게임메카닉스 시뮬레이터 구현

장 희 동†

요 약

개발비용이 70억원에서 200억원에 이르는 블록버스터 게임들이 자주 등장하면서 전반적으로 게임개발 규모가 급속히 커지고 있다. 이로 인해 게임의 기술요소들이 집중되어 있는 게임메카닉스의 품질관리가 중요하게 되었다. 본 논문에서는, 설계단계에서, 게임메카닉스의 공학적 품질 분석을 정확하고 경제적으로 할 수 있는 컴퓨터 시뮬레이터를 구현하여 제안하였다. 제안하는 시뮬레이터는 모델링을 편리하게 할 수 있도록, Petri net[7,8]과 Smalltalk[9]을 지원한다. 기존 방법들은, 실제의 게임플레이 환경을 고려하지 않는, 제한된 범위의 품질분석이 가능했지만, 제안하는 시뮬레이터는 플레이어의 게임플레이 행동패턴과 게임의 배경맵과 아이템 DB를 사용하기 때문에 그 평가 결과가 매우 실제적이다. 제안하는 시뮬레이터의 성능을 검증하기 위해 Dungeon & Dragon[13,14]의 게임메카닉스와 게임 배경맵과 아이템 DB를 사용하여 시뮬레이션을 총 80회 실시하였다. 실험 결과에 의하면, 제안하는 시뮬레이터는 무결성, 최적화, 밸런스 품질을 모두 평가할 수 있고, 다른 방법보다 품질평가 능력이 우수하였다.

키워드 : 컴퓨터게임, 게임메카닉스, 페트리네트, 컴퓨터 시뮬레이션

An Implementation of the Game Mechanics Simulator

Hee Dong Chang†

ABSTRACT

The scale of game development are rapidly increasing as the blockbuster games which cost 7~20 billion won, often appear on the markets. The game mechanics which is concentrated on technological elements of the game, necessarily requires the management of quality. In this paper, we propose a computer simulator for the quality evaluation of game mechanics which can analyze the quality accurately and economically in the design phase. The proposed simulator provides Petri net[7,8] and Smalltalk[9] for convenient modeling. The simulator gives the realistic evaluation like play test because it uses the realistic data of gameplay environment such as player action-pattern, game world map, and item DB but the previous evaluation methods can not consider the realistic gameplay environment and can only cover a limited scope of evaluation. To prove good performance of the proposed simulator, we have 80 simulations for the quality evaluation of the game mechanics of Dungeon & Dragon[13,14] in a given world map. The simulation results show that the proposed simulator can evaluate the faultlessness, optimization, and play balance of the game mechanics and gives better good performance than other evaluation methods.

Key Words : Computer Game, Game Mechanics, Petri Net, Computer Simulation

1. 서 론

2004년도 세계게임시장 규모는 611억불 그리고 국내게임시장 규모는 4조 5,351억원으로 추정되고 있다. 특히 국내 게임시장의 규모는 영화 및 음악의 국내시장규모의 2.4배이며 작년대비 15.8%의 고속성장을 하고 있다[1]. 또한 게임의 개발규모 측면에서 보면, 블리자드 엔터테인먼트사의 「월드 오브 워크래프트(WOW)」의 개발비는 약 200억원, NC 소프트사의 「리니지 2」는 약 1백억원, CCR사의 「RF온라인」은 80억원, 그리고 넥슨의 「마비노기」는 72억원으로 블록버스터 게임들이 자주 등장하고 있다. 이러한 개발 규모의 증가는, 국내 게임산업체들에게, 실패를 최소화하고 성공적인 개발을 위한, 전문적인 프로젝트 관리기법을 요구하고 있다. 게임 개발 과정은 실시간 데이터베이스 소프트웨어 개발과정과 비슷하기 때문에 게임 분야에 적합한 소프트웨어 공학에 근거한 프로젝트 관리기법을 적용할 수 있다[2].

전통적인 게임개발과정은, 일반 소프트웨어 개발과정과는 달리, 반복적인 시행착오를 통해 점진적으로 게임의 형상을 구체화하고 결정하기 때문에 설계단계부터 구현단계까지 게임의 형상변경이 자주 일어난다. 이로 인해 품질관리 활동들이 알파버전과 베타버전의 테스트 기간에 치중되어 있다. 소

※ 본 연구는 2004년도 호서대학교 기초학문연구비 지원으로 이루어졌음.
† 정 회 원 : 호서대학교 컴퓨터공학부 게임공학과 조교수
논문접수 : 2005년 6월 29일, 심사완료 : 2005년 8월 11일

프트웨어공학 이론에 따르면, 문제발견 및 해결 활동이 구현 단계 후 시험단계에서 이루어지는 비용은, 설계단계에서 이루어지는 비용보다, 상대적으로 약 100배에서 200배 정도 높게 나타난다[2].

게임은 게임메카닉스(game mechanics)와 게임 컨텐츠(game contents)로 구성되어 있다. 게임메카닉스는 특정 플랫폼에서 게임플레이가 실행되게 하는 구체적인 방법으로 체계적인 시스템이고 게임 컨텐츠는 게임그래픽, 게임사운드와 같은 데이터들이다. 게임메카닉스는 게임의 공학적 품질을 결정하고 또한 감성적 품질인 게임연출성에도 많은 영향을 준다. 게임 컨텐츠는 게임의 감성적 품질을 결정한다.

따라서 게임메카닉스가 공학적 품질을 결정하기 때문에, 설계단계에서 정확한 품질 분석이 이루어져 오류해결 및 품질개선의 비용을 최소화해야 한다.

게임개발의 설계단계에서 게임메카닉스의 오류발견, 최적화, 그리고 밸런싱에 관한 연구결과들은 참고문헌 [3]~[6]에서 찾을 수 있다.

참고문헌[3]에서는 설계단계에서 게임플레이 밸런스를 찾기 위해, 각 의사결정에 대한 보수행렬(pay-off matrix)의 수학적 해를 구하는 방법을 소개하였다. 그러나 소개된 방법은 수학적 해를 구하기 위한 다양한 제한조건들을 사용하기 때문에 실제 측정값과 차이가 있고 분석범위도 매우 제한적이다.

참고문헌 [4]에서는, 설계단계에서 게임메카닉스의 품질분석을 위한 시뮬레이션 방법들을 조사하여 그 특징들을 분석하였다. 조사된 방법들은 보수행렬(payoff matrix) 방법, 마이크로소프트 엑셀(Microsoft Excel) 방법, 게임플레이 모형 방법, 프로토타입(prototype) 방법, 그리고 컴퓨터시뮬레이션 방법이다. 특히 컴퓨터시뮬레이션은, 다른 방법들 보다 분석범위가 넓어, 게임메카닉스의 무결성, 최적화, 그리고 밸런스의 품질을 모두 분석할 수 있다.

참고문헌 [5]에서는 게임구성요소의 각 속성에 대한 상대적 중요도를 나타내는 가중치(weight)들을 정한 후, 게임구성요소들 간에 밸런스가 이루어지는, 속성 값들을 찾는 방법이다. 이 방법은 설계자의 주관적 판단에 의해 가중치들이 결정되기 때문에 결과의 정확성이 떨어진다.

참고문헌 [6]에서는 TRPG(Table-top Roll Playing Game) Dungeon & Dragon의 전투규칙 부분을 Petri net으로 모델링하고 컴퓨터 시뮬레이션을 통해 각 경우의 전투액션에 대한 사용 빈도수(frequency)를 분석하였다. 이 방법은 배경맵, 등장캐릭터, 그리고 마법과 같은 플레이요소들을 고려하지 않고 단지 전투규칙에 대한 논리적인 흐름만을 고려하여 분석하였기 때문에 분석결과의 적용범위가 제한적이다.

본 논문에서는 게임개발의 설계단계에서 게임메카닉스의 무결성, 최적화, 그리고 밸런스의 품질들을 정확하고 경제적으로 평가할 수 있는 컴퓨터 시뮬레이터(computer simulator)을 구현하여 제안한다.

본 연구에서 제안하는 시뮬레이터는 게임메카닉스의 모델링을 정확하고 효율적으로 할 수 있도록 Petri net[7, 8]과 객체지향언어인 Smalltalk[9]을 사용한다. 또한 기존 품질분석 방법들은 실제 게임플레이 환경을 반영하지 못해 단편적인 환경에서 평가하지만, 제안하는 시뮬레이터는 게임유저의 플레이 행동패턴과 게임에서 실제 사용하는 게임배경맵, 캐릭터 클래스 속성들, 무기/방어/마법/기타 아이템들에 대한 게임데이터 DB를 사용하여 실제적인 플레이환경을 반영하여 평가하기 때문에 실험결과가 실제적이고 정확하다.

본 논문은, 2장에서 게임플레이와 게임메카닉스의 개념들과 서로의 연관성에 대하여 설명하고, 3장에서 시뮬레이터 구현에 대하여, 4장에서 시뮬레이션 내역에 대하여, 5장에서 시뮬레이션의 결과에 대하여, 그리고 6장에서 결론을 다룬다.

