벨보다 높기 위해 훨씬 많은 경험치를 소비한다.

그래프 표현!

레벨별 총 경험치



일반적인 경험치/레벨 차트

상세 분석!

경험치 시스템과 관련해 문제를 해결하기 위한 두 가지 방법이 있다. 방법 1은 훨씬 인기있는 방법으로 위에 나타난 것과 같이 경험치 곡선을 사용하는 것이다. 다음 레벨로 올라가기 위해 경험은 비선형으로 증가한다. 레벨 2로 올라가기 위해서는 오직 1,000이 필요할 수 있지만 레벨 20에 도달하기 위해서는 약 200,000이 필요하게 된다.

이 시스템은 모든 RPG 게임의 증조할아버지 격인 던전 앤 드래곤(Dungeons and Dragons)에서 사용했다. 이 시스템의 이점은 좀 더 큰 몬스터가 게임 플레이어에게 더 큰 XP 값을 제공하기 때문에 애플 투 애플(apples to apples)을 기반으로 몬스터를 비교하는 것은 매우 쉽다. 또한, 게임 플레이어는 좀 더 큰 리워드를 얻는 성취감을 느끼게 된다.

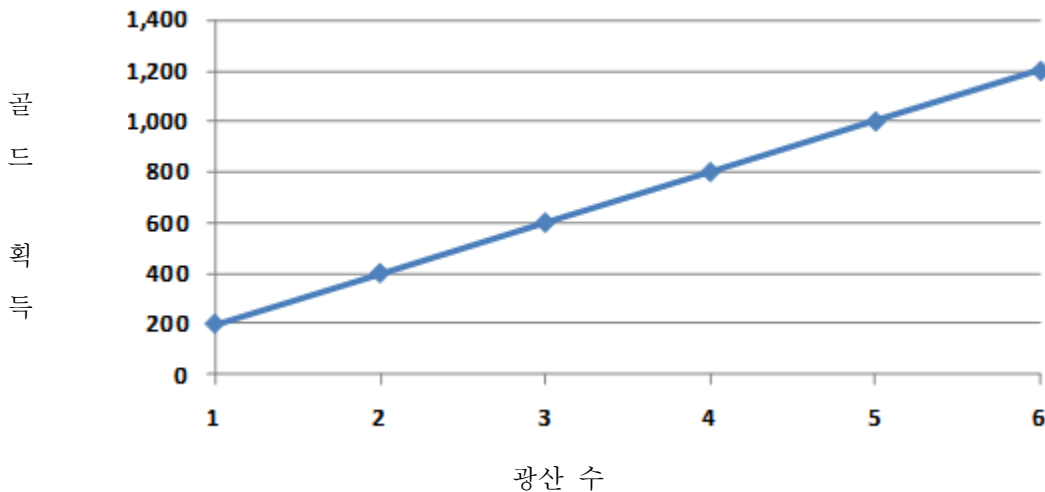
완전함을 위해 XP/레벨에 대한 다른 일반적인 시스템은 지속적으로 레벨이 증가(레벨별 1,000 XP)하지만 게임 플레이어와 관련이 될 몬스터에 대한 XP 리워드를 조정한다. 예를 들어, 레벨 1 캐릭터의 코볼드를 무찌르면 50 XP를 얻을 수 있지만 레벨 10 캐릭터의 동일한 코볼드를 무찌르면 5XP를 얻게 된다.

게임 메커니즘 #2: *에이지 오브 엠파이어(Age of Empires)*: *에이지 오브 킹즈(the Age of Kings)* DS의 골드 획득

빠른 설명: 이 게임에서 게임 플레이어는 각 광산에서 200 개의 골드를 얻는다. 3 개의 광산을 가지고 있으면 1 개의 광산을 가지고 있을 때보다 수입이 정확히 3 배($3 \times 200 = 600$ 개 골드) 더 많다. 기술 보너스를 세지 않는다.

그래프 표현!

광산 수에 대한 골드 획득



에이지 오브 엠파이어: 에이지 오브 킹즈 DS의 광산 수에 대한 함수로서의 골드 획득

상세 분석!

이 수입 메커니즘은 완전히 선형이다. 농장이 많으면 생산되는 음식의 양도 많아 이는 직접적으로 비례한다.

이 경우 게임 설계자가 선형 메커니즘으로 선택할거라고 생각하나?

이것은 간단한 경우로 원하는 경제적인 시뮬레이션을 얻는 가장 간단한 방법이라고 나는 확실하게 생각*하고 있다. 즉, 비선형 메커니즘을 가지기 위한 확실한 원인은 존재하지 않는다. 2 개의 광산이 1 개의 광산보다 2 배 많이 생산한다는 것은 논리적이며 당연하게 예상할 수 있는 결과이다. 게임이 핵심 전략과 전술에 초점을 두는 상황에서 경제의 복잡성에 대한 2 차 효과를 모형화 하는데 강제적인 정당화가 존재하지 않았다.

