한국게임학회 논문지
제7권 제3호 2007년 9월

캐릭터간의 대전 시스템을 위한 밸런스 구현

박찬일°, 양해술
청강문화산업대학 컴퓨터게임과°, 호서대학교 벤처전문대학원 컴퓨터응용기술학과
cipark@ck.ac.kr°, hsyang@office.hoseo.ac.kr

Balance for Fighting System between Characters

Chan-Il Park°, Hae-Sool Yang
Dept. of Computer Game, Chungkang College of Cultural Industries°, Dept. of Computer Science and Application, Hoseo Graduate School of Venture

요 약

게임 밸런스는 게임 디자인의 매우 중요한 요소이며, 게임을 좀 더 재미있게 구성하기 위한 필수 요소이다. 대부분의 게임 개발사들은 성공적인 게임 개발을 위하여 자체적인 밸런스 방법을 사용한다. 효율적인 밸런스 방법이 항상 성공적인 게임 개발을 보장할 수는 없으나 게임 개발자의 입장에서 효율적인 게임 개발을 위하여 표준적인 밸런스 방법이 필요한 것이 사실이다. 캐릭터간의 전투 시스템은 주로 액션을 포함한 장르에서 구현된다. 캐릭터간의 전투를 위한 직접적인 밸런스 요소는 각 캐릭터들이 가지게 되는 특성 및 공격 형태들이다. 본 논문에서 우리는 캐릭터간의 대전 시스템을 갖는 게임에서 캐릭터와 관련된 직접적인 밸런스 요소를 구분하고 이에 근간한 수학적 밸런스 방법을 제안한다.

ABSTRACT

Game balance is an very important factor for game design, it is necessary factor to make game more fun. Most of game companies have been using their own balance method for successful game development. An efficient balance method always can't guarantee successful game development but it is true for game developer to need standard balance method for efficient game development. Fighting system between characters is mainly implemented in genre including action. Direct balance factors for fighting between characters are their characteristics and attack patterns. In this paper, We identify direct balance factors for game including fighting system between characters and propose mathematical method for it.

Keyword : game balance, balance factor, character balance

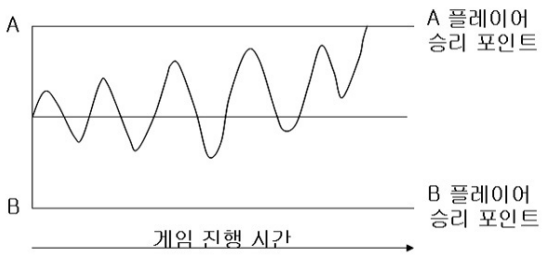
1. 서 론

게임 밸런스를 맞춘다는 것은 게임을 완성하는데 있어 매우 중요한 요소이다. 게임 밸런스는 게임 디자인의 핵심이며, 게임 시작점에서 승리할 수 있는 기회를 공평하게 제공하는 기초적인 조건이다[1].

다양한 재미요소를 가지고 있음에도 불구하고 게임의 밸런스를 맞추지 못하여 시장에서 외면되는 경우도 있었다[2]. 특히 유명 게임에서 새로운 패치의 영향으로 게임 밸런스에 부정적으로 영향을 미쳤던 요소는 많은 사용자들에게 반향을 불러 일으켰던 예도 있다. 게임 회사는 이러한 이유로 밸런스를 맞추기 위하여 상당한 자원을 투자하고 있으며, 이런 차원에서 게임의 밸런스를 맞추는 방법을 정형화하여 시스템적으로 접근 할 수 있다면 이는 매우 좋은 게임 개발 방향을 정립할 수 있다고 판단된다.

애쉬론즈 콜(Asheron' c call2)에서는 게임 개발자들이 수치해석 기술을 이용하여 밸런스를 수행하였으며, 마이크로소프트 액셀과 확률을 처리하는 플러그인(Plugin) Palisade's @Risk를 이용한 위험분석(Risk Analysis) 방법을 이용하기도 하였는데 이러한 방법은 개발자로 하여금 주요 게임 시스템을 모델화하고 분석하는 것이 가능하게 했다[3].

게임 디자이너들은 캐릭터간의 전투 시스템에서 하나의 유용한 성과가 하위의 지속적인 성과를 보다 쉽게 만들어 주는 개념의 포지티브 피드백을 이용하여 〈그림 1〉와 같은 이상적인 게임 진행과정을 구현하기 위한 게임 밸런스에 노력하였다. A와 B는 플레이어이고 각 영역 끝 부분 선은 승리 기준선을 의미한다[4].



