

스크립트 DB를 이용한 MMORPG의 게임 시나리오 개발

송현주⁰, 이대웅

상명대학교

대학원 컴퓨터과학과 게임학전공 박사과정⁰, 소프트웨어대학 디지털미디어학부

dimfe@nate.com, rhee219@smu.ac.kr

A MMORPG Game Scenario Development with Script DB

Song Hyun Joo⁰, Dae Woong Rhee

SangMyung University

Ph.D. Course, Dept. Computer Science, Graduate School⁰,

Dept. Digital Media, Graduate School of Software

요 약

게임은 사용자가 선택하고 행동해야 하는 양방향성을 지닌다. 이런 측면에서 게임 시나리오는 단순한 스토리의 전달을 넘어 사용자를 훈련하고 진행을 유도하는 역할을 하게 된다. 하지만 게임 시나리오는 내용이 방대하고 각 사건 단위별로 연계가 되어 실제 제작 시 어려움이 따랐다. 본 논문에서는 최소 단위인 스크립트를 DB화해 이를 바탕으로 퀘스트를 작성하고 이를 다시 퀘스트 DB로 만드는 방법을 제안한다. 스크립트는 형태적으로 텍스트형, 그래픽형, 사운드형으로 분류하고 이것을 퀘스트의 구조에 맞춰 배치한다. 이 방법을 사용하면 기존 시나리오를 재사용, 재활용 할 수 있으며 제작자에 따라 시나리오의 퀄리티가 일정하지 않던 단점을 극복할 수 있다.

ABSTRACT

The game has both directionalities that users are to choose and act accordingly. In this regard, the game scenario goes beyond just conveying the story, and trains users and guide them in playing the game. However, this game scenario is huge in size and every event is linked with each other so that it was rather difficult to have it made in practice. A study proposes that a script, the minimum unit, is built into a database, and based on which, quests are made, then the resulting quests are built back into a database. Scripts are classified into text, graphic and sound type and these are positioned in accordance with the quest structure. With this method, one can re-use the existing scenarios and can overcome the negatives of irregular qualities of scenarios.

Keyword : Storytelling, Script DB, Game scenario

1. MMORPG의 퀘스트 DB 필요성

인터넷과 저장매체의 발전으로 온라인 기반형 게임이 많이 등장하고 있다. 다수의 사용자가 동시에 접속해 게임을 즐기는 MMORPG는 결말에 해당하는 부분이 없다는 특징 때문에 기존 게임과는 차이가 있다.

화려한 영상과 그래픽 효과, 이야기의 진행 구도를 볼 때 MMORPG의 시나리오는 영화나 애니메이션과 비슷하다. 하지만 게임의 특징인 양방향성 때문에 영화나 애니메이션의 연구를 그대로 적용하기는 어렵다. 게임은 사용자가 선택하고 행동하지 않으면 아무것도 진행하지 않는다. 이를 게임의 양방향성이라 한다[1]. MMORPG에서는 사용자의 아바타가 게임 내 존재하지 않아도 시간은 간다. 현실적인 시간관념에 의해 양방향성은 더욱 강조된다. 그래서 시나리오는 단순한 스토리의 전달에서, 사용자에게 진행을 유도하는 역할을 맡는다. 그 중에도 직접적으로 사용자에게 진행을 유도하는 방법으로 퀘스트가 있다. 퀘스트는 게임 안에서 주어진 임무 또는 목적을 위해 수행하는 행위를 말하는데[2], 예를 들어 '소를 잡아오면 가죽을 주겠다' 라는 퀘스트가 있다고 하자. 이 경우 퀘스트의 목적은 가죽을 얻는 것이며 우리는 이 퀘스트로 사용자의 행동을 유도한다. 하지만 사용자는 퀘스트에 담긴 스토리를 통해 퀘스트의 내용, 즉 소를 잡아야 한다는 것을 인지한다.