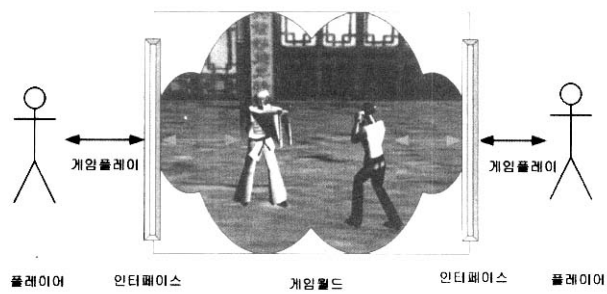
2. 게임플레이와 게임메카닉스 개념

게임[10]은 일반 놀이와는 달리 주어진 목표를 두고 참여자들끼리 상호 영향을 주고받으며 상대와 겨루는 놀이를 의미하고 컴퓨터게임은 사이버스페이스(cyberspace)[1]에서 이루어지는 게임을 말한다.

컴퓨터게임은, (그림 1)과 같이, 게임월드(game world)와 인터페이스(interface), 그리고 게임플레이(gameplay)로 구성되어있다.

게임월드는 공간상의 배경(예: 들판, 바다), 등장요소(예: 캐릭터들), 플레이요소(예: 아이템, 마법, 스킬)들을 포함하고 주어진 규칙에 따라 인터페이스를 통한 플레이어의 선택행위에 반응하는 시스템이다. 인터페이스는 실세계에 있는 플레이어와 가상세계에 있는 게임월드 사이를 연결해 주는 접촉면으로, 입출력기기들을 통해 구현된다. 게임플레이는 플레이어가 인터페이스를 통해 가상공간인 게임월드와 인터랙션(interaction)하는 게임행위이다.

게임메카닉스(game mechanics)는 게임플레이를 게임플랫폼에서 실행되도록 특정한 방법으로 구현된 것을 의미한다[11]. 예를 들면, 주인공 캐릭터를 A지점으로 이동시키는 행위의 게임플레이를 마우스 왼쪽버튼을 클릭하여 주인공 캐릭터를 선택한 다음 A지점으로 오른쪽 버튼을 클릭 함으로서 주인공 캐릭터가 A지점으로 이동하는 것이 구현되었다면 이는 주인공 캐릭터를 A지점으로 이동시키는 하나의 게임 메카닉(game mechanic)이 되는 것이다. 게임메카닉스는 한 게임에서 필요한 모든 게임플레이를 구현한 게임 메카닉들이



(그림 1) 컴퓨터게임 개념도

1) 컴퓨터 또는 컴퓨터네트워크에 의해 만들어지는 가상공간

〈표 1〉 게임메카닉스 품질 부문 및 평가대상 항목들

부문	공학적 품질 부문			감성적 품질 부문
	무결성	최적화	밸런스	연출성
평가 대상	①게임플레이 흐름 및 판정 ②정보의 입출력 ③게임월드 데이터 관리	①게임플레이 흐름 및 판정 ②정보의 입출력 ③게임월드 데이터 관리	①플레이어/플레이어 밸런스 ②플레이어/게임플레이 밸런스 ③게임플레이/게임플레이 밸런스	①게임월드 표현성 및 반응성 ②캐릭터 표현성 ③게임플레이 시점 ④인터페이스 .인터랙션 정보 .인터랙션 동작 .인터랙션 디자인 ⑤특수효과 연출

게임흐름에 따라 연결되고 통합된 것이다.

게임메카닉스는 게임의 기술적 요소들이 집중되어 있기 때문에, 설계단계부터 품질관리가 이루어져야 하는 대상이다. 게임메카닉스의 품질은, <표 1>과 같이, 무결성, 최적화, 밸런스, 연출성 부문들로 나눌 수 있다. 무결성 품질 평가는 게임메카닉스가 실행 중에 오류가 발생하는지 또는 잠재적 오류가 있는지를 테스트하는 것이고 최적화 품질 평가는 게임메카닉스가 실행 중에 병목현상, 그리고 불필요한 부분들이 없는지 테스트한다. 밸런스 품질 평가는 게임등장요소들과 게임플레이요소들 간의 밸런스[3]를 테스트한다. 연출성 품질 평가는 게임플레이 컨셉(concept)의 오락성과 예술성이 게임메카닉스에서 정확하게 반영했는지를 테스트한다.

3. 시뮬레이터 구현

본 논문에서 제안하는 시뮬레이터(simulator)는 설계단계에서 게임메카닉스의 무결성, 최적화, 그리고 밸런싱 부문의 품질을 효과적이고 경제적으로 평가하기 위함이다. 또한 게임개발 특성상 자주 일어나는 형상변경을 유연하게 수용할 수 있어야 한다.

게임메카닉스의 모델링과 품질평가를 위해, 시뮬레이터는 독일의 IBE사가 개발한 PACE5.0e[12] 소프트웨어를 사용해 개발하였다.

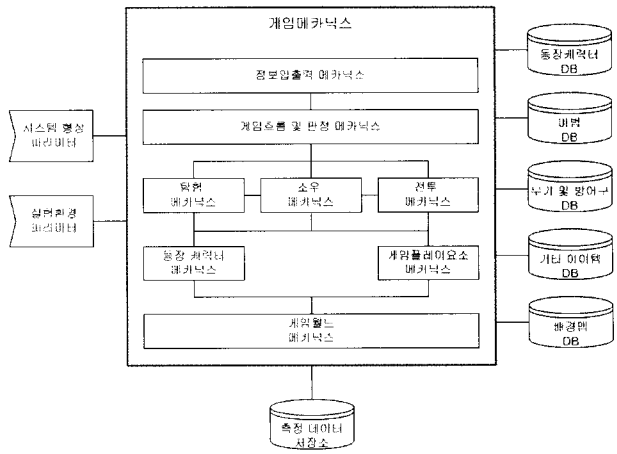
PACE5.0e는 마이크로소프트 윈도우즈 운영체제에서 실행되며 Petri net[7, 8]을 기반으로 하는 객체지향 시뮬레이터 개발 시스템 (object-oriented simulator-development-system)이다. 또한 시뮬레이터 개발언어로 객체지향 프로그래밍언어인 smalltalk[13]을 지원한다.

본 논문에서 제안하는 시뮬레이터는 편의상 RPG(Roll Playing Game) 또는 MMORPG(Massively Multi-player Online RPG)장르에서 널리 활용되고 있는 Dungeon & Dragon(이후로 D&D로 표기함)[13, 14]의 게임메카닉스를 기본적으로 사용하였다.

D&D에서는 성직자, 전사, 마법사 등 7 종류의 캐릭터 클래스들이 존재하고 캐릭터는 고유한 능력과 약점들을 갖고 있다. 예를 들면, 성직자는 모든 방어구 사용과 언데드 몬스터의 퇴치능력이 있고, 전사는 모든 무기 및 방어구 사용과 근접 전투에 능하고, 마법사는 마법사용에 탁월한 능력을 갖고 있다.

3.1 시뮬레이터 구조

시뮬레이터의 구조는, (그림 2)와 같이, 중앙에 게임메카닉스 모델, 우편에는 등장캐릭터, 마법, 무기 및 방어구, 기타 아이템, 그리고 배경맵에 대한 DB들, 좌편에는 게임메카닉스 시스템 형상파라미터들과 실험환경 파라미터들에 대한 입력 데이터들, 그리고 아래편에는 측정 데이터를 수집하여 보관하는 저장장소로 구성되어 있다. 시스템형상 파라미터는 게임메카닉스의 형상을 결정하는 파라미터들이다. 예를 들면, 이동몬스터 조우표 ID와 PC 파티(party)의 경험치 배분방식 파라미터들이다. 실험환경 파라미터는 실험환경을 결정하는 파라미터로써, 예를 들면, 시뮬레이션 시간과 종료조건 그리고 등장인물 파라미터들이다.