[그림 1] 이상적인 게임 진행 상황

이상적인 게임 밸런스를 위하여 〈그림 1〉에서 플레이어 A,B의 승리 포인트인 Y축의 격차가 너무 크다면, 실제 플레이어는 승리하기까지 너무 많은 시간을 자신에게 유리하게 이끌고 유지하여야 한다. 이는 플레이어에게 게임의 지루함을 느끼게 할 수 있다. 또한 승리 포인트까지 도달하는 게임 플레이 시간인 X축 시간이 너무 길면 대전 시간이 길어지는 문제점이 있다. 승리까지 도달하기 위하여 각 플레이어의 유리함을 나타내는 파장의 개수 및 폭도 플레이어로 하여금 재미를 느끼게 구성하여야 한다.

캐릭터 상호간의 대전 시스템 밸런스는 위의 〈그림1〉에서 보여 주듯 x, y축의 시간적 길이 그리고 파장의 개수 및 폭의 넓이를 찾는 것이다. 게임의 밸런스는 보상과 위험도 사이의 자연적인 규형을 유지하는 것[5]으로 그래프상 Y축 파장 개수와 폭을 조정하는 개념이다. 이러한 값들은 게임의 전투 상황 등에 따라 매우 유동적으로 달라진다.

캐릭터간의 경험에 의한 능력차가 현저히 발생하면 쉽게 대전이 종료 될 수 있고, 아이템 획득 또는 맵 오브젝트에 의한 캐릭터의 공격 및 방어 능력의 변화로 인하여 플레이 시간이 변화한다.

빈번한 아이템의 잘못된 사용은 게임 밸런스를 무너뜨리는 현상을 만들 확률이 높다[6]. 이는 전체 밸런스 조정 실패를 의미한다. 아이템이 캐릭터에 적용된 후 밸런스는 역시 동등한 값을 갖도록 조정되어야 한다.

RPG에서는 게임의 전개 방향에 따라 각 대전이 유동적이며, 액션 대전 게임에서는 게임의 매 단계마다 동등한 값을 가정한다.

게임 밸런스의 기본적인 디자인 과정은 게임의 내부 경제구조[7][8]를 구성하는 것으로 기본적인 규칙과 기술을 전제로 구성되는데, 재미있고, 플레이하고 싶은 상태를 유지하게 하는 것을 기본 목표로 한다. 이러한 거시적인 조정을 "Macrocalibration"이라 하고 이 단계가 완성되면 "Microcalibration"단계로 넘어가 미세 조정하게 된다[9].

거시적 조정 단계는 수학적 규형을 맞추는데 중점이 주어지며, 미시적 단계는 세부적인 게임 진행에 따른 요소들을 조정한다. 대전 시스템의 밸런스 조정은 우선 독립적인 대전을 전제로 조정하는데, 이는 위의 거시적 밸런스처럼 수치적 규형을 조정하는 단계이다. 거시적 단계에서의 수치적 밸런스는 전체 게임 밸런스의 중심을 찾는 핵심적인 부분이다.

많은 게임 개발사들 내에서 게임 밸런스는 개발자들의 경험이나 내부 비공개 방법들에 의하여 이루어져 매우 어려운 요소로 인식되었다. 게임 장르가 매우 다양하여 게임 밸런스를 전부 정형적인 방법에 의하여 시스템적으로 절차를

구성한다는 것은 매우 어려운 문제이다.

게임 밸런스는 모든 게임에서 독창성이 보장되나 다양한 게임에 적용될 수 없다는 시각이 있다. 그러나 캐릭터가 가지는 기본적인 요소들은 대부분 게임에서 공통적으로 활용된다. 캐릭터간의 대전 시스템을 가지고 있는 게임은 캐릭터 요소를 바탕으로 공통적으로 활용할 수 있는 효율적인 시스템을 구축할 수 있다.

효율적인 시스템이 성공적인 게임을 제작하기 위하여 필수 불가결한 것은 아니나 전체 게임을 개발함에 있어서 자원의 활용을 최적화할 수 있다. 정형화된 게임 밸런스 방법을 게임에 공통적으로 적용할 경우 기본 시스템의 오류도 방지할 수 있다.

본 논문에서는 대전 시스템을 위한 캐릭터에 관련된 직접적인 밸런스 요소들을 구분하고 액션 및 RPG 등의 대전 시스템에 공통적으로 적용될 수 있는 거시적 밸런스를 위한 수학적 수식을 제공하여 정형화된 밸런스 방법을 제안한다.

