

## 배회 에이전트에 의한 동적 레벨 디자인

김종호<sup>0</sup>, 조형제, 한성호  
(동국대학교 영상대학원), 광운대학교 교육대학원  
j05y@dgu.edu<sup>0</sup>, chojh@dgu.edu, digieye@kw.ac.kr

### Dynamic Level Design with Wandering Agent

Jong-Ho Kim<sup>0</sup>, Hyung-Je Cho, Sung-Ho Han  
(Dongguk University Graduate School of Digital Images and Contents),  
Kwangwoon University Graduate School of Education

#### 요약

현재 게임 레벨 디자인은 게임 유저로 하여금 게임에서 일정수준의 재미를 느낄 수 있게 게임을 구상하고 적용하는 것으로 게임개발과정에서 중요한 역할을 해왔다.

이러한 레벨 디자인은 레벨 디자이너에 의해 설계되어 왔으며, 대부분의 게임 개발팀에서는 한 레벨 디자이너가 한 레벨씩 맡아서 설계하고 있다. 따라서 게임이 복잡해질수록 레벨 디자이너의 숫자는 늘어나고, 다양한 레벨을 만들기 위해 들어가는 인력과 작업시간은 늘어나야 한다.

본 논문에서는 레벨 구조에 배회 에이전트를 도입하여 하나의 레벨만으로도 다양한 레벨들의 효과를 경험해 볼 수 있도록 레벨 구조가 동적으로 변환되는 방법을 제안한다. 또한 배회 에이전트에 의해서 사용자들 간의 밸런스 조절도 가능함을 보이고자 한다.

#### Abstract

Until now, the level design for games has played a very important role in the process of developing games since it has been mapped out and applied so the game players have fun to certain level in playing games.

So as game becomes more complex the number of game designers also grow and to make diverse levels, work force and working hours also should grow larger.

In this treatise, it suggests a way where level's structure is changing dynamically so with only one level you can experience diverse levels by introducing wandering agent to the level structure. Also, I am trying to show the possibility of balance between the players by wandering agent.

Keyword : wandering agent, level design

### 1. 서론

레벨 디자인(Level Design)이란 롤플레이팅 게임(Role Playing Game)에서 흔히 말하는 등급의 레벨이 아니고 게임 디자인에서 나온 게임 시스템 설계를 기본으로 하여 제

작한 그래픽의 작업물을 이용하여, 게임 플레이어의 반응에 따라 다른 다양한 구조를 가진 게임의 환경을 구축하는 과정을 통틀어 말한다. 레벨 디자인의 결과에 따라 게임의 재미와 완성도가 좌우되는 만큼 레벨 디자인은 게임을 제작하는 핵심 디자인 중의 하나이다. 레벨 디자인에서 가장 중

요한 것은 게임 유저가 게임에서 일정수준의 재미를 느낄 수 있게 게임을 구성하고 적용하는 것으로 게임 진행의 흐름에 연속성과 조화성을 살리면서 게임의 난이도 조절을 잘하는 것이 중요하다[1, 2, 3].

이러한 레벨 디자인은 레벨 디자이너에 의해 설계되어왔으며, 대부분의 게임 개발팀은 한 레벨 디자이너가 한 레벨씩 맡아서 설계하고 있다. 따라서 게임이 복잡해질수록 게임 제작에 필요한 레벨 디자이너의 숫자는 늘어난다[3].

그러나 게임에 도입되는 다양한 레벨들은 디자이너의 많은 시행착오와 작업시간을 필요로 하는 문제점을 가지고 있다.

본 연구에서는 디자이너에 의해 정적으로만 설계되었던 레벨에 배회 에이전트(Wandering Agent)를 도입하여 한 레벨만을 가지고도 다양한 레벨들을 경험해 볼 수 있도록 레벨 구조에 동적인 변환을 주고자 한다. 이는 지금껏 다양한 레벨들을 만들기 위해 드는 인력과 작업시간을 보완할 수 있다.

본 연구는 다음과 같은 단계를 통하여 전개된다.

