

# 대용량 롤-플레이잉 게임을 위한 AOI 에 관한 연구

전일곤\*, 오삼권\*

\*호서대학교 벤처전문대학원

e-mail : [gonidal@lycos.co.kr](mailto:gonidal@lycos.co.kr), [ohsk@office.hoseo.ac.kr](mailto:ohsk@office.hoseo.ac.kr)

## A Study on AOI for Massively Multiplayer Online Role-Playing Games

Il-Gon Jeon\*, Sam-Kweon Oh\*

\*Graduate School of Venture, Hoseo University

### 요 약

MMORPG(Massively Multiplayer Online Role Playing Game)에서의 게임 서버는 모든 플레이어에게 게임 정보(오브젝트, NPC, 이벤트, 다른 플레이어들의 정보 등)를 실시간으로 전송해야 한다. 그러나, MMORPG 에 참여하는 플레이어의 증가로 인해 전송되는 정보의 양이 많아지면 네트워크 전송 지연과 서버의 부담으로 인해 플레이어들간 정보 전송의 문제가 발생한다. 이를 해결하기 위해 본 논문에서는 AOI(Area Of Interest, 관심영역)를 이용하여 자원 이동 및 동기화를 효율적으로 관리할 수 있는 구성요소들과 적용방안을 제시한다.

### 1. 서론

인터넷의 발전과 컴퓨터의 발전으로 인해 꾸준한 온라인 게임이 제작되고 있으며 온라인 게임을 즐기는 플레이어들도 증가하고 있다. 2 인에서 8 인 정도의 플레이어가 즐기는 소규모 온라인 게임에 비해 수 백 명 이상의 플레이어들이 동시에 동일한 가상환경에 접속하여 즐기는 MMORPG[1]는 많은 양의 정보들이 서버와 플레이어들 간에 전송 및 처리가 이루어 지기 때문에 대용량의 서버와 대용량의 네트워크 대역폭(Network Bandwidth)을 필요로 한다. 그러나, 대역폭과 서버의 증설에는 한계가 있다. 이러한 문제점의 해결 방안으로 전송되는 정보의 양을 줄이는 방법이 있으며, 플레이어가 불편을 느끼지 않는 범위에서 이루어져야 한다. 본 논문에서는 플레이어 주변에서 일정 범위의 정보만을 처리하는 AOI 를 제안하며, AOI 를 구성하고 있는 자원들의 분류와 적용방안에 대해 제시한다.

논문의 구성은 다음과 같다. 2 장에서는 관련연구를 살펴보고, 3 장에서는 AOI 에서 사용되는 자원들의 종류와 자원들의 적용에 대해 살펴본다. 그리고 4 장에서는 AOI 내에서 플레이어들간에 공유되는 자원의 동

기화에 대해 살펴본다. 마지막으로 5 장에서는 현재 AOI 의 문제점들과 차후 수정될 사항에 대해 알아본다.

### 2. 관련연구

#### 2.1 AOI 란

가상환경 연구에서는 일반적으로 참여자 별 인식영역을 AOI(Area Of Interest, 관심영역)[2]라고 하며, 게임 내에서는 플레이어 주변의 일정한 범위를 의미한다. AOI 는 게임 진행 시 플레이어에게 흥미를 유발시킬 수 있는 지역으로, 다음과 같은 사건들이 플레이어 주변에서 발생한다.

- 캐릭터(PC(Playable Character) / NPC(Non Playable Character))의 상태변화
- Object 의 상태변화
- 이벤트 지역의 상태변화
- 채팅(일반채팅)

플레이어의 AOI 내에서 위와 같은 사건이 발생하면 플레이어는 서버에게 발생한 사건에 대한 정보와

자신의 정보를 보내고 서버는 이러한 정보를 처리하여 요청한 플레이어와 주변의 플레이어들에게 갱신 정보를 보낸다.

2.2 View

게임 상에서 플레이어에게 보여지는 형태를 말하며 보여지는 시점에 따라 탑뷰(Top-view), 쿼터뷰(Quarter-view), 사이드뷰(Side-view), 프리뷰(Free-view) 등으로 나눌 수 있다. [표 1]과 같이 각 뷰에 따라 장·단점이 존재하며, 각 게임마다 특성에 맞게 뷰를 선택하게 된다. 요즘 나오는 3D 온라인 게임의 경우 플레이어의 시점이 자유로운 프리뷰를 많이 사용하고 있으며, 프리뷰의 경우 플레이어의 뷰에 따라 AOI의 범위도 바뀔 수 있다.