2. 대전 시스템의 게임 밸런스 요소

RPG의 밸런스 요소는 매우 다양하며 〈그림 2〉와 같은 RPG 요소들은 캐릭터의 기본적인 특성을 나타내는 종족적인 측면과 종족 내에서 직업 및 신분 등을 구분하는 클래스, 게임을 하면서 얻게 되는 플레이어의 기본 능력과 스킬 등의 향상 요소, 스킬을 높이기 위한 비용과 시간, 게임을 진행하기 위한 자원, 플레이어가 접하게 되는 환경적인 요소, 그리고 전투 시스템 등에서 직접 나타나는 플레이어 대 플레이어의 대립 요소가 있다.

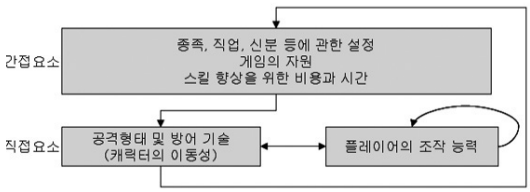
[그림 2] RPG의 밸런스 요소

RPG 밸런스는 위 각 요소들을 같은 수준에서 고려한다. 즉 위 그림에서 보여 주듯 각 요소들이 수평적인 연관관계를 가지게 되고 이를 바탕으로 밸런스를 수행한다. 이러한 점은 전체 RPG를 디자인하기 위한 거시적 방법이다.

게임 밸런스는 게임 규칙의 상호 작용과 관련이 있는 정적 요소와 시간의 흐름에 따른 게임 상의 모든 상호작용에 관련된 동적 요소가 있다. 정적 밸런스는 게임 상의 캐릭터 또는 객체들의 기본적인 요소를 추출하여 비교하며, 동적 밸런스는 실험과 경험을 바탕으로 요소들의 연관성을 도출하여 게임에 고유한 모델을 만들어 낸다[9].

게임 밸런스를 위한 첫 번째 고려에서 어떤 게임이든 캐릭터에 대한 정적 요소를 추출하여 캐릭터 상호간의 밸런스를 구현하는 것이 가장 기본적인 방법이다. 게임의 진행에 따라 캐릭터의 속성이나 난이도가 변화는 동적 게임 레벨 디자인[10]에서도 결국 동적으로 변화된 데이터의 밸런스는 항상 일정하게 유지되어야 한다.

〈그림 3〉은 캐릭터간의 대전 시스템에서 직간접적인 밸런스 요소들의 상호 연관관계이다.



[그림 3] 캐릭터간의 대전시 직간접 요소 관계

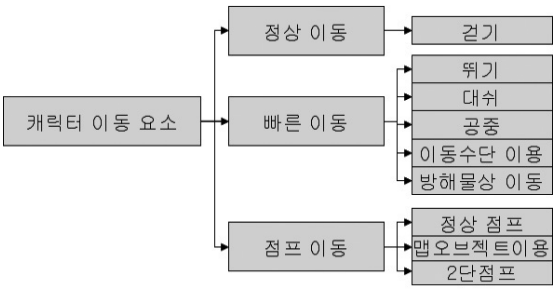
캐릭터간의 대전에 영향을 미치는 요소는 캐릭터의 공격과 방어와 연관된 직접적인 수치들과 주변 맵 오브젝트 등과 같은 환경적인 간접 영향 요소들이 있다.

대전 시스템은 최종적으로 캐릭터 상호 공격과 방어의 형태를 근간으로 밸런스가 이루어지며, 아이템의 획득에 따른 능력치의 변동, 종족간의 능력차, 자원 제약 요소 등은 공격과 방어에 따른 직접적인 요소로 반영되어 게임이 진행된다.

우리가 대전 전투 시스템에서 정형화 할 수 있는 것은 이런 캐릭터 요소에 근간한 공격과 방어에 관련한 직접적인 요소들이다.

캐릭터와 관련한 요소 중 가장 기본이 되는 요소는 캐릭터의 이동성에 관한 사항이다. 캐릭터 이동성은 방어와 공격 시 영향을 미치는 직접적인 요소이다. 캐릭터 이동성이 빠른 경우 공격적인 측면과 방어적인 측면에서 유리한 요소를 갖는다. 그러므로 이에 대한 수학적 밸런스는 거시적 밸런스의 기본이다.