1단계 : 레벨 구조의 정의 및 설계

2단계 : 레벨변환을 위한 배회 에이전트의 행동 규칙 설계

3단계 : 밸런스(Balance) 조절을 위한 배회 에이전트의 행동 규칙 설계

이들 과정에 필요한 모든 레벨의 그래픽 작업물은 3D 모델링 툴인 3DS Max를 이용하여 모델링한다.

본 논문의 구성은 다음과 같다. 2장에서 배회 에이전트의 행동규칙을 설계하기 위해 관련된 연구를 살펴보고, 3장에서는 레벨 구조를 설계하고 2장의 연구를 바탕으로 조건에 따라 배회 에이전트의 행동 규칙을 설계하여 동적인 레벨 변환과정을 소개한다. 끝으로 4장에서는 본 연구를 통해 얻은 연구 결과를 종합하고 앞으로 필요한 연구들을 제안한다.

## 2. 관련 연구

배회 에이전트가 공간을 자유롭게 이동하다 레벨을 만났을 때, 특정 행동규칙에 의하여 구조물에 대한 문제해결이 이루어짐과 동시에, 일련의 행동으로 인하여 동적 레벨을 설계하기 위해 쓰이는 기법들에 대해서 간략히 소개한다.

### 2.1 규칙기반 시스템(Rule-based system ; RBS)

RBS는 일련의 “만약...그러하면”(if...then)문장으로 이뤄져 있다. 이런 형태는 전문적인 지식이나 경험을 통해 얻은 규칙 등을 압축하여 표현하기에 적합하다. 모든 규칙을 규칙 메모리(rule memory)에 보관하고 현재 상태를 작업 메모리(working memory)에 유지하여 작업 메모리(working memory)와 매칭(matching)되는 규칙(rule)을 찾아내어 이를 실제로 적용한다. 이 기법은 배회 에이전트의 행동을 제어 하기에는 적합하나, 지형을 동적인 구조로 바꾸기에는 적합하지 못하다.

### 2.2 트리거 시스템(Trigger system)

트리거 시스템은 단순한 만약/그러하면(if/then) 규칙들을 게임 객체나 세계 자체에 내장시킬 수 있도록 하는 고도의 특화된 스크립팅 시스템이다. 개념이 매우 간단하고 안정적이기 때문에 레벨 디자이너에게도 유용한 수단이다. 트리거 시스템은 레벨 디자인 도구나 스크립팅 언어를 통해서 제공되곤 한다. 게임에서 적용되는 예를 간단히 살펴보자[5].

디자이너는 맵 한 가운데에 바닥 트리거 하나를 놓은 다음, 플레이어가 그 바닥 트리거를 밟으면(조건) 닫혀있던 문이 열리도록(반응) 설정한다.

이런 예에서 보듯이, 트리거 시스템은 복잡한 스크립팅 언어 없이도 디자이너가 특정 조건에 따라 사건들을 지정할 수 있게 한다.

### 2.3 트리거 시스템의 플래그 및 카운터와 결합

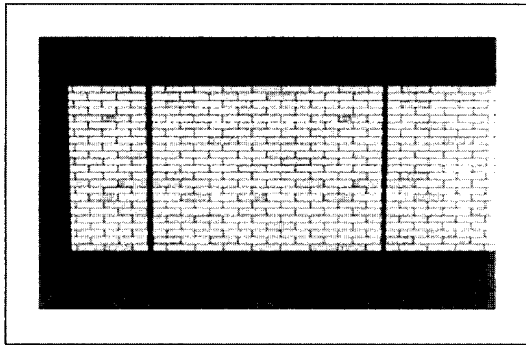
여러 트리거들이 함께 결합될 수 있으려면, 시스템 안의 모든 트리거들이 접근할 수 있으며 즉시 설정할 수 있는 상태들이 존재해야 한다. 따라서 트리거 시스템은 발동된 트리거들의 상태를 추적할 수 있는 일련의 플래그들과 카운터들을 갖출 필요가 있다. 이러한 플래그들과 카운터들이 있으면 트리거 시스템은 특정 이벤트에 표시를 하거나 발동 횟수를 셀 수 있으며, 이를 통해서, 본 연구의 예를 들면 세 개의 타일들을 특정한 순서로 밟았을 때, 트리거가 발동되도록 할 수 있다. 이 기법은 본 연구에서 사용자(Player)들 간의 밸런스(Balance) 조절을 하기에 적합하다.

