

게임 플레이 행동 패턴을 이용한 게임 환경 동적 생성 기법에 관한 연구

장새롬*, 이재열*, 최영미*, 주문원*, 윤태복**

*성결대학교 멀티미디어학과

**성균관대학교 컴퓨터공학과

tbyoon@skku.edu

Research of Dynamic Creation Method of Game Environment using
Game Player's Behavior Pattern

Sae-Rom Jang*, Jae-Yeol Lee*,

Young mee Choi*, Moon Won Choo*, Tae Bok Yoon**

*Dept. of Multimedia Engineering, Sungkyul University

**Dept. of Computer Engineering, Sungkyunkwan University

요 약

최근, 게임의 재미요소를 증대 시키고, 게임 생명주기(Life-Cycle)를 늘어나게 하기 위해 다양한 방법이 연구 중이다. 본 논문에서는 사용자의 게임 행위를 수집/분석 하여 동적으로 변화하는 게임 환경을 제공하는 방법을 제안한다. 플레이어의 게임에서 이동한 정보와 게임 결과 정보는 동적인 게임 지형 생성을 결정하는 기반 정보로 활용된다. 제안하는 방법의 실험을 위해 XNA기반의 게임을 만들었고 게임 플레이어의 패턴에 따라 동적으로 생성되는 게임 지형을 확인하였다.

ABSTRACT

Recently, various ways are being explored for enhancing the fun of computer games and lengthening the life cycle of them. This paper proposes the more active game environment by reflecting the decision that is made by collecting and analyzing user's behavioral data. The game results and traveling information of user is using for dynamic map creation. For experiment of the proposed method, we created games using the XNA Framework and confirmed the game environment which changing properly according to the user's game play.

Keyword : Intransitive relationship, Influence map, Strategy movement

접수일자 : 2009년 07월 29일

심사완료 : 2009년 08월 18일

※ 이 논문은 2009년도 정부(문화체육관광부)의 재원으로 한국콘텐츠진흥원의 지원을 받아 수행된 차세대 게임 전문 교육기관 지원 사업의 연구 결과입니다. 연구비 지원에 감사드립니다.

1. 서 론

IT기술의 발전과 함께 게임은 보다 현실적이고 지능적인 환경을 플레이어에게 제공하기 위하여 다양한 인공지능 기법을 이용하고 있다. 일반적으로 게임에 사용되는 인공지능 기법은 몰입감과 생명주기(Life-Cycle)를 늘리기 위한 연구가 대부분을 차지한다. 이러한 연구들은 정적인 게임 환경 속에서 NPC(Non Player Character)를 좀 더 지능적으로 표현하고 제어하기 위해 사용되어 왔다. 하지만 이러한 방식은 게임이 반복될수록, NPC의 속성들의 변화로, 게임의 난이도만 상승하게 되고, 사용자는 같은 환경에서 좀 더 어려운 NPC를 상대해야하는 상황이 발생한다. 이러한 기법들은 오히려 게임의 재미효과를 감소시키는 요인들로 작용할 수 있다. 즉, 플레이어와 상대적인 역할을 하는 NPC를 바꾸기 보다는, 게임의 환경적 요소(지형, 건물 등)를 변화시킴으로서 재미효과의 감소를 줄이고 게임의 난이도를 조절하는 방법이 필요하다.

본 논문에서는 플레이어의 게임 행위 데이터를 이용하여 게임 환경을 동적으로 변화시키는 방법을 제안한다. 제안하는 방법은, 먼저 게임을 플레이하면서 발생한 플레이어의 이동 정보를 수집한다. 플레이 시간동안 게임 지형에서 움직인 정보를 분석하여, 동적인 지형 변화를 위한 기반 정보로 활용한다. 특정한 위치에서 오랫동안 머물면서 게임을 플레이어에게 유리하도록 운영했다면, 다음 Stage의 지형 생성에 객체들의 생성과 배치를 변화시켜 사용자에게 좀 더 다양한 형태의 환경을 제공해준다.

실험에서는 Top-view 기반의 간단한 게임을 제작하고 제안하는 방법을 사용한 경우와 그렇지 않은 경우에 대하여 비교하여 유효함을 확인하였다.