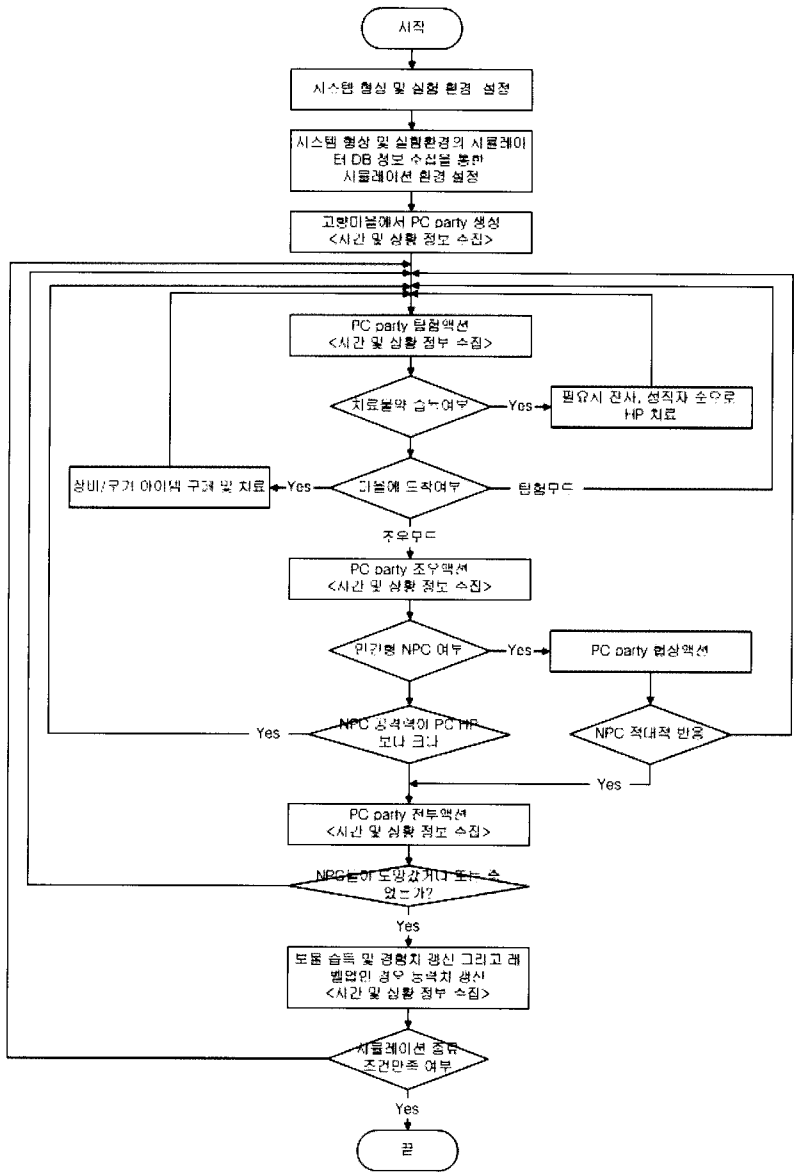
(그림 2) 시뮬레이터 구조

시뮬레이터의 게임메카닉스는 게임흐름 및 판정, 탐험, 조우, 전투, 등장캐릭터, 게임플레이 요소(예: 마법, 무기, 방어구, 보물, 마법아이템), 게임월드에 대한 부분들로 구성되어 있으며 정보입출력 부분은 시뮬레이션의 목적에 맞추어 게임데이터 DB와 입력 파라미터 값들의 입력과 측정자료 수집을 위한 출력으로 이루어져 있다.

3.2 시뮬레이터 동작

제안하는 시뮬레이터는 플레이어가 PC(PC: Player Character)[2]를 직접 조정하는 것이 아니라 아래와 같은 규칙에 따라 PC

2) 플레이어가 직접 조정하는 게임캐릭터



(그림 3) 시뮬레이터 동작 흐름도

액션들이 자동적으로 진행되는 방법을 사용하였다.

(1) 파티(party) 대형: 몬스터를 향하는 쪽에서 전사 캐릭터들이 제일 앞에 선다. 그 뒤는 성직자 캐릭터 그리고 그 뒤는 다른 캐릭터의 순서로 배치된다.

(2) 탐험 액션: 탐험경로는 고향마을에서 출발하여 배경맵의 트리구조의 타 지역 연결정보를 통해 탐험경로가 결정된다. 만약 타 지역이 연결되어 있지 않는 막다른 지역은 배경 맵의 트리구조의 연결정보를 이용하여 탐험하지 않은 지역을 찾아 탐험을 계속한다. 마을에 도착하면 상점에 가서 소유하고 있는 돈을 갖고 필요한 방어구나 무기를 구입한다.

(3) 조우 액션: 인간형 몬스터인 경우는 협상을 시도하고, 비 인간형 몬스터인 경우는, 상대 몬스터의 최대공격력이 PC 대형의 제일 앞쪽에 있는 PC들의 생명력(Hit Points)보다 클 경우는 도망간다.

(4) 전투 액션: 전투시작 바로 직전에 방어 마법이나 공격력 향상 마법을 사용할 수 있는 경우 사용한다. 공격시(時)에는 생명력이 가장 작은 몬스터를 공격 목표로 정한다. 만약 공격액션 시작 바로 직전에 상대 몬스터의 최대공격력이 PC 파티 대형의 제일 앞쪽에 있는 PC들의 생명력보다 클 경우는 도망간다. 방어 시(時)에는 PC 파티 대형의 제일 앞쪽에 있는 PC들이 몬스터들의 근거리 공격의 방어를 담당한다.

(5) 획득한 보물 및 경험점수 나누는 규칙: 몬스터를 죽이고 획득하는 보물이나 경험점수는 파티구성원들이 균등하게 나누어 가진다.

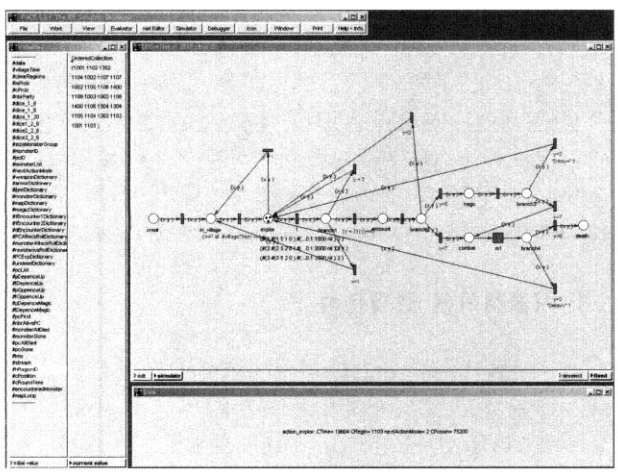
시뮬레이터의 동작은, (그림 3)과 같이, 크게 초기설정 단계와 PC 액션 시뮬레이션 단계로 구분된다. 초기설정 단계에서는 게임메카닉스의 형상과 실험환경을 결정하고 시뮬레이터 DB를 읽어 등장캐릭터와 배경, 그리고 아이템에 대한 초

〈표 2〉 시뮬레이션을 통한 게임메카닉스 품질평가 내역

품질부문	평가 항목	평가방법
무결성	① 게임플레이 흐름/판정의 무결성 ② 게임월드 데이터 관리의 무결성	① 시뮬레이션에서 수집된 PC party의 액션 히스토리 분석 ② 시뮬레이션에서 수집된 게임월드 데이터의 상태 히스토리 분석
최적화	① 게임플레이 흐름/판정의 최적화 ② 게임월드 데이터 구조의 최적화	상동
플레이어/플레이어 밸런스	① 캐릭터 클래스 레벨업 속도	① 캐릭터 클래스별 레벨업 속도 상대 비교
플레이어/게임플레이 밸런스	① PC 액션 분포의 적절성 ② PC 레벨별 물리친 몬스터 분포의 적절성	① PC party별 액션 분포의 상대 비교 ② PC 레벨별 물리친 몬스터 분포 분석
게임플레이/게임플레이 밸런스	① 검/방패와 양손검의 사용 효과 비교	① 검/방패와 양손검에 대한 전사캐릭터 레벨업 속도 상대 비교

기값들을 설정한다. 그리고 고향마을에서 출발하는 PC들을 생성하고 파티를 형성한다. PC 액션 시뮬레이션 단계에서는, 배경맵의 탐험경로를 따라, 탐험액션, 조우액션, 협상액션, 그리고 전투액션을 마을에 도착하기 전까지 반복수행하고 이를 통해 PC들은 보물/치료물약과 경험치를 획득한다. 마을에 도착하면 획득한 돈을 가지고 더 좋은 방어구와 무기 그리고 치료 아이템을 구입하고 필요한 마법을 배우게 된다. 그리고 다시 마을을 떠나 탐험경로를 따라 탐험을 계속한다. 이러한 방식으로 종료 조건이 만족될 때까지 시뮬레이션이 지속된다. 시뮬레이터는 시뮬레이션이 이루어지는 동안, PC들이 새로운 지역에 진입하거나 혹은 새로운 PC의 액션에 대한 사건들이 발생할 때마다 사건의 발생시간과 관련 정보들을 기록한다.

본 논문에서 제안하는 시뮬레이터의 실행 장면은 (그림 4)에 나타나 있다. 제일 상위 윈도우는 메뉴를 제공하고 좌측 윈도우는 시뮬레이션 모델에서 사용하는 변수들의 초기값과 현재 값들을 보여준다. 중앙에 있는 윈도우는 Petri net 모델과 토큰(token)들의 이동을 보여준다.



(그림 4) Petri net 모델을 사용한 시뮬레이터의 실행모습

4. 시뮬레이션 내역

시뮬레이션은, 본 연구에서 제안하는 시뮬레이터의 성능을 보여주기 위해 이루어졌다. 즉 제안하는 시뮬레이터가 게임 플레이 환경에서 게임메카닉스 품질평가에 얼마나 유용한지를 보여주기 위함이다.