(게임 설계자는 효율성 메커니즘을 포함해서 통합된 광산 네트워크가 증가할 때 효율성이 더 많거나 더 적게 시뮬레이션 할 수 있다. 그러나, 잦은 충돌에 대한 본 게임이 왜 지루한가? 복잡함이 필요하지 않은 곳에 복잡함을 추가했을 수도 있다.)

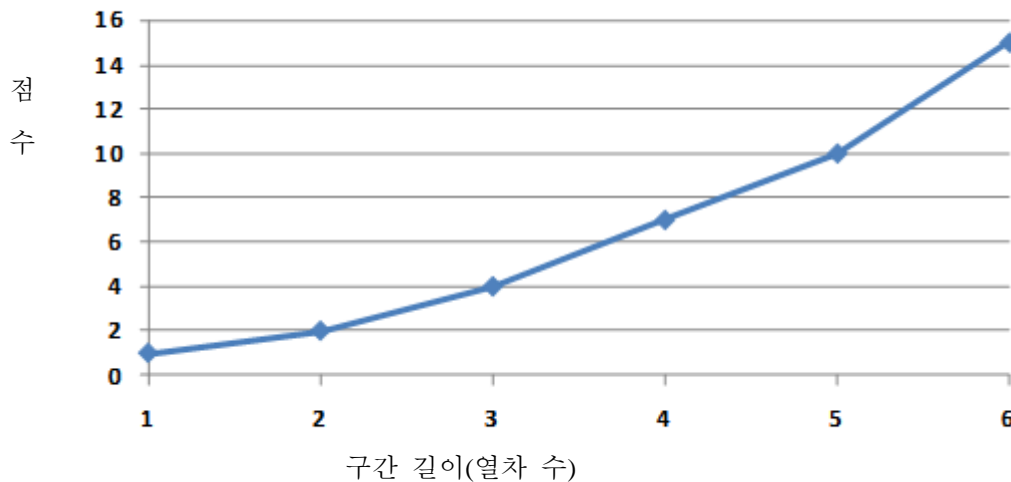
**내가 이러한 특정 시스템을 설계했기 때문에 내 생각은 사실이다.*

게임 메커니즘 #3: *티켓 투 라이드(Ticket to Ride)*의 경로에 대한 승리 포인트

빠른 설명: 이 게임에서 다른 도시로 이동시 열차로 이동하는 경우 승리 포인트를 얻게 된다. 열차 경로는 경로 길이를 지정하여 열차 컬러 카드를 매칭하여 얻게 된다(예를 들면, 4 개의 붉은 열차 카드는 네 개의 붉은 열차 경로를 요구한다). 승리 포인트는 구간 길이에 대해 비선형으로 얻는다. 6 경로 길이에는 15 포인트를 제공하지만 3 경로 길이에는 오직 4 포인트만 제공한다.

그래프 표현!

경로 길이에 대한 점수



티켓 두 라이드의 서로 다른 구간 길이에 대한 포인트 획득

상세 분석!

다시 한번, 우리는 비선형 메커니즘에 대해 세밀하게 살펴보자. 6 구간 길이를 요청하는 것은 3 구간 길이를 요청하는 것보다 2 배 이상 가치가 있다. 게임 설계자가 이를 왜 비선형 메커니즘으로 선택했다고 생각하는가?

많은 메커니즘이 예술적인 목적으로 선택하지만 이런 경우 나는 비선형 곡선을 사용하는 것만이 실질적으로 유효한 결정이라고 생각한다. 메커니즘이 선형인 경우 게임의 즐거움이 덜하기 때문이다.

6 구간 길이가 3 구간 길이에 비해 또는 4 구간 길이가 2 구간 길이에 비해 2 배만 더 필요하다면 게임 플레이어는 단순히 목적지까지 연결된 구간이 필요하다는 것을 제외하고는 더 긴 구간을 요청하기 위한 인센티브가 존재하지 않는다.

그러나, 경로 길이를 얻는데 비선형적으로 승리 포인트를 얻게 되면 게임 설계자는 추가적인 동기 부여를 할 수 있게 된다. 주어진 컬러의 카드를 절약하고 긴 구간 중 한 구간을 요청하는 경우 이것은 큰 수익이 될 수 있다. 이는 게임 내에서 기회와 결정을 창출한다. 게임 플레이어는 경로가 좋을 때마다 카드를 단순히 소비하는 것이 아니라 자신의 열차 카드를 보유하고 있는 이유이다. 이것은 결국 전략 게임이다.