게임 내에서 캐릭터가 가지는 밸런스 요소 중 이동 즉 움직임에 관련한 요소는 〈그림 4〉와 같이 분류한다.



[그림 4] 캐릭터 이동에 관한 요소

이동 요소는 캐릭터의 게임 플레이세계와 매우 밀접한 관계를 가진다. 캐릭터 이동이 모두 일정한 시간에 따라 움직인다면 게임 밸런스에 있어서 근본적인 요소 하나를 해결한 것이다. 모든 캐릭터들의 움직임이 동일하다면 대전에서 공격과 방어의 직접적인 요소들만이 밸런스 요소가 된다. 실제 격투 대전 게임에서 주요 캐릭터들의 이동 요소가 동일하게 구성되어 있다. 이러한 예는 철권, 길티기어 등의 전형적인 대전 격투 게임에서 관찰할 수 있다. 사용자 요구 반영과 게임의 재미를 향상시키는 차원에서 캐릭터 이동 속도는 다양해 질 수 있고 이의 정형화된 밸런스 조정은 매우 의미 있는 일이다.

게임 내에서 캐릭터 이동 속도가 서로 다르다면 공격을 하거나 피하는 시간적인 차이에 따라 게임 밸런스에 심각한 영향을 미친다. 각 캐릭터 간 이동 속도의 차이가 있는 게임의 경우 밸런스를 조정하기가 매우 난해하며 이를 정형화한다는 것은 게임을 효율적으로 개발하기 위한 기본요소를 달성하는 것이다.

캐릭터가 움직이는 환경은 일반적으로 지상상태에서 자연스럽게 움직이는 경우와 이동 수단 또는 뛰는 등의 기본 이동 속도와 차이가 나는 경우이다. 기본 이동이라 함은 정상적인 걷기 동작이며 응용 동작이라 함은 뛰기 또는 다양한 이동 수단을 이용한 방법으로 정상적인 걷기와 차별되는 요소이다.

이동 요소의 밸런스는 이성적 판단에 의하여 시작한다. 캐릭터의 정상 이동시에 구현되는 속도는 게임 내에서 이동하는 걷기 속도이고 응용 이동은 이에 대한 속도 가감으로 구분한다.

대전 시스템에서 점프는 대부분이 게임에서 모든 캐릭터들이 같은 속도와 높이로 현재까지 구현되어왔다. 점프 속성의 사용은 단지 플레이어의 조작 능력에 기인하게 개발되었으며, 공격 및 방어를 위한 캐릭터간의 차별적 요소는 아니었다.

점프 속성이 캐릭터마다 다르다면 상대 캐릭터의 공격을 피할 수 있는 캐릭터 능력치 중 하나가 된다. 점프 요소를 고려할 때 점프 길이와 점프 높이 그리고 점프에 걸리는 시간적 요소가 캐릭터마다 차별적으로 고려된다면 더욱 다양한 캐릭터 특성을 부여할 수 있다. 캐릭터별 차별적인 점프 요소는 플레이어의 조작 능력과 함께 게임의 재미를 부가할 것이다.

맵 오브젝트를 이용하여 점프의 높이나 길이 그리고 점프 시간이 변한다면 캐릭터 간 상호 보완적인 요소를 찾아야 한다.

공격 유형은 대전 시스템에서 상대 캐릭터에 영향을 주는 가장 직접적인 요소이다. 최근 게임들은 다양한 형태의 공격 형태를 화려하게 시전하고 시각적인 효과를 높이기 위하여 스프라이트(sprite) 같은 2차원적인 그래픽 요소와 파티클(particle) 및 3D 모델을 조합한 형태의 이펙트 효과를 이용한다. 대전이 전개될 때 각 캐릭터는 다양한 형태의 공격 형태를 취할 수 있으며 이를 밸런스 요소로 구분하여 보면 〈그림 5〉와 같다.

[그림 5] 공격 형태에 따른 밸런스 요소

공격 효과가 발동되기까지 걸리는 시간적 요소는 실제 플레이어가 공격하기를 원할 경우 공격 버튼 또는 마우스 등을 사용하여 공격을 명령할 때 캐릭터들의 준비 애니메이션에 걸리는 시간적 요소이다. 이런 요소의 구현은 플레이어로 하여금 공격 명령 후 공격이 시작되고 있음을 인지하게 하여 게임의 재미를 높인다.

공격 효과가 발동된 후 타격까지 걸리는 시간적 요소는 원격 공격물에 의한 공격 또는 손, 발 및 무기를 이용하여 타격하는 근접 공격을 할 때 캐릭터가 움직이기 시작하여 공격 애니메이션이 완료될 때까지의 시간적 요소이다.