이처럼 시나리오는 게임을 진행하는데 있어 필수적인 요소이다. 하지만 시나리오를 MMORPG에서 표현하는 일은 쉽지 않다. 프로그램이나 그래픽처럼 정량화되지 않으므로 퀄리티를 측정할 기준이 부족하다. 또 게임의 장르, 하드웨어의 기반에 따라 필요한 시나리오가 판이하게 다르기 때문에 제작의 어려움이 따른다. 실제 제작에 있어서도 별반 다르지 않다. 엔씨의 길드워나 블리자드의 월드오브워크래프트 등의 MMORPG 들이 연출까지 가능한 시나리오 Tool 을 보유하고 있는 것으로 알려져 있지만 여전히 다수의 게임은 텍스트를 기반으로 한 간단한 자체 프로그램으로 제작을 진행하고 있다. 이 경우 동영상이나 사운드 등의 연출이 필요한 이벤트를 표현할 수가 없고, 재사용이나 활용도 어렵다.

국산 게임의 단점으로 시나리오를 지적하는 경우가 많은 것도 이런 이유를 기반으로 할 것이다.[3] 결국 이러한 문제점을 해결하기 위해서는 퀘스트의 일정 퀄리티를 유지하면

서도, 재사용할 수 있는, 그러면서도 기본적인 이야기 구조를 유지할 수 있는 시나리오 저작도구의 연구가 필요하다.

본 연구에서는 MMORPG의 게임 시나리오가 직선적이지 않다는 것에서 출발했다. 게임은 사용자에게 어떤 목표를 주고 그것을 수행하면 보상을 하겠다고 이야기하며 사용자의 반응을 이끌어낸다. 이것을 퀘스트라 하는데 이 퀘스트와 퀘스트에 사용되는 스크립트를 DB화하는 것이 본 논문의 제안이다. 이것이 가능한 이유는 세계관이 다른 게임이라도 MMORPG라면 기본적으로 가지고 있는 퀘스트가 비슷하기 때문이다. 즉 대부분의 게임은 하나 이상의 NPC를 가지고 있으며 대부분의 NPC들은 처음 클릭과 동시에 '안녕하세요', '반갑습니다' 등의 대사를 하면서 인사를 하는 행동을 취한다. 퀘스트를 줄 때는 해야 할 일을 지시하고, 그것이 수행되는 경우 보상을 하고 해당되는 대사를 한다. MMORPG의 퀘스트 역시 기본적인 이야기의 구조인 기승전결을 지니고 있기 때문에 발생하는 진행방법이다.

그러므로 자주 사용되는 대사 및 행위를 DB에 미리 저장하고 이를 구성해 퀘스트를 만들면 반복적인 퀘스트 작업 시간을 크게 단축시킬 수 있을 뿐만 아니라, 게임 시나리오의 고질적인 문제인 기술 축적이 어렵다는 부분도 일부 해결할 수 있다. 또 개발을 통해 DB를 갱신하며 다양한 퀘스트를 누적시켜 궁극적으로는 정량적으로 풍부한 퀘스트 DB를 보유할 수 있다.

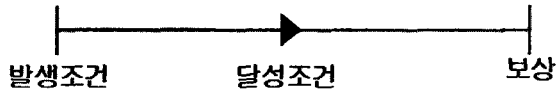
2장에서는 퀘스트의 구조에 대해 알아보고 3장에서는 MMORPG에서 퀘스트가 갖는 특징을 살펴본다. 4장에서는 퀘스트를 DB화 할 때 어떻게 구성 되어야 하는지, 후속작의 제작 시 재구성 방법을 제시한다.

2. 퀘스트 구조

게임 시나리오를 사용자에게 전달 위해 제작자는 여러 가지 방법을 사용한다. 그 중 퀘스트는 사용자의 행동을 유도한다는 면에서 게임의 특징인 양방향성에 가장 근접한 형태의 시나리오 표현방법이라고 할 수 있다.

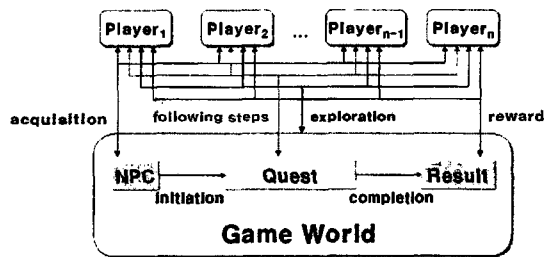
기본적인 퀘스트의 구조는 <그림 1>과 같이 3가지 요소로 구성된다. 퀘스트 발생조건, 퀘스트 달성 조건, 보상이다.