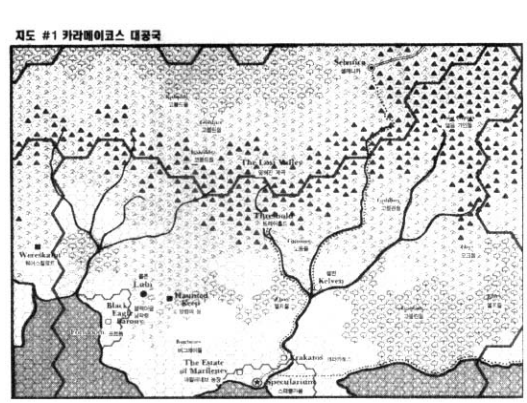
시뮬레이션을 통해 게임메카닉스의 무결성, 최적화, 밸런싱 부문의 품질을 평가하기 위해, 적용된 평가의 항목과 방법에 대한 내역은 <표 2>과 같다.

4.1 실험 환경

시뮬레이션은 1.3GHz 인텔 제온 CPU 2개와 1GB 메모리가 탑재된 윈도우즈2000 워크스테이션에서 이루어졌다.
시뮬레이션에서 사용된 게임DB는, 엑셀파일에 보관되며, 등장 캐릭터는 성직자, 전사, 마법사 클래스를 사용했다. 등장 캐릭터들의 능력적 특징을 비교하면 <표 3>과 같다.

〈표 3〉 성직자, 전사, 마법사 클래스별 능력적 특징 비교

항 목	성직자	전사	마법사
주요 스킬	언데드퇴치, 성직자마법	-	마법사 마법
주요능력치[3]	지혜	힘	지능
사용가능무기	철퇴, 곤봉, 전투해머, 슬링	모든 무기	단검류
사용가능 방어구	모든 방어구	모든 방어구	-

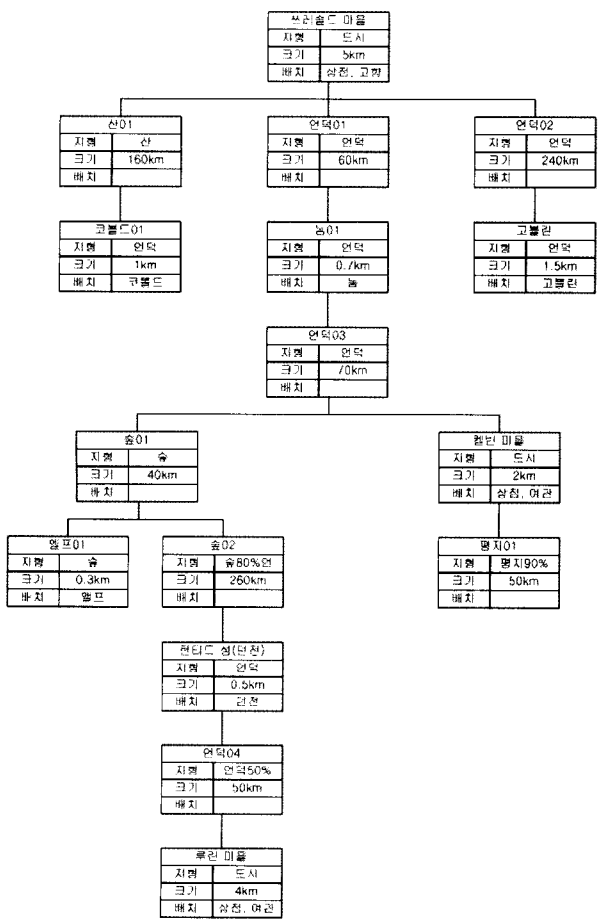


(그림 5) 카라메이코스 대공국 지도

3) 경험치 계산 時 보너스/패널티 규칙에 사용되는 능력치

〈표 4〉 시뮬레이션 실험조건 내역

실험조건	등장 PC	시뮬레이션 종료조건	시뮬레이션 반복횟수	비고
1	성직자 1명	성직자 레벨 14	10회	.방어구/무기 갱신
2	전사 1명	전사 레벨 14	10회	.방어구/무기 갱신
3	성직자 1명, 전사 1명	성직자 레벨 14	10회	.방어구/무기 갱신
4	성직자 1명, 마법사 1명	성직자 레벨 14	10회	.방어구/무기 갱신
5	전사 1명, 마법사 1명	전사 레벨 14	10회	.방어구/무기 갱신
6	성직자 1명, 전사1명, 마법사 1명	성직자 레벨 14	10회	.방어구/무기 갱신
7	검/방패/가죽갑옷의 전사 1명	전사 레벨 14	10회	.초기 방어구/무기유지
8	양손검/가죽갑옷의 전사 1명	전사 레벨 14	10회	.초기 방어구/무기유지



(그림 6) '카라메이코스 대공국' 지도를 근거로 한 배경맵 DB

마법, 무기 및 방어구 그리고 기타 아이템은 등장 캐릭터 클래스에 관련된 것들만 선택하여 DB로 구성하였다. 또한 배경맵은, (그림 5)과 같이, D&D 플레이 룰 북(play-rule book)[14]에 나오는 '카라메이코스 대공국' 지형을 기본으로 하여, 지형종류별로 지역을 분할하고 이 지역들간의 연결 관계를 탐험경로에 따라 설정하여, (그림 6)과 같은, 트리구조의 배경맵 DB를 사용하였다.

배경맵의 각 분할지역 정보에는 지역ID, 지형종류, 지역크기, 주요 배치 정보, 그리고 타지역 연결정보로 구성되어 있다. 특히 타 지역 연결정보는 탐험경로와 주변지형의 특성에 따라 PC들이 선택할 확률을 지정하였다. 또한 배경맵 DB에는 이동몬스터 출현, 몬스터 본거지, 그리고 보물 배치에 대한 정보를 갖고 있다.

4.2 실험방법

본 연구에서 적용된 시뮬레이션은, (그림 6)의 배경맵 DB를 공통적으로 사용하였다. 그리고 실험조건은, <표 4>와 같이, 8가지이고 각 실험조건에서 시뮬레이션을 10번씩 반복 실시하였다. 따라서 총 80회 시뮬레이션을 수행하였다.

실험조건은 등장 PC에 따라 구분되는데 PC가 1명인 경우, 2명인 경우, 그리고 3명인 경우로 나누었다. PC가 1명인 경우에서 마법사인 경우는 제외되었는데 이는 방어능력이 매우 약하여 마법사는 단독으로 모험할 수 없기 때문이다. 나머지 실험조건은 검/방패와 양손검의 착용효과를 비교평가하기 위해 검/방패/가죽갑옷을 착용한 전사 1명인 경우 그리고 양손검/가죽갑옷을 착용한 전사 1명인 경우이다. 시뮬레이션 종료조건은 'PC들 중 한 캐릭터가 최대레벨 14에 도달할 경우'로 정했다.

실험조건 1~6에서는 PC들이 마을에 도착하면 소지한 돈이나 보물로 무기나 방어구를 갱신하는 경우이고 실험조건 7~8에서는 PC들이 초기에 착용한 무기나 방어구를 계속해서 사용하는 경우이다.

5. 시뮬레이션 결과분석

각 시뮬레이션이 수행될 때, 시뮬레이터는, <표 5>와 같이, 새로운 지역에 진입하거나, 새로운 액션이 시작되거나, 혹은 몬스터를 만나는 사건(event)이 발생하면 이 사건에 대한 시간과 상황 정보를 기록한다.

사건에 대한 정보는 사건이 있어난 지역, 지역의 진입 및 이탈 시간, PC 파티의 액션모드(1:마을, 2:탐험, 3:이동몬스터 조우, 4:배치몬스터 조우, 5:협상, 6:PC죽음, 7:전투), 액션시작 및 끝 시간, 조우 몬스터 ID, 죽인 몬스터 수, 그리고 PC당

〈표 5〉 성직자 1명 실험환경의 실험에서 시뮬레이터가 기록한 사건들

사건 ID	지역ID	지역진입시간 (턴)	액션모드	액션시작시간 (턴)	액션 끝 시간 (턴/라운드)	지역이탈시간 (턴)	조우 몬스터 ID	죽인 몬스터 수	각 PC가 획득한 경험치
1	1001	0	1	0	87	87			
2	1101	88	2	88	111				
3	1101	88	3	112	1				
…	…	…	…	…	…	…	…	…	…
81	1101	88	7	490	5		4030	1	14
82	1101	88	2	491	495				
…	…	…	…	…	…	…	…	…	…
423	1101	88	2	2684	2692	2692			
424	1201	2693	2	2693	2694				

획득한 경험치이다.

본 논문에서 제시하는 시뮬레이션 분석결과는 실험동안 사건들을 기록한 텍스트파일들을 엑셀 파일에서 불러온 후 엑셀 함수들을 사용하여 통계적인 자료 분석을 한 후 그 결과를 나타낸 것이다.

5.1 무결성(faultlessness)

무결성이란 게임메카닉스가 오류 없이 정확하게 동작하는 측면의 품질을 의미한다. 본 시뮬레이션을 통해 평가하는 부분은 게임플레이의 흐름과 판정의 무결성과 게임월드 데이터 관리의 무결성이다.