공격 패턴은 단방향, 양방향 또는 캐릭터를 중심으로 일정 거리를 유지한 형태로 이루어지는 원형 공격 등 방향성을 의미한다.

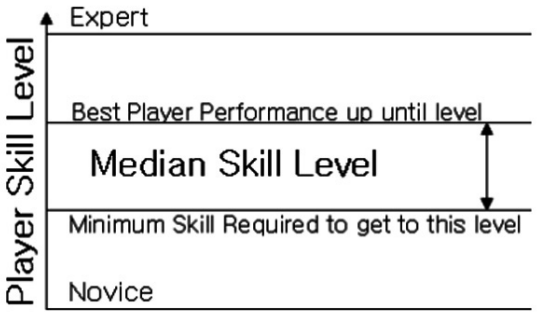
공격 범위는 캐릭터가 공격 시전 시 영향을 미치는 거리 요소이다. 거리 요소는 공격의 효과적인 측면에서 캐릭터간 밸런스가 이루어져하는 요소이다.

공격 효과의 강도는 공격에 의한 타격 시 해당 공격에 의한 상대 캐릭터에 대한 데미지(damage) 요소이다. 데미지 요소는 공격을 당한 캐릭터가 어느 정도의 영향을 받았는지에 대하여 상호 보완적으로 밸런스가 이루어진다.

공격 효과가 지속되는 시간은 공격을 당한 후 해당 캐릭터가 영향을 받는 시간적 요소로서 특정 공격을 받은 후 타격 결과에 따른 시각적 효과와 공격 받은 캐릭터가 다음 형태의 움직임까지 영향을 받는 시간적 단절 현상으로 구분한다. 공격을 시행한 캐릭터는 다음 연속 공격을 수행할 수 있는 시간적 이득을 받는 요소이고 상대적으로 공격을 받은 캐릭터는 방어하기까지의 시간적 손실을 받는 요소이다.

3. 대전에 직접적인 영향을 끼치는 캐릭터 요소들의 구현

캐릭터간의 대전을 즐기는 플레이어들은 낮은 수준의 레벨에서 게임을 시작할 수도 있고 높은 레벨의 수준에서 플레이할 수도 있다. 대전에 직접적으로 영향을 끼치는 속성이 반영된 캐릭터 스킬 수준을 결정하는 방법은 〈그림 6〉와 같이 낮은 레벨과 높은 레벨사이의 임계 구간내의 중간스킬 레벨을 결정하는 것이다[11].



[그림 6] 임계 구간을 갖는 플레이어 스킬레벨

캐릭터 속성들의 중간 스킬 레벨의 결정은 기준이 되는 캐릭터를 중심으로 상호 보완적으로 시작된다. 기준 캐릭터가 설정되고 관련 속성이 결정되면 이 값을 바탕으로 상대적인 캐릭터들의 설정 값들을 결정한다.

기준 캐릭터의 설정은 플레이 테스트에 의하여 정한다. 플레이 테스트는 게임 개발 기간 동안 특별히 유용한 방법이며 다중 사용자 게임에서 필수불가결한 요소이다[12].

3.1 이동 요소의 구현

캐릭터 이동에 관한 밸런스는 기준 캐릭터의 표준 이동 속도를 정하는 것에서 출발한다. 표준이 되는 캐릭터의 이동 속도를 기준으로 상대적인 각 캐릭터들의 이동 속도를 구해야 하므로 초기 기준 값의 설정이 중요하다.

캐릭터의 표준 이동 속도를 설정하기 위하여 표준 이동 환경을 정의한다. 표준이 되는 정상이동 방법은 지상에서의 걷기 동작이다. 게임이 수상, 공중 등 다른 세계를 기준으로 잡는다면 그것은 게임의 주요 무대가 어디인가에 대한 문제이며 밸런스 방법이 변화하지 않는다.