하나의 구조를 가진 퀘스트



〈그림 1〉 기본적인 퀘스트 구조

〈그림 2〉는 〈그림 1〉의 기본적인 퀘스트 구조를 MMORPG의 형태로 변형한 것이다. 퀘스트의 시작은 사용자가 어떤 행위를 수락하는 것이다. 이것을 게임 플레이에서는 '임무'라고 부른다. 임무를 받은 사용자는 게임을 탐색하면서 그 임무를 계속 진행한다. 그리고 완전히 임무를 완료하면 그 결과로 아이템이나 사이버머니 등의 보상을 받는다. 즉 MMORPG의 퀘스트 구조 역시 〈그림 1〉의 기본적인 퀘스트 구조와 같다.



〈그림 2〉 MMORPG의 퀘스트 모델 [3]

MMORPG의 경우 콘솔이나 패키지 게임과 달리 타 사용자라는 변수가 더 고려한다. MMORPG는 퀘스트의 진행 시 다른 사용자의 개입이 가능하다. 퀘스트 달성에 도움을 받을 수도 있지만, 반대로 방해받을 수도 있으며 혹은 협동해서 해결하는 경우도 생길 수 있다. 그래서 MMORPG의 경우 사용자와 사용자의 관계를 기준으로 '솔로잉 구조'와 '파티형 구조'로 나눌 수 있다[4].

솔로잉 구조는 사용자 홀로 진행하는 형태의 퀘스트이며, 누군가의 도움을 받을 수도 있고 못 받을 수도 있다. 물론 도움을 받는다고 해도 퀘스트 종료 후 받는 보상은 개인 보상이므로 소득을 나눌 수는 없다. 대부분 하나의 임무만을 지닌 퀘스트가 많아 퀘스트의 단위 시간이 짧다.

파티형은 여러 명의 사용자가 하나의 그룹을 형성해서 같이 임무를 해결하는 방법이다. 하나의 목표를 달성하는 과정에서 사용자는 협력을 할 수도 있고 다른 그룹과 경쟁을 할 수도 있다. 또 퀘스트의 보상은 참여한 모든 사람이 분배해 갖는다. 그룹을 통해 임무를 수행하므로 퀘스트에 대한 목표의식을 강하게 느낄 수 있으며, 게임의 임무보다 커뮤

니티 그 자체가 강화되기도 한다. 그래서 〈그림 2〉에서 Player n은 솔로잉 구조의 사용자로 볼 수도, 파티형 구조의 파티로 볼 수도 있다.

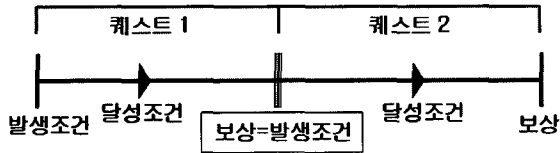
3. MMORPG의 퀘스트 특징

초기 MMORPG는 저장 공간의 부족과 서버 및 Network의 불안정을 이유로 퀘스트를 최대한 단순히 설계했다. 이 때문에 퀘스트를 위해 많은 저장 공간을 할당하거나 저작 도구를 만드는 경우는 드물었다. 퀘스트를 동작하도록 하는 기본 프로그램은 서버코드 부분에 하드코딩 되어 있는 경우가 많았기 때문에 기획자는 가변적인 퀘스트 구조보다는 하나의 구조를 가진 퀘스트를 만들어야 했다[5]. 기본적인 퀘스트 구조를 벗어날 수 없었기 때문에 몬스터를 더 많이 잡거나 더 많은 아이템을 가져오라는 등의 양적인 조건만 늘어난 퀘스트를 제공할 수밖에 없었다. 넥슨에서 98년에 서비스한 바람의 나라[6]도 초기 서비스에서는 퀘스트의 난이도를 아이템의 양으로 조정했다. 하지만 이런 방법은 플레이 시간의 연장 외에는 변화가 없는 형태이므로 사용자에게 지루함을 유발한다.

오리진에서 개발한 MMORPG 울티마 온라인[7]에서는 처음으로 사용자와 사용자 간의 인터랙션을 중심으로 한 퀘스트가 등장했다. 사용자들이 같이 퀘스트를 진행하는 파티형 퀘스트 구조가 등장했으며 NPC를 통한 퀘스트 전달도 울티마 온라인에서 정형화 되었다.