### 3. 배회 에이전트의 행동규칙 설계

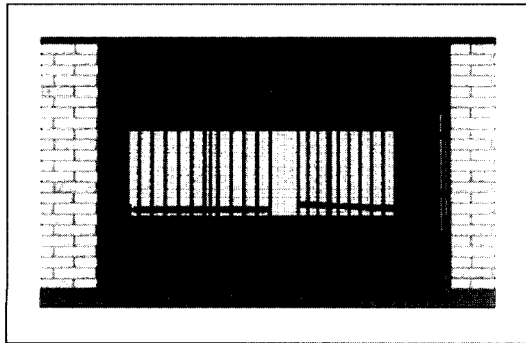
본 장에서는 배회 에이전트가 배회를 하다가 만날 수 있는 레벨 구조를 정의하고, 이를 바탕으로 레벨 변환을 위한 배회 에이전트 행동규칙과 사용자들 간의 밸런스 조절을 위한 배회 에이전트 행동규칙을 설계한다.

#### 3.1 레벨 구조의 정의

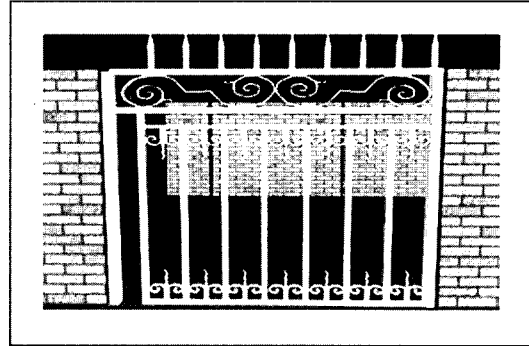
이동에 물리적으로 영향을 미칠 수 있는 레벨을 정의한다. 여기에는 바닥, 벽, 문, 나무 등이 포함된다. 이외에도 게임 세계에 존재하는 대부분의 정적 개체는 레벨에 포함된다[7]. 본 연구에서는 단순히 바닥만을 정적 개체로 사용하며, 벽과 열린 문(Open door), 닫힌 문(Close door)은 동적인 레벨로 사용한다. 벽은 동적 레벨이지만 배회 에이전트에 의해 변환되지 않는 레벨에 해당되며, 열린 문과 닫힌 문은 변환되는 레벨에 해당된다.



[그림 1] 벽



[그림 2] Open door



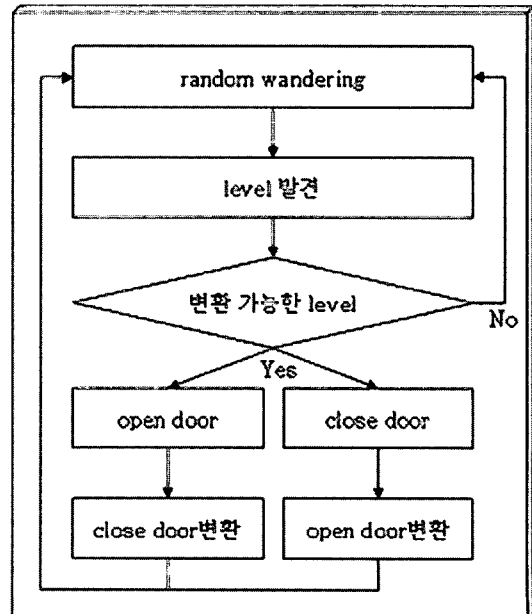
[그림 3] Close door

[그림 1, 2, 3]은 동적인 레벨로 설계하기 위해 3D 그래픽 툴인 3D Studio Max를 이용하여 모델링한 것이다.

#### 3.2 레벨 변환을 위한 배회 에이전트 행동 규칙 설계

앞 절의 레벨 구조의 정의를 바탕으로 배회 에이전트 행동 규칙을 설계한다.