게임 내에서 캐릭터의 이동은 이동 동작을 위한 캐릭터 애니메이션 프레임수로 결정되며 맵 위에서는 단지 적절한 속도에 의하여 이동시킨다. 캐릭터의 이동 속도를 결정하기 위하여 캐릭터 동작 애니메이션 프레임 수의 조정이 필요하며 적절한 이동 동작이 인지되지 않으면 캐릭터에디터 등의 툴을 이용하거나 해당 캐릭터의 이동 애니메이션을 수정하여 적절한 움직임을 확보한다. 적절한 이동 동작이라 함은 게임에서 맵의 사이즈와 맵 위에 위치하게 되는 캐릭터의 크기, 갯수, 맵 오브젝트들의 크기 및 그에 따른 영향을 고려한 이동 동작을 의미하는 것으로, 게임 디자인에서 수직적, 수평적 평형상태를 느끼게 하는 균형을 유지[13]하게 하는 것이다. 이는 게임 플레이어들의 경험적 방법에 의하여 결정한다.

지상에서 적절한 표준 이동 속도가 정의되면 이 속도를 기준 값으로 하여 다른 응용 이동 수단을 이용한 속도를 정의한다. 응용 이동 수단의 속도는 설정 값을 기준으로 상대적으로 크거나 작은 수이며, 응용 이동 방법에 따라 가감 배수로서 쉽게 결정한다.

응용 이동 = 기준 캐릭터의 이동 속도 * 가감 배수

게임 내에서 캐릭터들의 상대적 이동 속도는 해당 캐릭터의 체력 또는 능력치에 따라 다르며, 게임 디자이너가 고려한 수치는 상대적인 값으로 결정한다. 이때 캐릭터들의 이동 속도가 다르면 해당 캐릭터의 이동 동작 애니메이션의

프레임수를 연동하여 변경한다.

점프에 대한 요소는 게임 내에서 캐릭터 이동 속도에 대한 보완적인 차원에서 다음과 같이 구현한다.

Effectiveness(E) = 점프길이 * 점프높이 * 점프 속도

이때 유효(Effectiveness) 값은 캐릭터 상호간의 영향을 값으로 정의하고 일정 값을 유지하게 한다.

점프 길이는 "E = 점프 길이 * 이동 속도"를 이용하여 구하며, E 값은 표준 이동속도와 표준 점프길이를 기준으로 구한다.

표준이동 속도는 캐릭터가 정상적인 이동을 한다고 가정하고 점프 요소를 고려하여 걷기와 점프를 같은 시간만큼 번갈아 한다고 가정하고 다음과 같이 구한다.

$$\text{기준 캐릭터의 평균 이동 속도} = \frac{[m * \{(100/100) * k\}] + [n * A * \{(100/100) * k\}]}{(m+n)}$$

$$= \frac{K(m+n)}{(m+n)}$$

이때,
m = 뛰는 시간 (단위 : 초)
n = 점프 시간 (단위 : 초)
k = 뛰는 속도 상수
A = n초 동안 기준 캐릭터의 점프 길이로 정의된다.

이때 각 캐릭터의 평균 이동 속도와 기준이 되는 캐릭터의 평균 이동 속도는 같아야 한다.

$$\frac{[m * \{(V/100) * k\}] + [n * a * \{(V/100) * k\}]}{(m+n)} = \frac{K(m + An)}{(m+n)}$$

그러므로
각 캐릭터의 점프 길이 a는 다름과 같이 구할 수 있다.

$$\frac{100(A + m) - mV}{nV}$$

이때,
m = 뛰는 시간 (단위 : 초)
n = 점프 시간 (단위 : 초)
k = 뛰는 속도 상수
A = 기준 캐릭터의 점프 길이
a = 각 캐릭터의 점프 길이
V = 각 캐릭터의 뛰는 속도이다.

순간 이동 또는 데쉬(dash) 이동의 구현은
"기준 캐릭터의 대쉬 이동 속도 = 각 캐릭터들의 평균 데쉬 이동 속도" 를 만족해야 하므로
$100 * x = V * y$에서

$$y = \frac{100 * x}{V}$$

로 쉽게 구할 수 있다.
이때,
x = 기준 캐릭터의 Dash 이동 속도 상수
V = 각 캐릭터의 이동 속도
y = 각 캐릭터의 Dash 이동 속도 상수이다.

3.2 공격 형태에 따른 구현

공격 형태에 따른 응용 공격들은 〈표 1〉과 같다. 공격 형태에 따른 밸런스는 다음에서 설명하는 바와 같이 쉽게 정형화하여 구현할 수 있다.

공격 형태는 범위, 직선, 곡선, 내려치기, 회전 형태로 크게 구분하며, 이런 기본 유형을 조합하여 구성한다.