그 후 버턴트에서 개발한 퀘스트 중심형 MMORPG인 에버퀘스트가 등장하면서 퀘스트끼리 시나리오가 이어지는 연계 퀘스트가 등장한다[8]. 〈그림 3〉처럼 연계 퀘스트는 한 퀘스트의 내용이 끝나는 것이 아니라 그것의 달성이 또 다른 퀘스트의 조건이 되는 퀘스트이다. 연계 퀘스트는 하나의 구조를 가진 퀘스트에 비해 달성조건을 다양하게 만들 수 있다. 한 퀘스트가 다음 퀘스트의 전 단계 스토리가 되므로 시나리오의 전달에도 효과적이다.

위와 같이 MMORPG의 퀘스트는 단발성 퀘스트보다는 에버퀘스트와 같이 연계 퀘스트로 진행되는 경우가 많아지는 추세이다. 연계 퀘스트는 단발성 퀘스트에 비해 퀘스트의 진행 시간이 길고 퀘스트 간의 연계성이 강하다. 그래서 방대한 세계관과 시나리오를 갖는 게임에 더 적합하다.

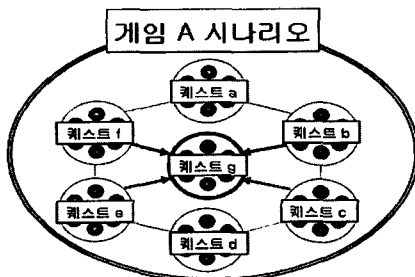


〈그림3〉 연계 퀘스트

4. 시나리오 도구 구성의 고려사항

3장까지의 논의를 종합적으로 살펴보면 게임의 특징을 고려한 시나리오 도구를 구성하기 위해서 몇가지 요소를 가져야 한다. 첫째, 서사구조를 지니면서도 사용자와 게임 간의 양방향성 요소를 줄이지 말아야 한다. 사용자가 선택을 통해 행동을 하는 것은 게임의 중요한 양방향성이다. 서사구조를 해치지 않는 범위에서 사용자가 자신의 행동을 결정할 수 있도록 제작되어야 한다. 둘째로 퀘스트의 세 가지 요소를 디자인 할 수 있어야 한다. 퀘스트 발생조건과 달성 조건 그리고 보상 조건을 가진 구조로 만들어져야 하며 이것들이 서사적으로 연결될 수 있어야 한다. 세 번째는 재사용이다. 만들어진 퀘스트를 다시 사용해 제작 시간을 줄이고 비용을 절감할 수 있어야 한다. 네 번째는 퀄리티 및 공정의 명확성이다. 퀘스트 제작도구를 사용하는 사람이 누구이든 제작물의 퀄리티와 소요 시간이 비슷해야 한다.

또 MMORPG의 시나리오의 특징을 고려해야 한다. MMORPG는 패키지나 휴대용게임처럼 기승전결의 이야기 구조를 이루지 않는다. 즉 시작은 있되 끝이 없는 시나리오가 MMORPG이다. MMORPG의 경우 결론이 없는 대신 캐릭터에게 다양한 임무를 부여하면서 게임을 지속할 수 있게 만든다. 이 퀘스트는 선형의 구조가 아니라 〈그림 4〉처럼 그물형태의 연결구조를 갖는다.



〈그림4〉 MMORPG의 시나리오의 예

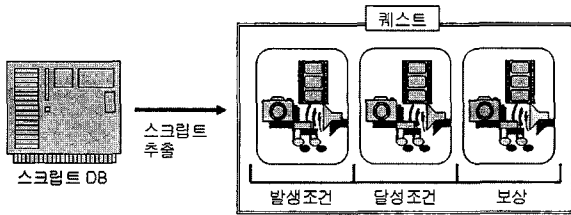
〈그림 4〉와 같이 7개의 퀘스트를 가지는 게임 A의 시나리오를 살펴보자. MMORPG의 퀘스트 특징은 서로 연계하고 있다는 것이다. 퀘스트 ④는 ⑤, ⑥와 연결된 연계 퀘스트이다. ④의 보상이 곧 ⑤, ⑥의 발생조건이 된다. 퀘스트 ④가 끝나면 사용자는 ⑤나 ⑥를 선택한다. ⑤, ⑥는 ④를 제외하면 두 가지의 결과로 진행할 수 있다. 즉 연계되어 있는 퀘스트의 숫자가 많을수록 발생 조건이 많은 퀘스트가 된다.