여기서는 사용자들은 반드시 열린 문으로만 이동 가능하고 열린 문과 닫힌 문을 임의로 변환 시키지 못하며, 배회 에이전트에 의해서만 열린 문과 닫힌 문은 변환 시킬 수 있다고 가정한다.



[그림 4] 배회 에이전트의 행동 규칙

배회 에이전트는 지형을 랜덤하게 배회를 하면서 레벨을 만났을 경우 [그림 4와 같은 규칙을 수행한다. 변환을 시킬 수 있는 것인가? 그렇지 않는 것인가? 변환을 시킬 수 있다고 판단이 되었을 경우 열린 문의 경우 열린 문으로 지나간 후 닫힌 문으로 변환 시키고, 닫힌 문의 경우 그 자리에서 열린 문으로 변환 시키고 열린 문으로 지나간다. 변환규칙을 수행한 후 배회 에이전트는 처음 상태로 초기화 된다.

배회 에이전트의 행동 규칙 알고리즘은 [알고리즘1과 같다.

```

if (랜덤 배회 = TRUE) THEN(
    if (레벨 발견 = TRUE) THEN(
        if(변환 시킬 수 있는가? = TRUE) THEN(
            if(Open door? = TRUE)
                Open door로 지나간 후 Close door로 변환
            else(Close door? = TRUE)
                Open door로 변환한 후 Open door로 지나감
        )
    )
)
    
```

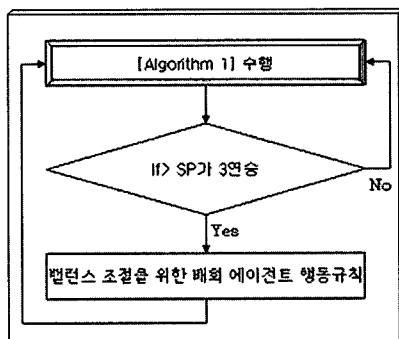
[알고리즘 1] 배회 에이전트의 행동 규칙

3.3 밸런스 조절을 위한 배회 에이전트 행동 규칙 설계

본 절에서는 3.2절의 배회 에이전트의 랜덤한 레벨 변환 행동규칙을 확장하여 사용자들 간의 밸런스 조절을 위한 레벨 변환 행동 규칙을 설계한다.

3.3.1 조건에 따른 배회 에이전트 행동 규칙 구조

대표적인 경우로서 대전게임을 예를 들어 그 개념을 소개한다. 초기에 사용자들 간의 동일한 조건의 능력치를 부여하고, 총 10회에 걸쳐서 게임이 진행되는 동안 한 플레이어가 3연승을 이룰 경우 그 플레이어를 SP(강한 플레이어)라 하고, 3연패 당한 플레이어를 WP(약한 플레이어)라 본 논문에서는 명한다.



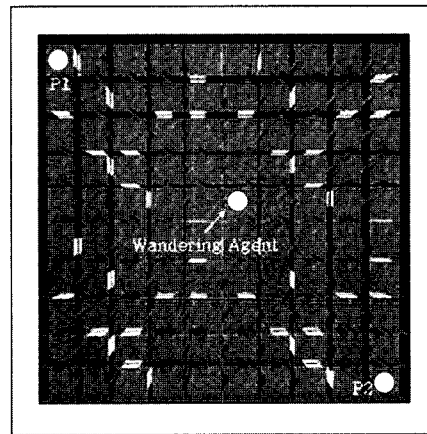
[그림 5] 조건에 따른 배회 에이전트 규칙

배회 에이전트는 [그림 4에서 보듯이 랜덤하게 배회를 하면서 [알고리즘1을 수행하다가 SP가 3연승을 이룰 경우 [그림 5와 같이 밸런스 조절을 위한 배회 에이전트 행동규칙을 수행한다.

3.3.2 밸런스 조절을 위한 배회 에이전트 행동 규칙

SP가 WP에게 연속해서 3번의 승리를 이룰 경우 배회 에이전트는 더 이상 행동 규칙을 수행하지 않고 WP에게 도움을 주는 레벨 디자인을 한다.