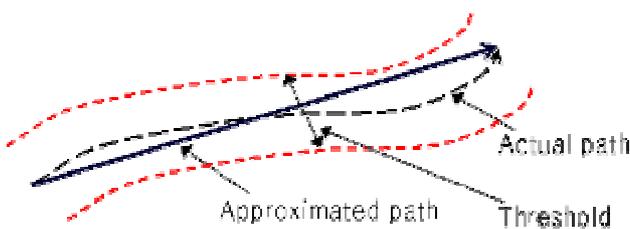
프리뷰를 많이 사용하고 있는 3D MMORPG의 경우 방대한 크기의 지형 데이터를 가지고 있다. 이러한 지형 데이터의 실시간 렌더링을 위해 플레이어의 현재의 위치에서 거리에 따라 지형의 Detail에 차이를 두는 LOD(Level Of Detail) 지형 알고리즘이 다수 제안되었다[3, 4].

[표 1] 시점에 따른 뷰

구분	설명
탑뷰	· 위에서 내려다보는 시점 · 다양한 전략 구사 가능 · 사물 묘사 한계 · 캐릭터의 움직임, 개성 표현 어려움
쿼터뷰	· 비스듬한 시점으로 내려다 보는 시점 · 캐릭터의 입체적 표현 공간의 느낌 잘 표현
사이드 뷰	· 캐릭터의 다양한 애니메이션 등으로 좀 더 생동감 있는 전투 가능 · 공간적 한계, 전략적 전투의 어려움
프리뷰	· 다양한 시점 · 사실적이며, 사물의 표현이 자유로움

2.3 Dead Reckoning

플레이어들과 서버간에 정보를 실시간으로 전송해야 하는 게임에서 발생하는 전송 지연(latency)을 해결하기 위한 기술인 Dead Reckoning은 Deduced-Reckoning인 “추론하여 계산한다”라는 뜻으로 위치 예측이라는 의미를 가지고 있으며 ‘추측항법’이라고 한다[5].



[그림 1] Dead Reckoning

Map 상에서 객체가 이동 시 플레이어는 서버에서 신호가 없을 경우 이동하고 있는 객체의 위치

(position), 방향(orientation), 속도(velocity), 가속도(acceleration)를 참조하여 정해진 시간 후의 위치를 예측하게 된다. 이 예측된 위치는 임의의 정해진 오차(threshold)에 벗어나지 않는 한도 내에서 플레이어의 Map 상에 적용된다. 즉 서버의 신호가 없어도 플레이어가 주변 객체들의 위치를 알 수 있다. 오차를 크게 잡을 경우 객체가 순간이동을 하는 것과 같은 급격한 움직임(Jerk)이 보여질 수 있다[6].

3. 자원(Resource)

자원이란, 게임상에서 서버 프로세스가 동작하는 중 내부에서 사용되는 메모리 영역에 위치한 데이터들을 말한다. 서버는 플레이어의 요청에 반응하여, 자원들의 생성, 참조, 조작 및 소멸 작업을 수행하면서 게임을 진행시켜 나가게 된다.

자원을 구성하는 객체들은 크게 동적 객체(Dynamic Object)와 정적 객체(Static Object)로 나눌 수 있다. [표 2]와 같이 동적 객체에는 PC, NPC, Item이 있으며, 정적 객체로는 Map이 있다.

동적 객체들은 항상 정적 객체인 Map 위에 존재하며, Item의 경우 PC/NPC에도 위치할 수 있다. 게임상에서 자원의 이동이나 데이터의 변경 시 객체를 참조하려 할 때, 객체 검색의 효율을 높이기 위하여 단계적으로 Map > PC/NPC > Item 순으로 검색한다.

서버에서는 이러한 자원들의 동기화와 동기화를 수행하는 객체(이하 동기화 객체)들의 관리, 그리고, 그에 관련된 인터페이스를 제공한다. 자원의 동기화 부분은 4장에서 설명한다.

[표 2] 게임 내의 Resource

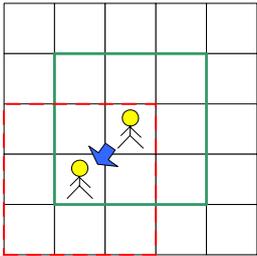
구성요소		설명	비고
동적 객체	PC	· 플레이어가 직접 플레이 하는 캐릭터	· 참조 가능하나, 데이터 조작 및 소멸이 불가능한 객체
	NPC	· 게임 서버의 AI엔진에 의해 플레이 되는 캐릭터	
	Item	· PC, NPC가 소유하거나 Map상에 존재할 수 있는 게임 내의 모든 아이템	
정적 객체	Map	· 게임의 Map에 대한 정보(지형정보, 특정 위치에 있는 PC, NPC, 아이템 정보)	· 게임 진행에 따라 생성, 조작, 소멸이 자유로운 객체

3.1 Map

Map 객체는 Map 좌표상의 데이터를 유지하면서 객체들의 이동 시에 이를 논리적인 좌표계 위에 유지하기 위한 객체이다. 플레이어에게 보이는 형식(3D 나 2D)에 상관 없이 서버에서의 Map은 2D의 형태를 가지며, 분할 방식에 따라 Hexagon Cell, Locales, Cell of ring, Aura, multi-layered aura 등으로 나누어진다[7].