⑧의 경우 4개의 퀘스트와 연계되어 있는데, 발생조건이 다른 퀘스트보다 하나 더 있기 때문에 ⑧가 시나리오 A에서 가장 해결하기 어려운 퀘스트가 된다. ④에서 시작한 게임이라고 예를 들자. 그러면 ⑥와 ⑤, 그리고 ③과 ②의 퀘스트를 마쳐야 퀘스트 ⑧를 발생시킬 수 있다. 즉 퀘스트를 어떻게 연결하느냐 만으로 난이도를 조정할 수도 있다는 것이 MMORPG의 특징이다.

5. 스크립트의 분류와 DB 제작

퀘스트들은 각각의 이야기를 표현하기 위해 대사와 NPC의 동작, 동영상들을 가진다. 이것을 스크립트라 한다. 스크립트는 그 스크립트가 무엇을 하나에 따라 성립 조건, 달성 조건 그리고 보상 조건 스크립트로 나눌 수 있다. 퀘스트 역시 하나의 이야기 구조를 가진다. 앞서 언급한 든 '소를 잡으면 가족을 주겠다' 라는 퀘스트를 예로 들자. 이 경우 이 퀘스트는 우선 누군가에게 소를 잡으라는 임무를 부여 받을 것이다. 이것을 성립조건 스크립트라 한다. 그 후 캐릭터가 쉽게 소를 잡기 위해서 소가 어디에 위치하는지, 잡으려면 뭘 해야 하는지 알려주는 부분이 필요하다. 이것을 달성 조건 스크립트라 부른다. 그 후 소를 잡아 오면 가족을 주는데 이 부분을 보상 조건 스크립트라 한다. 이때 고려해야 할 부분은 스크립트라고 해서 대사만 나오는 것이 아니라 동영상이나 NPC의 움직임, 사운드 등도 같이 등장한다는 것이다. 실제 게임에서는 '소를 잡으면 가족을 주겠다.' 라는 간단한 퀘스트를 위해서 수백 줄의 텍스트형 스크립트와 동작, 영상 그리고 수십 개의 음원이 들어간다. 〈그림 6〉은 퀘스트의 시작인 성립조건 스크립트를 일부 제작해본 것이다.

형태에 따라 대화와 지문의 형태를 갖는 텍스트형, 컷 씬이나 동영상, NPC의 동작 등의 형태를 갖는 그래픽형, NPC의 음성이나 효과음 등의 형태를 갖는 사운드형을 갖는다.



〈그림5〉 스크립트에서 퀘스트로 구성되는 과정

해당 퀘스트에 맞춰서 스크립트를 제작하지만 미리 스크립트가 만들어 있다면 퀘스트에 맞춰서 구성할 수 있다. 〈그림 5〉은 미리 스크립트들이 DB화 되어 있다는 가정 하에 퀘스트를 구성하는 것을 도식화 했다. 만약 자주 사용되는 스크립트, 예를 들어 NPC가 인사를 하는 장면이나 ‘어서오세요’, ‘안녕히 가세요’ 등 빈도수가 많은 대화를 미리 DB화 했다면 조건에 맞춰서 추출, 배치 할 수 있다. 거꾸로 퀘스트에 쓸 적당한 스크립트가 없다면 스크립트 DB내에 추가해도 된다.

즉 스크립트 DB는 매 퀘스트마다 추가되고 갱신된다. NPC와 간단히 인사를 나누는 퀘스트이건, 배달을 하는 퀘스트이건 퀘스트의 발생, 달성 그리고 보상 단계는 최소한의 필수조건이므로 이렇게 세단계의 영역을 만들고 이미 작성된 DB에서 스크립트를 추출, 배치를 하면 퀘스트마다의 공동 스크립트는 한번만 제작하면 계속 사용할 수 있어 재사용이 가능하다.



대사 : 잘 왔네. 자네가 영웅이라는 얘기를 듣고 기다리고 있었지 이 편지를 빨리 A 마을에 전해주게. 늦으면 많은 사람들이 죽음을 면치 못할거야
지문 : 새로운 퀘스트가 추가되었습니다.