여기서는 [그림 6과 같이 배회 에이전트에게 가로10 \* 세로10의 이산적인 구조를 갖도록 제안한다.



[그림 6] 전체 레벨 구조

플레이어1은 P1 이라 명하고, 플레이어2는 P2라 명했으며 P1과 P2의 거리 차는 격자모양의 블록에서 떨어져 있는 블록의 수로 보았다.

```

if( SP가 3연승 = TRUE ) THEN
{
    if( P1 과 P2의 거리차가 6이상 = TRUE )
        배회에이전트는 [알고리즘1]을 수행
    else( P1 과 P2의 거리차가 5이하 = TRUE ){
        배회 에이전트는 두 플레이어 사이로 이동시작
        if( P1 과 P2 의 대전 중 SP에게 WP가
            쫓기는 경우 = TRUE ){
            WP가 지나간 후 배회에이전트는
            Open door -> Close door로 변환
        }
    }
}
    
```

[알고리즘 2] 밸런스 조절을 위한 배회 에이전트 행동 규칙

두 플레이어의 거리가 6이상일 경우 배회 에이전트는 [알고리즘1]을 따르다가 두 플레이어의 거리가 5이하일 경우 배회 에이전트는 두 플레이어 사이로 이동하기 시작한다. P1 과 P2의 대전 중 SP에게 WP가 쫓기는 경우 WP가 지나간 후 SP가 쫓지 못하게 열린 문을 닫힌 문으로 변환 시켜 WP가 회복할 수 있는 시간을 주어 P1 과 P2의 밸런스를 조절한다. 자세한 내용은 [알고리즘2]를 참고하면 된다. 특히 배회에이전트가 P1 과 P2의 거리차가 5이하일 경우 두 플레이어 사이로 이동한다고 되어 있는데 두 플레이어 간의 거리보다 배회 에이전트가 도달하는 거리가 먼 경우에 배회 에이전트의 역할은 무위로 될 수 있기 때문에 거리가 먼 경우에는 속도를 올려서 두 플레이어 사이로 이동할 수 있게 설계하였다.

#### 4. 결론 및 향후 연구 방향

본 논문에서는 배회 에이전트(Wandering Agent)에 레벨 변환 규칙을 설계하여 사용자에게 하나의 레벨만으로도 다양한 레벨들을 경험해 볼 수 있도록 동적인 레벨 변환 과정을 제안하였다.

게임제작 과정의 핵심 요소 중의 하나인 레벨 디자인은 레벨 디자이너에 의해 지금껏 설계되어왔으며, 게임이 복잡해질수록 게임제작에 필요한 레벨 디자이너의 숫자는 늘어나야만 했다. 게임 플레이어가 한 레벨을 클리어 했을 때 다른 레벨들이 미리 만들어져 있어야 한다. 이런 다양한 레벨들은 디자이너의 많은 시행착오와 작업시간을 필요로 하는 문제점을 가지고 있다.

본 연구에서는 이를 해결하기 위하여 하나의 레벨에 레벨 구조를 변환 시킬 수 있는 레벨 변환을 위한 배회 에이전트의 규칙을 설계하여 게임이 진행되는 동안 계속해서 레벨의 구조가 변환 되는 과정을 설계하여 보았다.

이와 같은 방법으로 배회 에이전트의 또 다른 규칙을 설계하여 배회 에이전트에 의해 두 플레이어 간의 밸런스 조절까지도 설계하여 적용해 보았다.

본 연구를 통하여 디자이너에 의해서 정적으로만 설계되었던 레벨의 구조를 게임이 진행되는 동안 동적으로도 변환 가능한 것을 관찰할 수 있었다. 여기서 배회 에이전트를 두어 레벨이 동적으로 변환되는 과정을 살펴보았지만, 기상의

영향이 라던가 날씨의 영향 등 기타 영향으로도 레벨의 동적 변환은 좋은 연구 과제가 될 것이다.

본 연구는 게임 설계자 입장에서는 충분한 연구가치가 있을 것으로 판단된다.