유형	범위예	사방성	응용분야	공통고려요소
범위		사방	마법진 등	효과 발동까지 걸리는 시간에 필요한 캐릭터 애니메이션 시간 효과 발동 후 타격 가능한 시간에 필요한 캐릭터 애니메이션 시간
직선		단방	화살, 창, 스트레이트, 주먹, 직선선상의 마법 등	
직선 및 곡선		단방과 사방 사이	손 발 등을 이용한 휘두르기, 어퍼컷, 무기를 이용한 직선 및 곡선이 조합된 응용 공격 등	
내려치기		단방	주먹 및 발 등을 이용한 내려찍기, 손 및 무기류를 이용한 단방향의 내려치기	
회전		순차 사방	발을 이용한 돌려차기, 무기를 이용한 순차 공격 유형 등	

[표 1] 게임에서 공격 유형에 따른 구분

〈표 1〉에서 범위 공격은 격투 대전 게임에서는 드물게 이루어지는 형태이나 RPG등 캐릭터 상호간의 전투 시스템에서 마법진 등의 형태로 구현된다. 범위 공격의 경우 원격 공격뿐만 아니라 근접 상황에서도 이루어 질수 있는 공격이다.

직선 공격은 가장 전형적인 공격 방법 중 하나이다. RPG 게임의 캐릭터 상호간 전투에서 화살 및 창 등의 공격 형태를 예로 들 수 있다. 대전 격투 게임에서는 스트레이트 주먹 공격 형태이다.

직선 및 곡선 조합의 공격 형태는 일반적인 게임에서는 볼 수 없는 요소이다. 이를 응용하면 대전 격투 게임에서 발로 찍어 돌려 차기 등 다양한 응용 공격 동작과 방어 동작을 구현할 수 있다. 칼 또는 창을 가지고 전투를 실행할 때 다양한 공격 유형을 창조할 수 있다.

내려치기 공격은 대전 격투 및 창 또는 칼 같은 무기류를 이용한 대전에서 사용되는 전형적인 예이다. 모든 공격은 시작점과 공격이 마감되는 종료 점까지의 이동에 의한 공격 형태이다.

회전 공격은 순차 사방성을 갖는 공격으로 공격 범위의 일정 부분이 사방성에서 제외되는 것이 범위 공격과 차별된다.

공격 형태에 따른 밸런스를 거시적으로 구현함에 있어 절대 단 하나라도 쓸모없는 파라미터가 없고, 계산식은 간단할수록 좋고, 모든 파라미터가 충분한 역할을 수행할 수 있는 범위를 가지도록 해야 한다[14].

위와 같은 원칙하에 공격 형태에 따른 밸런스 구현은 다음과 같다.

A= (공격범위 * 사방성 * 효과발동 후 타격 가능한 시간)

B= 효과 발동까지 걸리는 시간

C= 캐릭터 이동 속도

A,B,C를 위와 같이 정의하고 이에 따른 밸런스 값은 다음의 유효 값으로 정의한다.

즉, Effectiveness(E) = A / B * C로 정의한다.

이때, 각 값들은 E값을 일정 기준 값으로 한 상대적 값을 사용하여 기준 캐릭터의 평균 공격 범위, 평균 공격 사방성, 평균 효과 발동 후 타격 가능한 시간 그리고 평균 효과 발동까지 걸리는 시간을 순차적으로 적용하면서 구한다.

3.3 밸런스 방법의 효과 분석

본 논문에서 제시한 밸런스 방법을 캐릭터 상호간의 대전 시스템에 적용하면 다음과 같은 효과를 얻을 수 있다. s개의 스테이지, c' 개의 캐릭터 능력 향상치를 갖는 c개의 캐릭터, 서로 다른 공격물 a개, m' 개의 맵 오브젝트 수를 갖는 m개의 맵에서 플레이가 진행된다고 가정하고 동일 캐릭터들의 대전을 허용할 경우 "(m * m' * (c * a * c') * m * m' * (c * a * c')) * s"수만큼의 게임 플레이 테스트가 밸런스를 검증하기 위하여 진행되어야 한다. 이때 맵 오브젝트는 상대 캐릭터 상호간에 같은 영향을 가지므로 게임내의 스테이지 개수만큼의 테스트로 충분하며 캐릭터의 능력치는 실제 공격 및 방어적인 직접 요소에 반영되므로 고려하지 않아도 된다. 그러므로 승리를 위한 테스트만을 고려할 때 c * a * s개수만큼의 테스트로 충분하다.

상용 버전의 테스트 시, 캐릭터 능력치에 대한 적절한 값의 확보, 플레이 테스트 및 게임 오브젝트들의 영향 요소 분석 등 간접 요소들을 활용한 미시적 밸런스 조정을 위한 추가 테스트는 필요하다.