2. 관련연구

게임에서 주로 사용되는 캐릭터 인공지능은 게임 속 NPC들을 좀 더 사실적이며 플레이어와 유

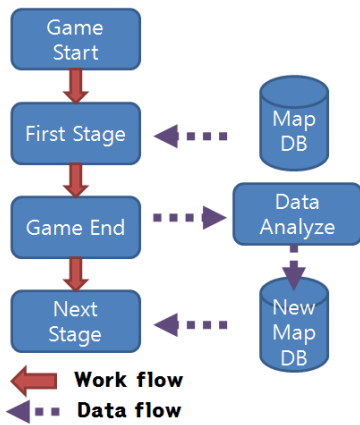
사하게 행동하게 하는 것이 중요한 목표였다[1]. 그러나 이러한 방식들은 정적인 게임 환경을 제공하며 NPC의 고정된 형태의 수치를 변화시켜주는 것으로, 게임이 후반부로 진행되어 갈수록 좀 더 강한 NPC 혹은 빠른 NPC 등 정형화된 패턴의 NPC들이 나타나게 된다. 이러한 변화는 사용자의 플레이 시간이 늘어날수록 난이도가 높아져서 게임의 몰입도가 낮아진다는 문제점이 있다. 이러한 문제점을 해결하기 위해 연구되었던 방법인 플레이어의 패턴을 학습하는 형태의 NPC 생성 역시 다양한 형태의 객체를 만들 수 있었으나[3,4] 사용자의 행동 패턴 이상의 형태는 나올 수 없다는 문제점이 있었다. 이러한 문제점을 보완하고자 [5] 게임 환경 변화 인공지능을 주목하게 되었고 사용자가 게임을 즐기는 과정에서 나오는 데이터로 게임 환경이 변화하여 좀 더 다양한 형태의 플레이를 즐길 수 있도록 하였다[6,7]. 이러한 게임 환경의 변화는 기존에 캐릭터 인공지능만 가지고 변화하는 게임이 아닌 게임 자체의 환경을 사용자의 행동 패턴에 맞춰 스스로 변화할 수 있도록 하여 사용자가 게임을 플레이하면서 느끼는 재미와 몰입도를 증가시켜 사용자의 플레이 타임을 연장하는데 있다고 볼 수 있겠다[8].

3. 사용자에게 적응된 동적인 게임 환경

본 논문에서는 개별 사용자의 게임 플레이시 행동을 분석하여 사용자에게 맞는 게임 환경을 조성하여 제공하는 것을 그 목적으로 한다. 이를 위해 사용자의 행동 패턴에 대한 데이터를 수집하고 수집한 데이터를 분석하여 지형을 변화시킨다. [그림 1]은 사용자에게 적응된 동적인 게임 환경을 제공하기 위한 전체 작업 흐름도이다. 첫 Stage에서는 지정된 지형에서 게임을 시작하지만, 다음 Stage에서부터 플레이어의 행동 패턴에 따라 변화된 지형에서 게임을 진행하게 된다.

3.1 게임 플레이어의 위치 데이터

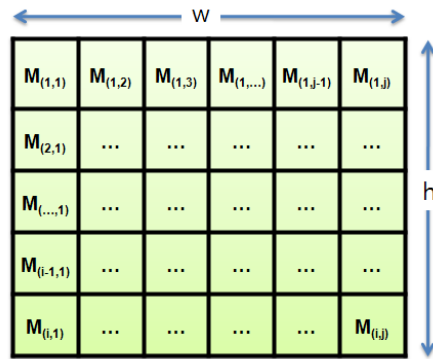
사용자가 게임을 즐기면서 미션을 완수 하기 위해 움직인 좌표 데이터와 NPC를 제거하기 위해 움직인 행동반경을 데이터화 하여 전체 Map에 객체들의 위치와 형태를 변화시켜 같은 게임을 즐겨도 게임을 하는 매회 다른 환경을 접할 수 있도록 하였다.