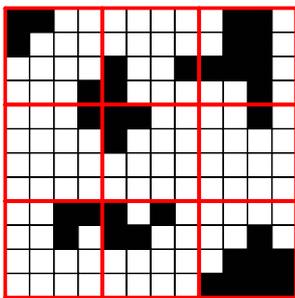
Hexagon Cell 방식은 Cell 이 [그림 2]와 같은 정사각형 모양을 가진 가장 기본적인 구조이며, 사용이 간편하다.

본 논문의 AOI 에서 사용될 Map 은 Grid 와 Cell 로 이루어진다. Cell 은 Map 을 구성하는 최소의 단위이고, 객체들이 위치할 수 있는 가상공간이다. Cell 의 N x N 집합은 Grid 를 구성하고, Grid 의 N x N 집합은 Map 을 구성한다. Map 을 Cell 과 Grid 로 나눔으로써 게임 Map 의 일정 범위 검색 시 효율을 높일 수 있다.



[그림 2] Hexagon Cell 방식

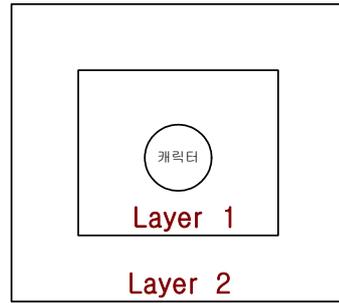
예를 들어, [그림 3]과 같이 12 x 12 의 Cell 로 이루어진 Map 영역에서 동적 자원을 검색하려 할 때, Cell 단위로 검색을 하게 되면 144 개의 Cell 을 검색해야 하지만, Grid 단위로 검색하게 되면 9 개의 Grid 를 검색하는 것으로 동일한 검색 작업을 할 수 있다. Grid 에 대응된 Cell 개수의 단위를 높이면 효율을 높일 수 있는 장점이 있지만, 검색의 정밀도는 그만큼 떨어지게 된다. 그러므로 Grid 에 대응된 Cell 개수 단위의 적절한 조정이 필요하다.



[그림 3] Cell 과 Grid

Map 에서 AOI 를 좀더 효율적으로 구성하기 위하여 플레이어의 AOI 에 포함되는 정보를 선별하는 필터링 기법을 사용한다. 본 논문에서 제시하는 필터링 기법은 공간 관계적 필터링(spatial structure filtering)으로서 게임내에서 객체간의 거리차만을 고려한 방법을 사용한다. [2, 8]

[그림 4]는 캐릭터를 중심으로 거리에 따라 두 개의 Layer 로 나눈 것이며, Layer 에 따라 서버에서 처리하는 정보가 달라지게 된다. Layer 1 은 동적 객체의 초기화와 이동정보, 상태변화 같은 Detail 한 Action 정보를 처리하며, Layer 2 는 동적 객체의 이동이나 초기화 같은 정보만을 처리한다.



[그림 4] 공간 관계적 필터링

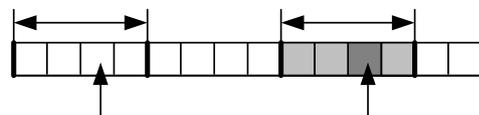
### 3.2 이동

AOI 에서 이동은 Map 에서의 PC/NPC 의 이동, Item 의 이동 같은 Map 에서의 동적 객체 이동을 말한다. 이동은 Cell 단위로 이동을 하게 되며, 다음 Cell 로 이동 시 플레이어는 서버에게 이동 가능 여부를 물어 보며 서버는 이에 대한 응답을 한다. 이때 서버는 PC/NPC 의 상태, 타일 속성, PC/NPC 충돌, 높이, 이벤트 등의 정보를 처리하여 AOI 내의 주변 플레이어들에게 이동에 대한 처리 정보를 전송한다.