#### 참고문헌

- [1] 게임기획디자인연구회, 게임기획전문가, 박문각, 2004
- [2] Francois Dominic Laramee, 게임 개발 프로젝트를 성공으로 이끄는 게임기획&디자인, 정보문화사, 2003
- [3] Richard Rouse, 유명한 게임 디자이너들로부터 직접 게임 디자인을 배운다! 게임디자인 이론과실제, 정보문화사, 2001
- [4] 이재영,이형구,이용희, 실용적 예제로 배우는 게임 인공지능 프로그래밍, 사이텍미디어, 2006
- [5] 가이 W. 렉키 톰슨, 레벨 디자인, 지형, 음악 그리고 사운드, 정보문화사, 2003
- [6] Francois Dominic Laramee, 게임 개발 프로젝트를 성공으로 이끄는 게임기획&디자인, 정보문화사, 2003
- [7] Austin Grossman, 인공지능 게임 프로그래밍 실전가이드(최신 AI 기법을 적용한), 에이콘, 2005
- [8] 단테 트레글리아, Game Programming Gems 3, 정보문화사, 2003
- [9] 권기덕, 김인철, 동적 환경을 위한 정황-민감형 에이전트 구조, 한국정보과학회:학술대회지, 한국정보과학회 2004 가을 학술발표논문집(1), pp.118-120 , 2004
- [10] 김현, 이승관, 정태충, 방향 벡터를 이용한 다중에이전트 휴리스틱, 한국정보처리학회:학술대회지, 제22회 추계학술발표논문집(상), pp.525-528 , 2004
- [11] 이승관, 재미 모델 성능에서 다중 에이전트 상호작용 전략의 효과, 한국콘텐츠학회논문지, 1598-4877, 제5권3호, pp.193-199 , 2005
- [12] 임원택, 김구수, 선승상, 엄영익, MANET 환경에서의 이동 에이전트를 이용한 동적 클러스터링 기법, 한국정보과학회:학술대회지, 한국정보과학회 05 추계 학술발표논문집(1), pp.313-315 , 2005
- [13] Richard Rouse III, Game Design: Theory and Practice, Wordware Publishing, 2001

- [14] Andrew Rollings, Ernest Adams, Andrew Rollings and Ernest Adams on Game Design, New Riders Games, 2003
- [15] Tracy Fullerton, Christopher Swain, Steven Hoffman, Game Design Workshop: Designing, Prototyping, and Playtesting Games, CMP Books, 2004
- [16] Tom Meigs, Ultimate Game Design: Building Game Worlds, McGraw-Hill Osborne Media, 2003



김종호

2003년 동양대학교 멀티미디어공학과(학사)  
 2006년 동국대학교 영상대학원 멀티미디어학과(석사)  
 2006년 ~ 현재 디지털코리아(주) 연구원  
 관심분야 : 컴퓨터비전, 컴퓨터 그래픽스, 게임제작, 애니메이션 및 영상물제작



조형제

1973년 부산대학교 전자공학과(학사)  
 1975년 한국과학기술원 전기·전자공학과(공학석사)  
 1986년 한국과학기술원 전기·전자공학과(공학박사)  
 1986년 ~ 현재 동국대학교 멀티미디어학과 교수  
 관심분야 : 컴퓨터 그래픽스, 게임, 컴퓨터비전, 멀티미디어 정보처리



한성호

1997년 단국대학교 경영학과 졸업(학사)  
 2002년 동국대학교 연극영화학과 졸업(석사)  
 2006년 동국대학교 영상대학원 멀티미디어학과 박사수료  
 1997 ~ 1999 KONET Lead Designer  
 1999 ~ 2002 디지아이 프로덕션 대표  
 2001 ~ 2004 광운대학교 정보과학교육원 교수  
 2005 ~ 현재 광운대학교 교육대학원 초빙교수  
 2005 ~ 현재 세일즈스토리(주) 가상현실연구소 소장  
 관심분야 : 게임그래픽, 가상현실, 실험 애니메이션, 모션캡처 등

논문투고일 - 2005년 8월 13일

심사완료일 - 2005년 9월 15일