동영상 : 퀘스트 01 전용 동영상 출력
NPC 동작 01 : 고개를 끄덕이며 캐릭터를 쳐다 봄
NPC 동작 02 : 수확 시 기뻐하는 동작



BGM : 긴박감 01 BGM 출력
효과음 : 퀘스트 추가 효과음 출력

〈그림6〉 스크립트로 만든 퀘스트의 예

〈그림 6〉은 스크립트를 이용해 만든 퀘스트를 예로 든 것이다. 첫 부분은 대화와 지문의 텍스트이다. 두 번째 부분은 영상과 동작 관련 그래픽이다. 세 번째 부분은 사운드 관련 부분이다. 이 부분들은 실제 사용자의 눈에는 동시에 발생하는 것으로 보이게 되고, 하나의 퀘스트가 된다.

이렇게 만들어지는 퀘스트는 퀘스트 DB에 저장한다. 그 이유는 MMORPG가 대부분 비슷한 기본 구조의 퀘스트들을 갖기 때문이다. 예를 들어 새로운 대륙으로 이동하거나 새 물건을 구입하거나 몬스터 사냥을 의뢰받는 일은 MMORPG에서 흔히 볼 수 있는 퀘스트이다. 이런 퀘스트들 역시 단위별로 DB화 시키면 이후 제작 시 그대로 사용하고 몇몇 스크립트의 변화만으로 기본적인 퀘스트를 빠르게 제작할 수 있다.

6. DB를 이용한 게임 제작

이렇게 DB화하는 작업은 하나의 게임만을 제작할 때는 도려 일반 게임 제작보다 더 시간이 든다. 그 이유는 스크립트와 퀘스트들을 조건에 맞게 분류하고 DB화 해야 하기 때문이다. 하지만 두 번째 제작부터는 비슷한 유형의 퀘스트를 스크립트와 함께 끌어다 사용하고 스크립트의 몇몇 부분을 수정해서 사용하는 방식으로 만들 수 있어 제작기간을 크게 단축시킬 수 있다. 또 정량적으로 처리하기 힘든 게임 시나리오를 데이터로 DB화 시키는 것이라 반복적으로 게임을 제작하며 재구성할수록 더 풍부한 스크립트와 퀘스트가 있는 DB를 얻을 수 있다.

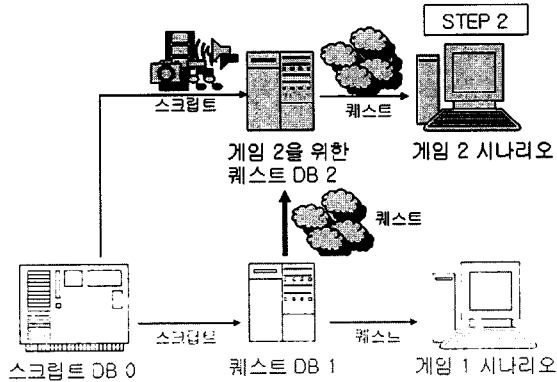


〈그림7〉 첫 번째 게임의 시나리오 제작과정

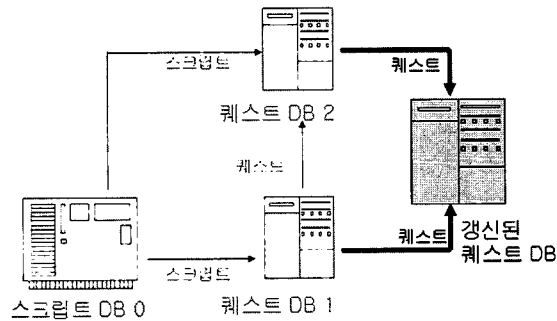
이것은 세계관이 비슷하거나 속편을 제작하는 경우에 더 활용도가 높다. 즉 비슷한 배경 스토리를 지니거나 퀘스트의 이야기 진행 구조가 같다면 더 많이 차용해 올 수 있기 때문이다.

우선, 첫 번째 게임을 만들 때는 스크립트 DB에서 퀘스트를 제작해 DB를 거쳐 게임 시나리오를 만든다. 〈그림 7〉에서 나와 있듯 1번 퀘스트 DB는 게임 1을 위해서 만든 전용 DB가 된다. 두 번째 게임을 제작할 때는 앞서 제작된 퀘스트 DB를 활용해서 또 하나의 퀘스트 DB를 제작한다. 〈그림 8〉이 두 번째 게임 시나리오의 제작 과정이다. 이 때 새

로운 퀘스트를 제작하는 경우라면 스크립트를 구성하고 추가된 스크립트는 DB화시킨다. 이 때 원하는 스크립트가 없으면 추가하므로 해당 스크립트 DB는 처음 게임을 만들 때보다 커진다.