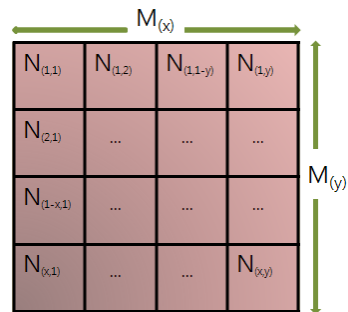
[그림 1] 제안하는 방법의 작업 흐름도

[그림 2]의 구조와 같이 Size가 $w \cdot h$ 크기의 Map을 $i \cdot j$ 개의 구역으로 나뉘, 각각의 영역에서 플레이어가 게임을 클리어 하기 위해 움직인 데이터를 수치화 한다. 정확한 수치화를 위해 [그림 3]과 같이 하나의 M영역을 $x \cdot y$ 크기의 셀로 다시 구성 한다. 각 M에서의 수치를 토대로 다음 Stage에서 새로운 환경의 Map을 조성한다. 만약 플레이어가 $M_{(1,1)}$ 지역에서 많은 활동을 보이고 게임을 승리로 이끌었다면 $M_{(1,1)}$ 에 속한 셀들에 대해 사용자가 머문 시간에 대한 데이터가 쌓이게 될 것이다. 그렇다면 그 다음 Stage에서는 $M_{(1,1)}$ 지역과 근접해 있는 $M_{(1,2)}$, $M_{(2,1)}$, $M_{(2,2)}$ 지역의 장애물을 단순화하여 기존의 지형 보다 NPC와 사용자가 더 많이 충돌 할수 있도록 만들어, 사용자가 이전 Stage보다 Map을 많이 활용하며 게임을 즐길 수 있도록 Map을 변화 시킬 것이다. 만약 사용자가 다양하게 Map을 이동하며 게임을 즐긴다면 전체

적인 Map의 셀들은 비슷한 수치의 데이터가 누적되게 될 것이다. 그렇다면 다음 Stage에 Map의 복잡도는 그대로 두고 전체적인 모양만 변화 시켜 같은 패턴의 행동으로 인한 게임의 몰입감 저하가 발생하지 않도록 한다.



[그림 2] 게임 지형의 영역 분할

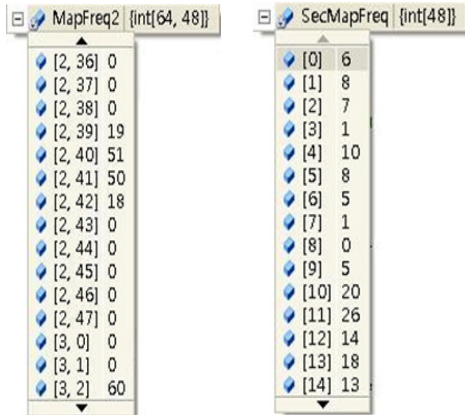


[그림 3] 게임 지형의 영역 분할2

3.2 데이터 분석

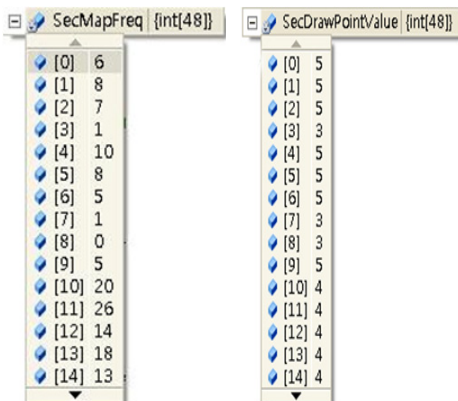
사용자가 게임 내에서 캐릭터를 움직이면 각 셀과의 충돌 체크로 움직인 좌표의 셀에 대해 시간당 누적도를 증가 시켜 어느 지역에 위치한 셀을 얼마나 지나쳐 갔는가에 대한 데이터를 수집한다. 게임 완료 후 각각의 구역에 대한 셀의 누적도의 합을 구한 후 셀의 개수로 나누어 지역에 대한 셀의 누적도 평균을 계산하였다. 누적도의 평균에 대하여 맵에 그려줄 장애물의 시작 포인트의 개수를 정해준다. 이 장애물의 시작 포인트의 개수는 게임

의 종류와 Map의 크기 게임의 첫 스테이지 난이도에 따라 가변적일 수 있다.