5.1.1 게임플레이 흐름/판정의 무결성 평가

게임플레이의 흐름과 판정과정에서 오류가 있는지 탐지하는 1차적인 방법은, <표 5>와 같이, 시뮬레이터가 기록한 사건들을 조사하는 것이다. 구체적으로 PC 파티 일행들이 주어진 탐험경로와 이동속도가 정확한지 조사하고 이동몬스터나 배치몬스터가 지형과 위치에 맞게 정확하게 조우되었는지 조사하고 PC 파티 일행들의 액션모드 흐름과 액션지속시간이 정확한지 조사하고 그리고 경험치 계산이 정확히 이루어졌는지를 조사한다. 관련 평가내역은 <표 6>와 같으며, 시뮬레이션을 통한 게임플레이 흐름/판정에 대한 평가결과는 모두 정상으로 나타났다.

〈표 6〉 게임플레이 흐름과 판정의 무결성 평가내역

세부평가 항목	평가방법	평가결과
탐험경로	배경맵 DB내용과 비교조사	정상
이동속도	지역별 이동시간, 지역크기 정보를 통해 이동속도를 계산하여 턴당 20미터와 일치하는지 조사	정상
몬스터 조우	조우된 몬스터들이 해당 지역 몬스터 출현표에 있는 몬스터인지 조사	정상
액션흐름	마을, 탐험, 조우, 협상, 전투, 죽음과 같은 액션모드의 흐름이 정확하게 진행되는 지 조사	정상
액션지속시간	각 액션의 지속시간이 적절한지 조사	정상
경험치 계산	죽인 몬스터의 ID와 전투위치, 그리고 죽인 몬스터수를 근거로 경험치를 계산하여 정확한가를 조사	정상

게임플레이 흐름과 판정의 무결성을 평가하기 위한 2차적인 방법은 시뮬레이션 중에 Petri net 토큰들의 흐름과 속성값들, 시뮬레이션 변수값들, 그리고 출력 메시지 내용들을 조사하여 오류존재 여부를 검사하였다. 2차적인 방법의 검사결과에서도 정상으로 나타났다.

5.1.2 게임월드 데이터 관리의 무결성 평가

게임월드 데이터 관리의 무결성은 시뮬레이션 중에 시뮬레이션 변수들의 현재 값들의 변화를 관찰하여 오류를 검사하는 것을 기본으로 하고 나머지는 토큰들의 흐름과 속성 값들을 관찰하여 오류를 검사한다. 이러한 방법을 통한 검사결과는 모두 정상으로 나타났다.

5.2 최적화

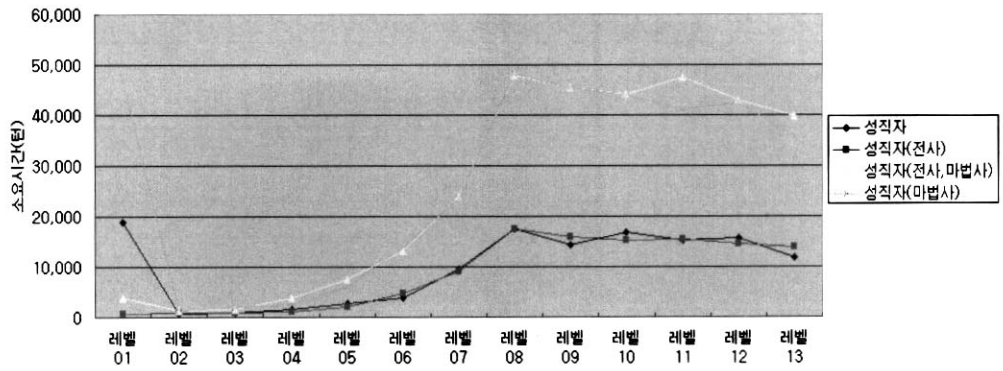
게임메카닉스의 최적화 평가는 시스템차원에서 필요 없는 기능이나 동작들이 있는지를 파악하고 병목현상이 일어나는 부분이 없는지를 검사하는 것이다.

5.2.1 게임플레이 흐름의 최적화

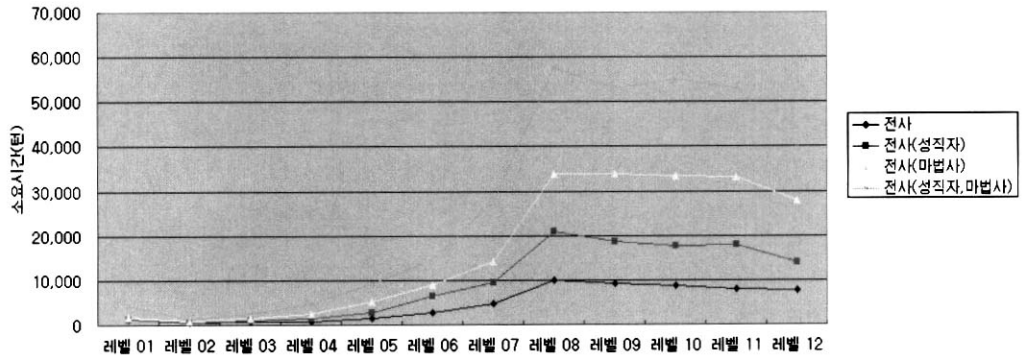
게임플레이 흐름의 최적화는 사용된 Petri net 토큰들의 움직임을 관찰하여 검사할 수 있다. 1차적으로는 토큰들의 움직임을 통해 불필요한 부분이 없는지를 검사하고 2차적으로는 각 트랜지션(transition)의 실제 처리시간을 관찰하여 병목현상이 나타나는지를 검사하였다. 검사결과 토큰들의 움직임에서는 불필요한 부분이 발견되지 않았다. 하지만 공통적으로 데이터를 초기화하는 트랜지션에서 처리시간이 상대적으로 많이 소요되었다. 이 트랜지션은 등장 PC에 대한 정보와 배경맵 DB 정보의 초기화 과정을 처리하는 부분으로 최적화의 필요성이 발견되었다.

5.2.2 게임월드 자료구조의 최적화

게임월드 데이터 구조의 최적화 평가는 게임월드에서 사용되는 자료구조가 게임플레이가 가능하도록 충분한지 또한



(그림 7) PC 파티별 성직자의 능력레벨 소요시간 분포



(그림 8) PC 파티별 전사의 능력레벨 소요시간 분포

불필요한 부분은 없는지를 평가하는 것이다. 이를 위해, 시뮬레이션 변수들의 현재 값들의 변화를 관찰하여 필요 없는 부분이 없는지를 검사하였다. 검사결과 배경지형의 속성을 나타내는 자료구조에서 「다운링크 탐험완료 요구지역」필드는 불필요한 것으로 관찰되었다.

또한 PC의 캐릭터 시트의 인벤토리(inventory) 자료구조에서도 「사용가능마법」필드와 「두루마리 마법」필드는 하나로 통합할 수 있는 것으로 분석되었다.

5.3 플레이어/플레이어 밸런스

플레이어/플레이어 밸런스는, 게임플레이에서, PC 클래스(player character class)간의 게임플레이 공평성을 의미한다. RPG(Roll Playing Game)장르에서는 PC의 능력레벨의 상승속도의 공평성이 제일 중요하기 때문에 시뮬레이션 결과분석도 여기에 초점을 맞추었다.

5.3.1 성직자 클래스 능력레벨 소요시간 분포

(그림 7)의 성직자 클래스의 결과분석에 의하면, 성직자 단독으로 모험을 한 경우와 성직자와 마법사가 파티를 이루어 모험을 한 경우는 능력레벨 1에서 매우 큰 시간이 소요되었다. 특히 성직자와 마법사의 파티를 이룬 경우에서는 약 47,130턴 정도의 시간이 소요되어 다른 어떤 레벨 소요시간보다 길었다. 이는 두 캐릭터 클래스 모두 근거리 공격력이 약하고 또한 획득한 경험치를 모두 균등하게 배분하는 규칙의 영향인 것으로 판단된다. 반면에 전사가 포함된 party에서는 레벨별 소요시간 분포 그래프들이 정상적이고 안정된 증가곡선형을 나타내었다. 또한 성직자 단독모험과 성직자/전사 모험은 능력레벨 2이상부터는 서로 비슷한 소요시간 분포가 나타났고 성직자/마법사 모험과 성직자/전사/마법사 모험도 동일한 현상이 나타났다.

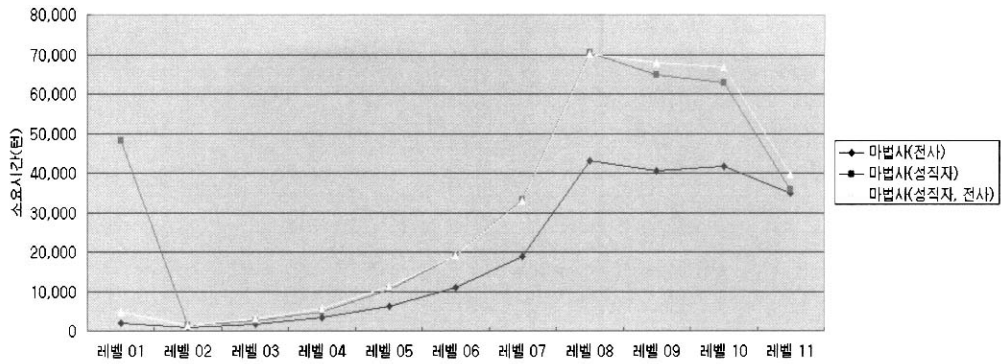
5.3.2 전사 클래스 능력레벨 소요시간 분포

(그림 8)의 전사 클래스의 결과분석에 의하면, 전사 단독모험 혹은 파티를 이룬 모험에서 모두 정상적인 증가곡선형이 나타났다. 이는 전사의 강한 근거리공격력이 몬스터를 죽여서 경험치를 획득하는데 큰 역할을 수행한 것으로 나타났다. 파티를 구성하여 여러 PC들이 모험할 때는 전사가 포함되어야 하위레벨에서 능력레벨의 상승이 빠르고 안정적으로 이루어짐을 알 수 있다.