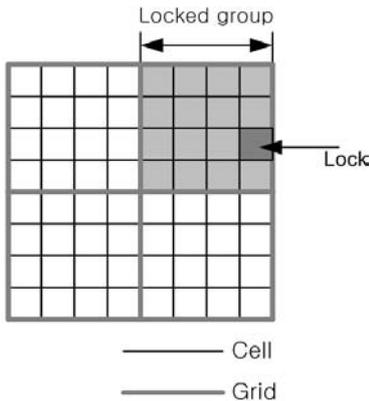
## 4. 동기화

멀티스레드(Multithreaded)로 구현된 온라인 게임 서버에서 여러 플레이어들이 공유되어 있는 동일한 하나의 객체에 접근하려 할 때 예상치 못한 결과가 발생할 수 있다. 이러한 문제를 방지하기 위하여 자원의 동기화가 필요하다. 객체에 대한 접근은 객체 핸들(Object Handle) 값으로 한다. 이 객체 핸들은 스레드(thread) 사이에서 공유되어 있으며, 공유된 핸들에 대하여 다른 플레이어(스레드)가 사용 중일 경우 나머지 플레이어들은 그 핸들에 접근을 하지 못하도록 한다. 동기화가 필요한 객체는 동적 객체, 정적 객체 모두 해당된다. 동적 객체는 객체 내부의 데이터 조회/변경 시, 객체 소멸 시 동기화를 하며, 정적 객체(Map)는 일정 범위 검색 시, Cell 위에서 동적 객체 핸들 이동 시, Cell 의 속성 조회/변경 시 동기화를 하게 된다.

동적 객체의 동기화 객체 수는 동적 객체의 개수에 비례하여 일정개수만 생성하며, 동적 객체의 일정 개수 별로 그룹화 한다. 그리고, 정적 객체의 동기화 객체의 수는 Map 의 크기에 비례하며, Grid 의 개수와 같고, Grid 별로 그룹화 한다. 즉 그룹내의 객체에 대하여 동기화를 할 경우 그룹전체를 동기화 시키게 된다. 이는 동기화 객체를 줄이고, 객체 핸들의 검색 수를 줄여 검색 속도를 높일 수 있다.



[그림 5] 동적 객체의 동기화



[그림 6] 정적 객체의 동기화

- No. 18, Elsevier, 2001.4
- [8] Mostafa A. Bassiouni, et al, “Performance and Reliability Analysis of Relevance Filtering for Scalable Distributed Interactive Simulation”, ACM Transaction on Modeling and Computer Simulation, Vol. 7, No. 3, July, 1997

## 5. 결론

온라인 게임에서 게임에 대한 최적화는 매우 중요하다. 특히 많은 플레이어들이 즐기는 MMORPG의 경우 더욱 최적화에 대한 중요성이 강조된다. 최적화는 플레이어의 어떠한 환경에서도 불편을 느끼지 않게 하는 것을 의미하며, 플레이어 환경(클라이언트)에서의 최적화도 중요하지만 온라인 게임 특성상 네트워크와 모든 플레이어들을 관리하기 위한 서버의 최적화가 더욱 필요하다. 이러한 네트워크와 서버의 최적화를 통하여 플레이어가 느끼는 불편을 줄일 수 있어야 한다. 본 논문에서 제시한 AOI를 통하여 많은 플레이어가 즐기는 MMORPG에서의 네트워크 트래픽과 서버의 연산을 줄여 서버와 네트워크의 효율을 높일 수 있다.

향후 서버와 플레이어간에 전송되는 AOI 내에서의 정보와 프로토콜을 줄일 수 있는 방안과 객체의 이동을 위한 Dead Reckoning, 그리고, 객체의 동기화에 대한 연구가 더 수행되어야 할 것이다.

## 참고문헌

- [1] 전재우, 오삼권 “대용량 롤-플레이팅 게임을 위한 규모조절 능력이 있는 고성능 게임 서버 구조”, 호서대학교 논문지 제 21 호, pp. 83 ~ 90, 2002
- [2] 김상욱, 이태중, 김성조 “대규모 분산 가상환경 상에서 관심영역의 부하 적응적 관리”, 정보과학회 논문지, Vol. 28, No. 7, pp. 0317 ~ 0330, 2001.08
- [3] 김대성, 한정현 “네트워크 환경에서의 대규모 지형 데이터 전송 및 렌더링”, 정보과학회 학술발표논문집, Vol. 30, No. 01, pp. 184 ~ 186, 2003. 04
- [4] 김혜선, 이동춘, 박찬용, 장병태 “대규모 지형의 3D 게임맵 구성을 위한 저작 시스템 개발”, 정보과학회 학술발표논문집, Vol. 29, No. 02, pp. 451 ~ 453, 2002. 10
- [5] [http://www.gameis.org/Korean/Game\\_Intro/Game\\_Genre/OnlineGame.htm](http://www.gameis.org/Korean/Game_Intro/Game_Genre/OnlineGame.htm)
- [6] Jesse Aronson “Dead Reckoning: Latency Hiding for Networked Games”, [http://www.gamasutra.com/features/19970919/aronson\\_01.htm](http://www.gamasutra.com/features/19970919/aronson_01.htm), September 19, 1997
- [7] Seok-Jong Yu “A Dynamic Message Filtering Technique for 3D Cyberspaces”, Computer Communication, Vol. 24,