[그림 4] (좌) 게임 지형의 최소 단위 영역 설정 (우) 게임 지형의 그룹 영역 설정

예를 들어 게임에서는 2048*1572 픽셀의 크기의 맵을 제작 하였고, 그 맵의 각 셀은 32*32 픽셀의 크기의 64*48개의 최소단위 영역을 가진다고 가정하자. 그룹 영역당, 최소 단위 영역을 8*8의 픽셀로 64개를 가지게 된다. 전체 맵에서 그룹 영역의 개수는 8*6으로 총 48개의 영역으로 나뉜다. [그림 4]의 (좌)와 같이 플레이어의 이동 정보를 수집했다고 가정하자. 숫자가 '0'인 지역은 캐릭터가 가지 않은 위치의 영역이며 누적도는 당연히 '0'이다.



[그림 5] (좌) 셀의 누적도 평균 (우) Seed Point(장애물 생성 좌표 개수)

[표 1] Seed Point 생성 규칙

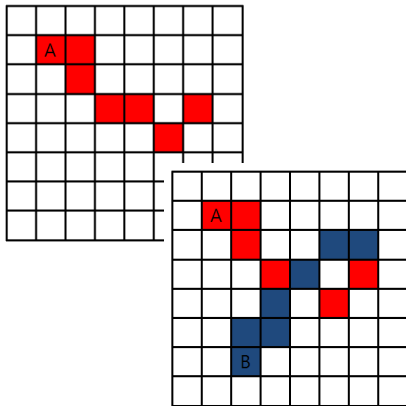
누적도의 평균	Seed Point의 개수
300 이상	Seed Point 17개
200 ~ 300	Seed Point 27개
100 ~ 200	Seed Point 37개
10 ~ 100	Seed Point 47개
10 ~ 1	Seed Point 57개

[그림 4]의 (우)와 [그림 5]의 (좌)는 각 영역의 누적도를 그룹 영역으로 묶어 평균을 구한 것이다. 영역의 위치는 [0]번 지역부터 X축부터 순서대로 Index번호가 들어가며 총 8*6 으로 나뉜다. [그림 5]에서 (우)와 같이 영역의 누적도를 구한 후에 그 영역에 맞는 Seed Point의 개수를 생성해야 한다. 그래야 다음 Stage에서 맵을 동적으로 변화 시켜줄 수 있기 때문이다. Seed Point의 개수는 개발자가 게임을 개발하면서 게임에 맞게 변화 시켜주어야 한다. 여기서 사용한 규칙은 [표 1]과 같다. Seed Point 개수는 게임의 종류와 난이도에 따라 가변적일 수 있다. 여기서 평균값이 '1'이하인 '0'이 나온다면 평균치인 Seed Point를 '3개'의 값으로 정하였으며 여기서 '3개'는 게임의 첫 Stage의 장애물이 생성될 때 적용되는 평균의 Seed point 값이다.

3.3 맵의 구조 변화

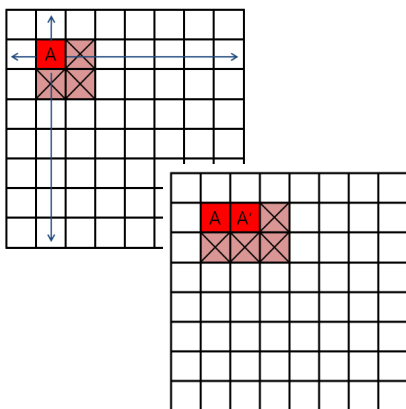
PC의 움직임에 대한 데이터를 구역 별로 받아 오면 위에 $i*j$ 로 나눈 크기의 영역에 새로운 Stage가 생성될 때 변화를 시켜준다. 기존에 있던 영역에 그림은 지워주고 복잡도에 따라 장애물을 배치하여 게임 환경을 다시 만들어 준다. 배경의 생성은 $i*j$ 로 나눈 하나의 영역에 임의 좌표에서 시작하여 X축과 Y축으로 0, 1 로 지정된 횟수만큼 한 칸씩 이동하면서 장애물을 생성해준다. 복잡도가 늘어날수록 맵에 생성되는 임의 좌표의 수가 증가한다. 임의의 좌표의 수에 최소 최대치는 위에서 언급한 것과 같이 게임의 종류, Map의 크기와 게임의 첫 Stage 의 Map에 난이도에 따라 달라질 수 있다. 본 논문에서 구현한 게임의 최소치는 1

개 최대치는 5개이며 평균 적인 랜덤 좌표의 개수는 3개로 정하였다. 랜덤 좌표의 수는 맵의 복잡도에 영향을 주며 사용자에게 다양한 맵의 형태를 보여주며 게임의 재미와 난이도를 증가시켜 준다.