4. 결론 및 향후 과제

우리는 캐릭터 속성을 중심으로 직접 대전 시스템에 적용되는 게임 밸런스 요소를 구분하였고 거시적 밸런스에 관한 방법을 수치를 사용한 공식으로 구현하여 누구나 쉽게 적용할 수 있게 하였다. 캐릭터를 근간으로 하는 전투 시

스템의 밸런스는 다양한 장르의 게임을 개발하는데 있어 공통적인 요소로 사용될 수 있다.

간단하고 효율적인 밸런스 방법을 구현하여 캐릭터간의 대전시 현재까지 잘 구현되지 않았던 캐릭터 이동성과 점프 등을 각 캐릭터가 상이하게 보유할 수 있게 하였다. 이것은 대전 격투 게임 및 RPG 등의 캐릭터 상호간의 전투 시스템에 다양성을 부가하여 대전을 좀 더 재미있게 구성하게 하는 것을 가능하게 하였다.

공격 형태에 있어서 현재까지 개발되었던 전투 공격 형태를 포함하고 원격 및 근접 공격의 동시 구현, 직선과 곡선 조합의 새로운 공격형태, 공격 결과에 대한 영향 요소 등을 구분하여 적용 가능한 수학적 밸런스 방법을 구현하였다.

그러나 게임 밸런스 요소 모두를 체계화하는 것은 현실적으로 어려운 문제이며, 미시적 밸런스 방법에 대한 체계적인 접근 방안에 대한 연구가 필요한 사항이다.

참고문헌

[1] Ernest Adams, A Symmetry Lesson, Gamasutra, October 16 1998

[2] 용대순, 재미요소를 고려한 게임 디자인론에 관한 연구, 게임산업저널, 2003

[3] Adam capenter, Applying Risk Analysis to Play-balance RPGs, CMP Media LLC. Gamasutra, June 11 2003

[4] Ernest Adams, Balancing Games with Positive Feedback, CMP Media LLC. Gamasutra, January 4 2002

[5] James A. Portnow, The Importance of Risk in Basic Design, Gamasutra, September 12 2006

[6] 류태영,오규환, 온라인 게임에서 부분 유로화를 위한 게임 디자인, 게임산업저널, 통권13, 1호, p26-p44, 2006년

[7] Justin Lloyd, Book Review:Andrew Rollings and Ernest Adams on Game Design, Gamasutra, March 30 2004

[8] David Kennerly, Better Game Design through Data Mining, Gamasutra, August 15 2003

[9] 김미진, 김재준, RPG 게임캐릭터의 정적 발란싱 디자인에 관한 연구, 한국콘텐츠학회 논문지 5권, 5호, p100-p106, 2005

[10] 아상경,정기철, 플레이어 적응형 GMM 기반 동적 게임 레벨 디자인, 한국게임학회 논문지 제6권, 1호, p3-p10, 2006

[11] Tim Ryan, Beginning Level Design Part 2: Rules to Design By and Parting Advice, Gamasutra, April 23 1999

[12] Pascal Luban, Multiplayer Level Design In-depth, part 3:Technical Constraints and Accessibility, CMP Media LLC, Gamasutra, November 22 2006

[13] Tito Pagan, Where's the Design in Level Design?, Part Two, Gamasutra, July 16 2001

[14] 정동현, 게임 제작의 알파와 오메가, p292-p313, 영진닷컴, 2004

박 찬 일 (Chan-Il Park)

1988년: 서강대학교대학원 전자계산학석사
1988년-2000년: LG software, LG전자, LG 인터넷 근무
2000년-2001년: 보이시안 부사장
2001년-2003년: 시네픽스 이사
2004년-현재: 청강문화산업대학 컴퓨터게임과 교수
서울벤처정보대학원대학교 컴퓨터응용기술학과 박사과정

관심분야 : 게임 디자인, 게임 밸런스, 게임 프로젝트 관리

양 해 술 (Hae-Sool Yang)

1991년: 일본 大阪대학교 소프트웨어전공 공학박사
1980년-1995년: 강원대학교 전자계산학과 교수
1995년-2002년: 한국소프트웨어품질연구소 소장
1999년-현재: 호서대학교 벤처전문대학원 컴퓨터응용기술학과 교수

관심분야 : 소프트웨어 공학(특히, S/W 품질보증과 품질평가, 품질감리, 품질컨설팅), 게임 프로젝트 관리 등