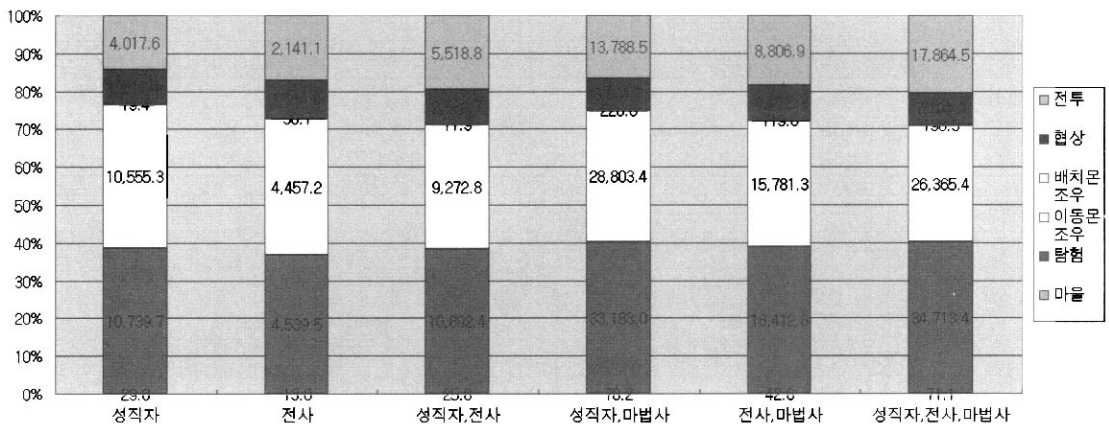
또한 전체적으로 전사의 단독모험이 가장 빠른 속도로 능력레벨이 상승하였는데 이는 직접 몬스터를 죽이는 역할을 전사가 중심적으로 수행한다는 것을 나타낸다. 파티를 구성하여 여러 PC들이 함께 모험을 한 경우에는, 비록 전사가 몬스터를 죽이는 데 중심적인 역할을 할지라도, 경험치 배분을 PC 모두에게 균등하게 하는 규칙의 영향으로 전사는 단독모험보다는 상대적으로 작은 경험치를 획득한다. 이로 인해 능력레벨 소요시간이 늘어난 것으로 판단된다.

5.3.3 마법사 클래스 능력레벨 소요시간 분포

(그림 9)의 마법사 클래스의 결과분석에 의하면, 전사가



(그림 9) PC 파티별 마법사의 능력레벨 소요시간 분포



(그림 10) PC 파티별 액션 빈도수 분포

포함된 파티를 이룬 모험에서 모두 정상적인 증가곡선이 나타났다. 또한 마법사/성직자 모험과 마법사/성직자/전사 모험은 능력레벨 2이상부터는 서로 비슷한 소요시간 분포로 나타났다.

또한 전체적으로 마법사/전사 모험이 가장 빠른 속도로 능력레벨이 상승하였다.

5.4 플레이어/게임플레이 밸런스

플레이어/게임플레이 밸런스는 PC의 능력 레벨과 게임 난이도 레벨사이의 밸런스를 의미한다. 결과분석은 전체적인 측면에서 PC 모험의 액션들이 적절한 비율로 분포되었는지를 분석하고 PC 능력레벨 측면에서 물리친 몬스터 내역을 분석하였다.

5.4.1 PC 액션 분포의 적절성

(그림 10)의 모험별 액션 빈도수 분포에 의하면, 모든 PC 파티경우에서 탐험액션 비율이 약 40%로 가장 높게, 그 다음은 이동몬스터 조우 액션이 약 30%~35%, 그 다음은 전투액션 이 약 15%~21%, 그 다음은 협상 액션이 약 10%로 나타났다. 배치몬스터 조우 액션외 다른 액션들의 비율은 적절한 것으로 판단된다. 하지만 배치몬스터의 조우 비율이 이동몬스터의 조우 비율이 비하여 매우 낮게 나타나 중소규모의 몬스터 본거지들을 배경맵에 더 추가할 필요가 있는 것으로 분석된다.

조우액션 비율이 전투액션과 협상액션 비율들의 합보다 높은 것은 PC 일행들이 자신들의 HP보다 공격력이 큰 몬스터들을 만나면 도망가는 규칙으로 인해 나타난 현상으로 분석된다.

5.4.2 PC 레벨별 물리친 몬스터 분포의 적절성

성직자와 전사의 단독모험에서 물리친 몬스터 수의 분포들이 <표 7>과 <표 8>에 나타나 있다. 이들 분포에서 몬스터 이름의 배열순서는 몬스터의 강함을 기준으로 하였다. 따라서 위쪽은 약한 몬스터이고 아래쪽은 강한 몬스터이다. 거인쥐~거대지네가 가장 약한 몬스터이고, 화이트 드래곤이 가장 강한 몬스터이다.

물리친 몬스터 수의 분포에 의하면 해골, 게 거미, 구울, 그리고 좀비 몬스터의 수가 상대적으로 매우 작았는데 이는 이동몬스터 조우표의 설정에서 밸런스가 맞지 않는 것으로 나타났다. 다른 몬스터의 경우에는 분포가 적절한 것으로 판단되었다.

전사와 성직자의 물리친 몬스터 분포를 비교하면, 전사가 성직자에 비해 강한 몬스터들의 물리친 수가 많았다. 예를 들면 가장 강력한 화이트 드래곤인 경우 전사의 모험에서는 총 74.6마리를 죽였고, 성직자 모험에서는 총 16마리를 죽였다.

5.5 게임플레이/게임플레이 밸런스

게임플레이/게임플레이 밸런스는 게임플레이 요소(예: 갑옷, 경험치)들 간의 밸런스 또는 속성(예: 가격, 공격력)들 간

〈표 7〉 성직자 단독모험에서 능력 레벨별 물리친 몬스터 수 분포

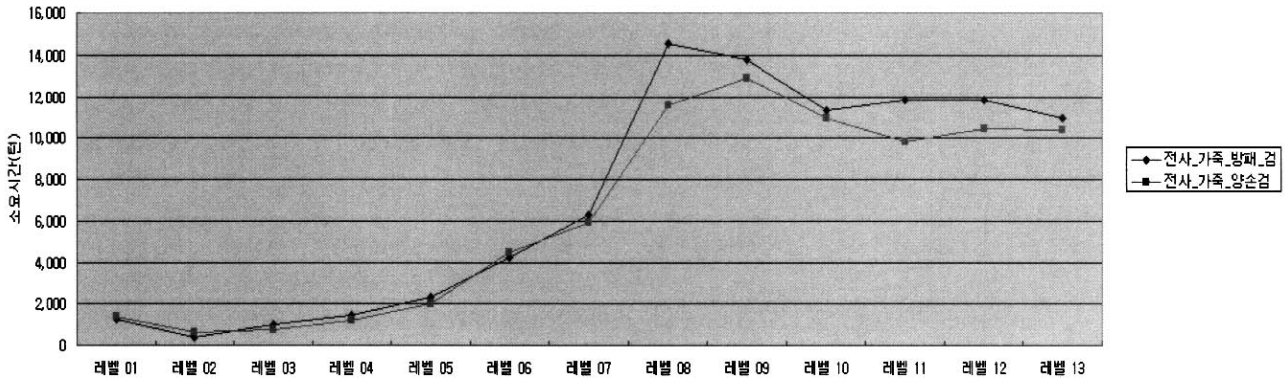
(단위: 명 또는 마리)