[그림 6] 그룹 영역에서 임의에 A, B 위치에서 지형을 생성하는 모습

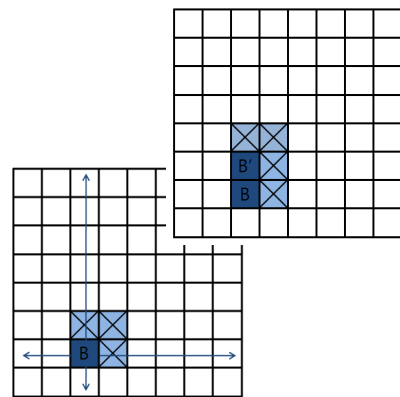
[그림 6]는 좌표에서 임의로 생성된 A혹은 B에서 시작하여 지정된 횟수만큼 장애물을 생성한다. 생성된 좌표에서 자신의 위치를 파악하여 상대적으로 먼 곳으로 움직인다. 그려진 좌표를 기점으로 빈 공간이 많은 곳을 우선으로 움직인다.



[그림 7] 'A' 장애물 생성 위치 비교

[그림 7]을 보면 장애물 'A'는 왼쪽 끝 좌표지점

에서 시작하였으며 영역에서 자신의 상대적인 위치를 검색 하여 자신이 위치한 지역에서 먼 쪽으로 이동하며 다음 장애물을 생성한다. X표는 앞으로 움직일 예상 후보지이며 (A', B') 지역은 한번 움직인 후 장애물이 생성된 장소이다. 마찬가지로 [그림 8]의 장애물 'B'는 상대적으로 왼쪽 아래 좌표에서 시작하였으며 다음 장애물은 오른쪽 위 좌표 쪽으로 치우치며 생성된다.



[그림 8] 'B' 장애물 생성 위치 비교

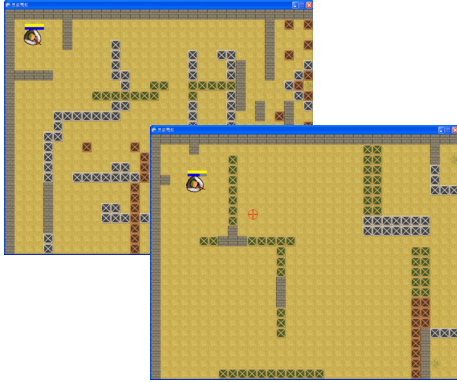
4. 실험

4.1 실험 환경

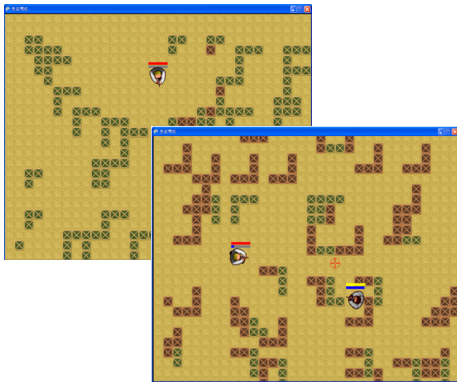
실험에서는 본 대학교 학부생 10명이 참여 하였다. 그리고 앞에서 설명한 방법을 바탕으로 Microsoft XNA Game Studio 3.0을 사용하여 간단한 탑뷰 (Top-View) 형식의 2D 액션 게임을 구현하였다. 게임에 등장하는 오브젝트로는 사용자 캐릭터, 적 캐릭터, 아이템, 장애물 등의 지형물이 존재하며 사용자가 입력할 수 있는 키(Key)로는 4가지 방향 키, 마우스와 무기를 고를 수 있는 숫자키가 있다.