(2 구간 길이는 1 구간 길이의 포인트보다 정확히 두 배 많다. 그러나, 2 구간은 1 구간보다 확실히 어렵지 않기 때문에 이 경우 비선형 단계는 불필요한 것 같았다.)

게임 메커니즘 #4: 이브 온라인(Eve Online)의 우주선에 대한 보험 정책

보험에 대해 얘기하는 것을 좋아하는 사람이 있을까? 그러나 이것은 게임이다. 그렇기 때문에 논할 가치가 있다. 여기서 말하는 부서진 메커니즘은 전투에서의 부서진 메커니즘과 크게 다르지 않기 때문에 논할 가치가 있는 것이다.

빠른 설명: 이브 온라인에서 당신은 당신의 우주선에 대한 교체 보험을 구매할 수 있다. 만약 이 우주선들이 정당한 이유로 파괴되었다면 보험 회사는 당신의 우주선의 가치와 당신이 구입한 보험의 수준에 따라 ISK(Interstellar Credits)를 지급한다.

그래프 표현!

보험 정책 선택 화면의 예는 다음과 같다.

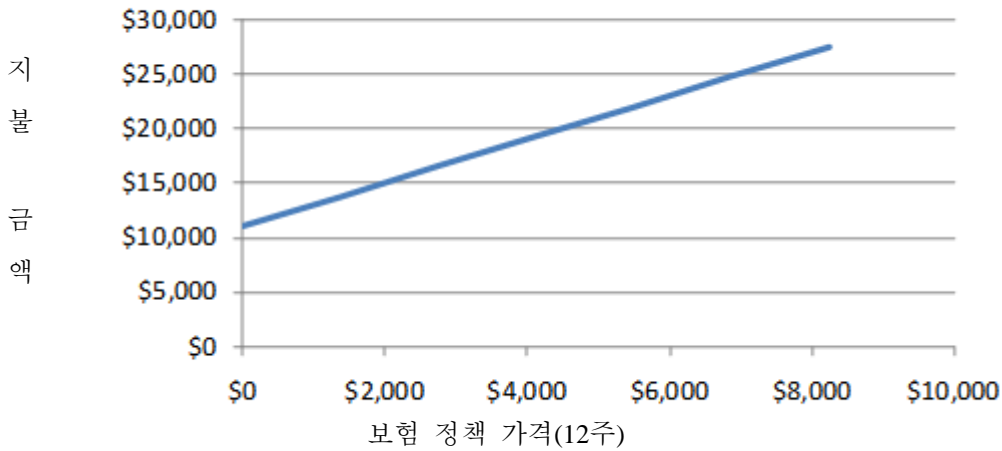


이브 온라인(Eve Online)에서 우주선에 대한 보험 정책 선택 사항

무료 보험은 우주선 가치의 40%를 보상한다.

이러한 값을 사용해서 그래프를 그려보자. 이 경우, "도메인"(X 값)은 보험의 구입가를 나타내고, 지불 금액은 Y 값으로 나타낸다. 게임 메커니즘 또는 함수는 보험 구입가와 지불금간의 비공개적인 연관성을 나타낸다.

이브 온라인 보험 정책



이브 온라인(Eve Online)에서 우주선에 대한 보험 선택 사항을 나타낸 그래프

상세 분석!

보다시피 메커니즘은 **선형 메커니즘**이다. 0 원을 지급하는 것은 우주선 가치의 40%를 보험에 드는 것과 같다. 이 우주선에 대해 8,250 을 지급하는 것은 우주선 가치의 99%를 보험에 드는 것과 같다.

이 경우 왜 게임 개발자들이 선형 메커니즘을 선택했을까? 비선형 보험 일정과 비교해 장점과 단점은 무엇인가?

우주선 보험 정책은 12 주 후 만료한다. 만료 시간이 보험 정책 가격 구성에 어떠한 영향을 미치는가?

본 문서 마지막으로의 점근적 접근

선형과 비선형은 일상적으로 내가 처리하고 있는 게임 시스템의 핵심 설계 요소이다. 단순히 곡선을 선택하면 제어 방식에서 트레저 드롭까지 모든 방면에서 큰 차이점을 보인다.

당신이 좋아하는 메커니즘이나 당신만의 설계 전쟁터에서 마주치게 되는 팁을 공유해라!

게임 제목과 이미지는 해당 소유자의 등록상표거나 저작권이다.

제목 사진: 케리(Kerri) 2009. 크리에이티브 커먼즈(Creative Commons) 라이선스 하에서 사용.