몬스터이름	레벨 01	레벨 02	레벨 03	레벨 04	레벨 05	레벨 06	레벨 07	레벨 08	레벨 09	레벨 10	레벨 11	레벨 12	레벨 13	합계
거인쥐(던전)	0.0	0.0	0.0	0.5	1.5	12.8	26.0	34.4	57.4	66.2	41.9	57.5	38.4	336.6
코볼드	0.0	3.1	6.6	10.7	12.2	18.1	35.4	76.9	56.7	86.4	68.7	72.2	100.5	547.5
거인벌	0.0	5.1	16.8	36.9	75.3	125.5	297.2	524.5	512.7	601.1	545.2	530.9	549.0	3,820.2
거대박쥐(던전)	25.2	1.4	4.7	9.8	11.7	13.7	41.4	102.0	73.0	79.1	73.6	89.2	86.8	611.6
거대지네	35.4	0.7	0.0	0.6	3.1	2.2	7.6	51.6	26.2	41.7	23.1	40.5	6.1	238.8
고블린	0.0	4.1	4.9	15.4	36.8	38.3	102.4	117.9	161.8	170.0	181.1	136.9	136.3	1,105.9
엘프	0.0	5.0	4.7	10.0	25.3	35.9	86.3	108.8	108.2	136.2	123.1	122.4	124.0	889.9
노움	0.0	2.3	3.3	8.0	18.0	24.9	66.4	73.1	98.0	101.8	111.5	82.9	105.5	695.7
오크	0.0	4.2	4.8	13.0	24.0	42.1	94.8	142.0	191.6	217.5	209.1	205.1	158.2	1,306.4
산적	0.0	2.8	10.7	26.5	45.6	98.4	194.4	344.4	383.7	355.4	363.3	368.9	382.6	2,576.7
서터지	72.6	6.5	2.2	12.3	23.5	25.7	60.3	131.1	189.8	161.1	176.1	179.1	106.8	1,147.1
해골	0.4	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
불꽃딱정벌레(던전)	0.0	0.0	0.3	0.5	1.2	0.6	6.0	29.4	20.4	32.1	19.5	36.8	6.7	153.5
맷돼지	0.0	1.2	3.1	5.5	13.5	13.9	46.5	102.8	79.6	111.8	86.2	98.2	75.5	637.8
트로글로다이트	0.0	0.0	0.7	1.1	2.8	7.0	18.7	29.6	46.6	46.8	44.2	47.5	37.2	282.2
거대메뚜기(던전)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0	0.3
게 거미	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.2
구울	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
좀비(던전)	0.0	0.0	0.0	0.0	0.6	0.0	0.0	8.1	5.1	4.1	3.2	6.1	1.9	29.1
게코도마뱀	0.0	1.1	1.2	5.0	8.4	9.9	29.9	72.7	51.6	68.5	57.9	69.8	54.5	430.5
큰개미	0.0	0.0	0.2	0.3	0.4	0.3	2.4	7.8	4.4	9.9	8.5	11.3	1.5	47.0
워울프	0.0	0.7	2.7	7.3	12.2	23.0	46.6	97.1	100.1	114.3	112.2	111.8	111.1	739.1
화이트드래곤	0.0	0.0	0.0	0.0	0.2	0.2	1.2	0.9	2.1	3.1	2.1	2.1	4.1	16.0

〈표 8〉 전사 단독모험에서 능력 레벨별 물리친 몬스터 수 분포

(단위: 명 또는 마리)

몬스터이름	레벨 01	레벨 02	레벨 03	레벨 04	레벨 05	레벨 06	레벨 07	레벨 08	레벨 09	레벨 10	레벨 11	레벨 12	레벨 13	합계
거인쥐(던전)	0.0	0.0	0.5	0.8	10.3	6.6	23.7	59.1	32.2	51.3	32.6	31.2	32.5	280.8
코볼드	1.3	1.7	3.6	6.9	10.4	22.1	38.1	109.0	136.0	123.2	91.5	142.6	96.0	782.4
거인벌	0.0	14.5	24.9	38.6	60.6	158.0	211.7	511.6	549.7	555.7	455.5	511.6	542.1	3,634.5
거대박쥐(던전)	7.6	0.3	2.3	5.2	7.5	14.9	37.2	74.9	63.3	69.9	58.2	64.7	52.4	458.4
거대지네	0.0	0.0	1.0	0.5	0.6	5.1	12.2	32.0	12.1	31.0	38.3	6.8	1.6	141.2
고블린	0.0	0.7	8.8	9.7	37.6	61.2	86.8	165.2	198.7	146.0	91.8	179.7	167.1	1,153.3
엘프	8.6	2.4	9.6	10.7	15.8	32.9	50.6	110.2	129.5	122.8	86.6	112.6	153.5	845.8
노움	0.0	2.7	7.7	7.3	21.5	28.7	61.7	117.0	144.7	123.0	68.5	92.3	140.3	815.4
오크	3.4	12.6	12.8	12.1	19.8	53.6	72.5	204.4	220.0	160.7	139.3	180.8	186.4	1,278.4
산적	13.8	11.3	19.6	24.6	57.1	102.1	189.7	390.2	376.4	384.2	349.2	406.1	380.1	2,704.4
서터지	19.5	0.0	10.1	13.2	13.6	43.3	75.5	157.1	126.2	152.8	151.9	156.2	180.9	1,100.3
해골	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.4
불꽃딱정벌레(던전)	0.0	0.0	0.8	0.4	9.6	5.7	30.7	48.6	8.7	47.2	62.6	18.2	3.0	235.5
맷돼지	0.0	1.1	7.3	3.9	13.1	20.0	42.3	97.7	70.5	90.6	98.6	66.4	57.5	569.0
트로글로다이트	0.0	0.4	1.7	2.2	4.2	18.8	32.1	63.1	52.4	71.2	67.5	49.7	65.2	428.5
거대메뚜기(던전)	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.6
게 거미	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.6
구울	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
좀비(던전)	0.0	0.0	0.0	0.0	0.0	0.0	8.2	14.6	1.0	1.9	22.7	1.1	0.0	49.5
게코도마뱀	0.0	1.3	5.0	3.9	10.4	18.4	53.7	113.7	87.1	95.1	115.1	79.6	73.8	657.1
큰개미	0.0	0.0	0.0	0.5	0.7	2.1	12.7	23.5	4.5	25.7	37.4	7.3	0.3	114.7
워울프	0.0	3.0	4.5	8.1	22.2	41.0	77.2	180.1	164.5	165.9	171.6	165.6	181.1	1,184.8
화이트드래곤	0.0	0.0	0.0	0.4	0.6	2.3	3.5	6.9	12.0	10.0	8.5	10.8	19.6	74.6



(그림 11) 전사 능력레벨별 소요시간 분포

의 밸런스를 의미한다. 본 결과에서는 검/방패를 사용한 경우와 양손검을 사용한 경우의 효과를 비교 분석하였다. 이는 방어도와 공격력의 업그레이드 가치를 상대적으로 비교하는 것을 목적으로 한다. 여기서 사용된 실험조건은 검/방패/가죽갑옷을 처음부터 끝가지 사용하는 전사 단독모험과 양손검/가죽갑옷을 처음부터 끝까지 사용하는 전사 단독모험이다.

5.5.1 검/방패와 양손검의 사용효과 비교

전사 능력레벨별 소요시간 분포를 (그림 11)을 통해 살펴보면, 레벨 1~7은 비슷하고 레벨 8부터는 양손검을 사용한 경우가 레벨업 속도가 더 높았다. 이는 레벨업 속도가 방어도의 상승의 효과 보다는 공격력 상승의 효과에 더 큰 영향을 받는다는 것을 의미한다.

〈표 9〉 검/방패를 사용하는 전사의 능력 레벨별 물리친 몬스터 수의 분포

(단위: 명 또는 마리)

몬스터이름	레벨 01	레벨 02	레벨 03	레벨 04	레벨 05	레벨 06	레벨 07	레벨 08	레벨 09	레벨 10	레벨 11	레벨 12	레벨 13	합계
거인쥐(던전)	0.0	0.0	0.7	0.0	2.9	7.1	23.1	33.9	19.0	39.1	35.8	43.2	30.4	235.2
코볼드	2.6	3.9	8.0	11.2	12.6	15.9	27.7	157.7	101.6	82.2	102.5	77.0	99.3	702.2
거인벌	0.0	4.3	21.4	53.8	66.8	135.6	244.6	585.3	527.8	489.6	513.3	485.2	548.8	3,676.5
거대박쥐(던전)	4.6	2.6	2.6	7.8	14.5	24.1	41.6	101.5	86.2	67.8	78.3	74.9	64.2	570.7
거대지네	0.0	0.0	3.2	3.9	8.1	18.0	35.7	8.9	30.0	29.0	15.0	36.9	17.2	205.9
고블린	0.0	2.0	8.1	15.1	14.8	19.4	26.2	178.2	123.7	127.3	174.3	100.9	110.0	900.0
엘프	4.0	2.2	8.8	8.1	10.4	17.6	50.9	194.7	157.4	125.4	173.0	111.5	139.5	1,003.5
노움	0.0	0.8	4.1	11.1	9.0	19.0	26.9	90.5	72.8	88.8	139.6	82.4	120.1	665.1
오크	3.1	3.9	5.3	8.4	22.7	48.3	65.3	204.5	183.1	187.4	247.2	172.2	205.6	1,357.0
산적	7.8	5.4	12.7	24.8	56.6	101.7	187.9	401.0	379.4	388.1	379.2	383.4	447.2	2,775.2
서터지	8.4	3.1	14.2	21.7	19.9	47.3	73.7	165.1	151.9	151.7	188.2	190.2	150.4	1,185.8
해골	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	1.0	1.4	2.7
불꽃딱정벌레(던전)	0.0	0.0	2.2	8.0	10.5	19.8	31.2	14.2	48.5	34.0	18.5	54.4	24.1	265.4
맷돼지	0.0	1.4	5.9	9.5	17.5	32.3	49.9	91.1	108.4	94.6	87.7	122.3	98.7	719.3
트로글로다이트	0.0	0.0	1.7	4.8	9.1	11.5	17.2	67.2	54.1	55.9	55.1	51.8	59.9	388.3
거대메뚜기(던전)	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.4
게 거미	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
구울	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
좀비(던전)	0.0	0.0	0.0	0.0	0.0	1.5	8.7	0.0	7.8	6.8	0.0	5.1	3.4	33.3
게코도마뱀	0.0	1.3	2.6	7.0	15.2	30.4	50.9	64.8	87.5	94.3	85.0	89.5	94.6	623.1
큰개미	0.0	0.0	0.2	1.7	2.8	7.2	13.4	2.8	16.1	15.4	7.5	30.2	13.7	111.0
웨울프	0.0	2.0	7.7	10.9	20.8	36.3	56.2	163.5	143.5	147.5	154.8	152.1	148.9	1,044.2
화이트드래곤	0.0	0.0	0.0	0.0	0.1	0.5	1.2	3.2	2.7	3.0	3.9	3.1	3.8	21.5