본 실험의 목적은 사용자의 행동패턴에 따라 게임 환경이 변화하는 것을 보이는 것이며 환경이 변화에 따라 게임의 재미도와 몰입도가 얼마나 증가하였는지를 보이는 것이다. 이를 위해 기본적인

첫 Stage의 맵을 제공하고 이 환경에서 사용자의 행동패턴과 성향을 저장하여 다음 Stage가 생성될 때 적용하도록 설정 하였다.



[그림 9] 초기 게임에서 제공되는 지형
(좌) 1 Stage, (우) 2 Stage



[그림 10] 변화된 형태의 Map (좌) 한지역의 복잡도 낮춤, (우) 전체적인 복잡도 증가

위에 [그림 9]의 배경은 개발자가 초기에 만든 게임 화면으로 시스템 인공지능이 적용되지 않은 상태이다. [그림 9]의 (좌)를 완료 한다면 사용자의 패턴에 상관없이 자동으로 (우)의 Stage로 넘어간다. 맵은 정해진 패턴의 장애물이 존재하며 일정 목표를 패트를 하는 적 NPC와 대치하게 된다. 게임을 클리어 하기 위해선 적 NPC를 처치하고 나오는 반지를 일정 개수만큼 모으면 다음 Stage로 넘어간다. [그림 10]은 사용자의 행동 패턴을 파악

하기 위해 기본적인 Map인 [그림 10]의 (좌)를 제공한 후 사용자가 첫 Stage에서 행동한 정보를 바탕으로 난이도와 복잡도를 정한 후 그려진 다음 Stage의 모습이다. [그림 10]의 (좌)는 사용자가 첫 Stage에 한 지역에서 주로 머물면서 플레이를 한 경우로 그 지역에 복잡도를 낮추어 한 지역이 아닌 다른 지역으로의 이동을 유도하도록 한 Map 이며 [그림 10]의 (우)는 사용자가 첫 Stage에서 Map을 고루 활용하였을 때 나타난 Map으로서 Map의 전체적인 복잡도를 증가 시켜 전체적인 Map을 자유롭게 이동하지 못하도록 유도한 형태이다.

두 가지 형태의 게임을 비교하기 위해 본 대학교 학부생 10명에게 게임을 플레이 후 Difficult, Interest를 체크 하도록 하였으며 최저점은 1점이며 최고점은 50점 사이에 점수를 체크하도록 하였다. Play Time은 한 Stage 별로 체크 하였으며 2개의 Stage를 완수한 시간을 더하였다.

[표 2]는 [그림 9]의 형태로 진행되는 게임에 대한 평가표이며 평균 어려움 정도는 22.6이 나왔으며 재미도는 20.4가 나왔다.

[표 2] 정적인 Map에서의 실험 결과

	Test 1(Static Map)		
	Difficult (1~50)	Interest (1~50)	Play Time
A1	9	18	1m,16s
A2	17	22	1m,33s
A3	27	30	1m,45s
A4	10	10	1m,27s
A5	13	21	1m,07s
A6	39	25	3m,47s
A7	16	10	1m,20s
A8	30	15	1m,32s
A9	30	30	2m,12s
A10	35	23	2m,31s
Avg	22.6	20.4	1m,51s

[표 3]은 [그림 10]의 형태로 진행되는 게임의 평가표이며 평균 어려움 정도는 24.8 재미도는

25.1 이 나왔다. [표 2]와 [표 3]을 비교하면 동적인 게임 환경이 정적인 게임 환경에 비해 게임의 난이도와 Play Time은 증가하였지만 재미도는 증가한 것을 볼 수 있다.

[표 3] 동적인 Map에서의 실험 결과

	Test 2(Dynamic Map)		
	Difficult (1~50)	Interest (1~50)	Play Time
A1	11	27	1m,23s
A2	23	24	2m,11s
A3	27	25	1m,55s
A4	20	15	2m,05s
A5	15	24	1m,34s
A6	30	36	2m,55s
A7	20	25	2m,09s
A8	27	20	1m,27s
A9	40	30	3m,12s
A10	35	25	2m,11s
Avg	24.8	25.1	2m,10s

5. 결론 및 향후연구

본 논문은 사용자의 행동 패턴과 성향에 따라 NPC의 인공지능의 변화가 아닌 환경변화 인공지능을 만드는 것이다. 기존에 개발자에 의하여 만들어져 있던 게임 환경에서 벗어나 다양한 사용자의 성향과 행동 패턴에 대해 반응하여 정적인 게임 환경이 아닌 유동적인 게임 환경을 조성하는 것이다. 실험을 통해 이러한 방법을 사용한 것이 사용하지 않은 게임에 비해 사용자가 느끼는 재미도와 Play Time이 상승한 것을 볼 수 있었다. 이러한 결론은 본 연구에서 제시된 방법이 게임의 생명주기에 영향을 줄 수 있다는 것을 보인 것이다.