(그림 8) 두 번째 게임의 시나리오 제작과정



(그림 9) 제작 후 갱신 퀘스트 DB 작성

〈그림 9〉처럼 두 개의 게임을 제작한 후 퀘스트 DB가 두 개가 되면 그것들을 합쳐 새로운 재구성 퀘스트 DB를 제작한다. 이미 퀘스트 DB 2는 퀘스트 DB 1의 퀘스트를 상당량 사용했으므로 사용하지 않은 부분과 새로 추가한 부분들을 합쳐서 하나로 만드는 것이다. 이를 퀘스트의 갱신이라고 부르며 게임 제작이 끝나면 전작과 후작의 퀘스트가 모두 합쳐진 갱신 퀘스트 DB가 나오게 된다. 재구성된 퀘스트 DB는 게임 1과 2의 장르 모두를 제작할 수 있는 DB가 된다. 이와 같은 방식으로 게임n을 제작할 때도 마찬가지로 기존의 퀘스트 DB를 갱신하면서 진행한다. 게임의 제작이 많아질수록 다양한 패턴의 퀘스트들이 서버에 체계화된다. 이것은 게임을 만들수록 퀘스트와 스크립트가 양적으로 증가한다는 장점이 있고, 첫 번째 게임을 제작할 때를 제외하고는 기존 방식보다 퀘스트 제작 기간이 짧다. 그리고 다양

한 MMORPG의 퀘스트가 구성된다면 여러 가지 세계관을 가진 MMORPG의 퀘스트 DB를 만들 수 있을 것이다.

7. 결론

본 논문에서는 MMORPG의 시나리오 구조가 기존 패키지와 달리 직선형 기승전결이 아니라 연계 구조라고 분석하고 거기에 맞춰 퀘스트를 제작할 수 있는 스크립트 DB와 퀘스트 DB의 설계를 제안한다. 연계 구조를 이루기 위해서 시나리오가 퀘스트라는 덩어리로 이야기가 설계되어야 하는데, 이 퀘스트는 대화와 그래픽 소스, 사운드 등을 가지고 있는 스크립트로 이루어져 있다. 임무를 주고 보상을 받는 퀘스트 형태에 맞춰서 스크립트를 미리 작성하고 그것을 DB화하면 다음번 게임 제작 시 활용할 수 있다는 것이 본 논문의 제안이다.

이렇게 구성하는 경우 두 번째 게임을 만드는 경우 간단한 퀘스트들은 대부분 추출해 사용할 수 있으며, 때 제작마다 여러 가지 형태의 퀘스트들을 정량적으로 늘릴 수 있다는 장점이 있다. 또 제작자에 따라 시나리오의 질이 일정하지 않던 단점도 극복할 수 있다.

4, 5장에서는 이를 이용해 실제로 게임을 만들 때의 예를 들었다. 첫 번째 게임을 만들 때는 기존 DB 방식과 같다. 하지만 두 번째 게임을 만들 때 퀘스트 DB 1을 참고해서 제작하므로 제작기간이 단축된다. 또 변경된 스크립트로 스크립트 DB를 재구성하므로 스크립트 DB는 정량적으로 양이 늘어난다. 두 개의 게임을 제작 한 후 두 게임의 퀘스트를 모아 갱신된 퀘스트 DB를 만든다.

하지만 퀘스트 DB를 이용한 게임 시나리오 제작은 몇 가지 문제점을 가지고 있다. 첫 번째 게임을 제작할 때 제작기간이 상당히 길다는 것이다. 스크립트 DB의 재구성을 수시로 진행하면서 퀘스트를 만들어야 하므로 첫 번째 게임을 제작할 때 제작 기간이 늘어난다. 또 배경이 상이한 게임을 후속작으로 제작하는 경우 활용도가 떨어진다는 단점도 있다. 현재 논문에서는 퀘스트의 구조인 임무를 받고 수행하는 진행에 초점을 두어 세 가지 분류를 제안했다. 하지만 성향 상 분류가 안 되는 상황이 발생하기도 했다.

그러나 기본적으로 퀘스트와 스크립트를 분류하고 DB화