〈표 10〉 양손검을 사용하는 전사의 능력 레벨별 물리친 몬스터 수의 분포

(단위: 명 또는 마리)

몬스터이름	레벨 01	레벨 02	레벨 03	레벨 04	레벨 05	레벨 06	레벨 07	레벨 08	레벨 09	레벨 10	레벨 11	레벨 12	레벨 13	합계
거인쥐(던전)	0.0	0.0	0.6	0.8	6.7	10.5	11.0	19.0	25.8	39.5	29.3	21.0	43.8	208.0
코볼드	1.6	2.5	3.3	7.4	16.4	24.5	22.4	160.5	136.8	88.9	85.1	93.5	135.3	778.2
거인벌	0.0	13.3	13.0	38.2	77.8	140.3	229.4	440.9	550.0	479.3	474.5	505.4	507.1	3,469.2
거대박쥐(던전)	6.6	1.0	3.7	5.6	12.6	25.5	43.6	66.8	84.7	69.9	55.6	62.2	59.0	496.8
거대지네	0.0	1.5	5.4	7.3	12.2	17.7	21.6	6.5	34.9	38.8	0.0	19.1	45.3	210.3
고블린	0.0	2.5	6.4	4.6	14.4	29.0	47.9	166.4	107.8	97.9	108.6	109.4	97.6	792.5
엘프	8.0	3.3	1.9	12.4	9.4	21.8	45.1	134.4	131.5	126.3	127.2	133.1	104.7	859.1
노움	0.0	0.8	1.3	7.0	9.9	15.3	37.9	103.9	98.8	80.6	107.7	94.7	91.6	649.5
오크	5.2	5.4	6.4	10.8	21.2	41.2	50.4	165.7	145.3	142.9	152.7	149.7	187.8	1,084.7
산적	9.3	5.8	14.6	25.5	56.5	97.3	171.6	416.7	387.8	374.2	380.0	349.1	365.1	2,653.5
서터지	5.8	11.6	2.1	15.1	22.3	40.8	88.4	153.8	143.9	145.9	124.2	149.5	155.0	1,058.4
해골	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	2.0
불꽃딱정벌레(던전)	0.0	1.2	3.1	5.6	12.2	16.5	35.3	5.4	38.7	58.9	0.0	32.9	50.1	259.9
맷돼지	0.0	2.8	8.2	13.0	18.2	36.3	50.3	71.5	91.0	93.6	64.2	90.0	107.5	646.6
트로글로다이트	0.0	1.0	1.0	2.8	8.1	18.2	20.6	48.5	55.3	49.1	56.5	65.7	54.0	380.8
거대메뚜기(던전)	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.8
게 거미	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
구울	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
좀비(던전)	0.0	0.0	0.0	0.0	0.0	6.6	6.2	0.7	2.5	16.2	0.0	0.7	15.6	48.5
게코도마뱀	0.0	1.2	4.7	8.1	14.9	41.9	70.1	64.9	109.0	118.5	65.9	95.4	102.1	696.7
큰개미	0.0	0.0	0.8	2.7	2.6	9.8	20.3	2.8	21.5	28.4	0.0	13.3	34.0	136.2
웨울프	0.0	3.7	6.2	11.0	21.4	53.6	79.1	152.1	166.3	171.1	179.2	174.4	188.3	1,206.4
화이트드래곤	0.0	0.0	0.0	0.2	0.3	0.9	2.7	5.8	7.5	5.6	7.2	8.5	7.7	46.4

전사의 능력 레벨별 물리친 몬스터 수의 분포를 <표 9>와 <표 10>을 통해 살펴보면, 양손검을 사용한 경우에서 강한 몬스터를 죽인 수가 높다는 것을 알 수 있다.

6. 결 론

게임개발과정은, 시행착오를 반복하여 점진적으로 목표 게임의 형상을 결정하며 개발하는 특징으로 인해 게임의 품질평가 및 분석은 알파버전 이상 구현 후 테스트단계에서 주로 이루어져 왔다. 소프트웨어 공학이론에 의하면 테스트단계에서 발견된 문제 해결의 비용은 설계단계에서의 문제해결 비용보다 일반적으로 100~200배 정도 높다. 이로 인해 품질개선 활동이 위축되고 소극적이기 쉽다.

본 논문은 설계단계에서 공학적 품질 분석을 효과적이고 경제적으로 할 수 있는 게임메카닉스의 시뮬레이터를 구현하여 제안하였다. 제안하는 시뮬레이터는 모델링에서 Petri net과 Smalltalk를 사용하기 때문에 경제적으로 정확한 모델을 구축할 수 있으며 또한 게임메카닉스의 형상변경을 유연하게 수용한다. 그리고 게임유저의 플레이 행동패턴과 게임에서 실제 사용하는 게임배경맵, 캐릭터 클래스 속성들, 무기/방어/마법/기타 아이템들에 대한 게임데이터 DB를 사용하여 시뮬레이션하기 때문에 실제적인 플레이환경을 반영한 평가결과를 얻을 수 있다. 그리고 D&D 게임메카닉스의 품질평가 및 분석결과들을 통해, 제안하는 시뮬레이터가 게임메카닉스

의 무결성, 최적화, 밸런스의 품질을 전체적이고 다양한 측면에서 평가할 수 있음을 보여주었다.

본 연구결과의 시뮬레이터는 주요 행동 규칙을 부여해서 인공적으로 구현한 PC의 액션 흐름을 사용하였다. 앞으로 실제 게임유저들의 플레이 특성을 완벽하게 반영한 PC 액션 흐름을 사용하는 시뮬레이터의 개발연구가 필요할 것으로 생각된다.

참 고 문 헌

[1] 문화관광부 한국게임산업개발원, "2004 대한민국 게임백서", (재)한국게임산업개발원, 2004.

[2] Steve McConnell, "Professional Software Development: Shorter Schedules, Higher Quality Products, More Successful Projects, Enhanced Careers," Pearson Education, Inc., 2004.

[3] Andrew Rollings, Dave Morris, "Game Architecture and Design," Coriolis, 2000.

[4] 장희동, "분석 및 설계를 위한 게임메카닉스 시뮬레이션 방법들", 방송공학회지 제10권 1호, pp.123-132, 2005.

[5] 임성진, "Game Balancing in Game Design," Proceedings of KGDC 2003, 2003.

[6] 김상준, 장희동, 김경식, "Petri Nets를 이용한 게임시스템 시뮬레이션", 게임산업저널, 2004년 봄호(통권 3호), 한국게임산업개발원, 2004.

[7] James L. Peterson, "Petri Nets," Computing Surveys, Vol.9, No.3, September, 1977.

[8] Claude Girault and Rüdiger Valk, "Petri Nets for System Engineering: a Guide to Modeling, Verification, and Application," Springer, 2002.

[9] Lewis Pinson, Richard Wiener, "An introduction to Object-Oriented Programming and Smalltalk," Addison-Wesley Publishing Company, 1989.

[10] Chris Crawford, "My Definition of Game," The Journal of Computer Game Design, Vol.4, 1998.

[11] Richard Rouse Ⅲ, "Game Design Theory & Practice," Wordware, 2000.

[12] IBE Simulation Engineering GmbH, "PACE User Manual PACE Version 5.0," IBE GmbH, 2003.

[13] D&D Homepage: http://www.wizards.com/.

[14] 유수현 외 4인 번역, "Dungeons & Dragons 한글판 룰북", (주)커뮤니케이션, 1995.

장 희 동

e-mail : dooly@office.hoseo.ac.kr

1984년 계명대학교 수학과(학사)

1987년 한국과학기술원 응용수학과(석사)

1995년 포항공과대학 수학과(박사)

1987년~1997년 한국전자통신연구소 영상통신연구실 선임연구원

1998년~2002년 숭의여자대학 컴퓨터게임과 조교수

2003년~현재 호서대학교 컴퓨터공학부 게임공학과 조교수

관심분야 : 게임메카닉스 설계, 게임밸런싱, 게임알고리즘