향후 연구로는 게임 환경에서 Map뿐만 아니라 다른 환경에 대한 변화를 동반하여 게임의 재미와 몰입도를 보다 증가시키는 연구와 실제 게임에서

위에 방법을 적용할 경우 생기는 현상에 대한 검증이 필요하겠다.

참고문헌

- [1] 조병현, 박창준, “게임 인공지능 연구동향”, 전자통신동향분석, 23권, 4호, 2008.
- [2] 박사준, 김성환, 김기태, “인공 유기체의 학습 행동이 게임 캐릭터의 전략에 미치는 영향”, 한국정보과학회 논문집, 29권, 1호, 2002.
- [3] 이면섭, 조병현, 정성훈, 성영락, 오하령, “대전 액션게임에서 유전자 알고리즘을 이용한 지능 캐릭터의 성능평가”, 전자공학회논문지, 41권, 4호, 2004.
- [4] 고찬, 조동면, “지능형 게임 캐릭터의 설계와 구현”, J. KSIAM IT series, Vol. 5, No. 1, 2001.
- [5] Youichiro Miyake, “Game AI Technologies for Online Games”, Korea Game Conference(KGC) 2007, 2007.
- [6] 연제혁, 김성수, 임형준, 이원형, “게임 난이도를 고려한 게임지형 자동생성 기법에 관한 연구”, 한국인터넷정보학회 추계학술발표대회 논문집, 제5권, 제2호, 2004.
- [7] 임형준, 최유표, 트리스탄 바사, 이원형, “유전자 알고리즘을 이용한 정적 오브젝트 자동 생성 기법에 관한 연구”, 한국인터넷정보학회 추계학술발표대회 논문집, 제5권, 제2호, 2004.
- [8] 오규환, “MMORPG 2.0을 향하여 : 플레이어에 의한 콘텐츠 메이킹과 다이내믹 게임월드”, Korea Game Conference 2006, 2006.



장새롬 (Jang, Sae-Rom)

2003년~현재 성결대학교 멀티미디어학부(학사과정)

관심분야 : 게임 인공지능(Game A.I.), 패턴인식



주문원 (Choo, Moon-Won)

1986년 캘리포니아 산호세주립대 수학과(학사수료)

1987년 뉴욕공과대학 전산학(공학석사)

1988년~1990년 삼성전자 종합연구소 연구원

1996년 스티븐스공대, 전산학(공학박사)

1997년~현재 성결대학교 멀티미디어학부 교수

관심분야 : 화상처리, 상황인지



이재열 (Lee, Jae-Yeol)

2004년~현재 성결대학교 멀티미디어학부(학사과정)

관심분야 : 게임 인공지능(Game A.I.), 게임 개발



윤태복 (Yoon, Tae-Bok)

2000년~2003년 (주)디지털솔루션

2001년 공주대학교 전자계산학과(이학사)

2005년 성균관대학교 컴퓨터공학(공학석사)

2007년 성균관대학교 컴퓨터공학(박사수료)

2005년~현재 성균관대학교 박사과정

관심분야 : 사용자 모델링(User Modeling), 게임 인공지능(Game A.I.)



최영미 (Choi, Young-Mee)

1979년 이화여자대학교 수학과(이학사)

1981년 이화여자대학교 대학원 전산학전공(이학석사)

1989년 Sydney University 전자계산학과(Visiting Scholar)

1993년 아주대학교 컴퓨터공학과(공학박사)

2001년 University of Pittsburgh 정보과학과(객원 교수)

1994년~현재 성결대학교 멀티미디어학부 교수

관심분야 : 게임 인공지능(Game A.I.), 교육용멀티미디어콘텐츠