

## 3D 체험형 어드벤처 게임 “HowSee” 개발\*

임창주\*, 정성택\*\*, 최승훈\*, 이창남\*, 송장섭\*\*\*  
 한국산업기술대학교 게임공학과\*, 한국산업기술대학교 컴퓨터공학과\*\*,  
 주식회사 비솔 개발팀\*\*\*

{scjlim, unitaek, shshoojjang, peaoth}@kpu.ac.kr, jsong101@hanmail.net

Development of Stereoscopic 3D Personalized Adventure Game: HowSee

C. J. Lim\*, S. T. Chung\*\*, Seung-Hoon Choi\*, Chang-Nam Lee\*, Song Jangsup\*\*\*  
 Dept. of Game Engineering Korea Polytechnic University\*, Dept. of Computer  
 Engineering Korea Polytechnic University\*\*, Visol Co. Ltd.\*\*\*

### 요 약

디스플레이 기술의 발달로 입체영상(S3D) 콘텐츠개발에 사회적 관심은 높지만 개발 방식의 표준화가 되어 있지 않아 인터랙티브 3D 콘텐츠제작 성과는 매우 저조하다. 본 논문에서는 적외선 카메라로부터의 실시간 영상을 입력으로 하는 주시선 기반의 사용자 인터페이스를 기반으로 개발한 3D 체험형 어드벤처 게임 “HowSee” 개발과정을 기술하였고, 그 과정에서 효율적인 3D 콘텐츠 제작방법을 제시하였다. 표준화되어 있지 않은 입체영상 기반의 게임콘텐츠 제작방법에 관해 본 개발과정에서 발생한 시행착오와 문제해결 방법 등은 입체영상 게임콘텐츠 제작을 효율적으로 하는 데 기초자료가 될 수 있다.

### ABSTRACT

Advances in display technology, stereoscopic (S3D) in the development of social interest in the content development approach is high but it does not make the standardization of interactive 3D content creation performance is very poor. In this paper, real-time video input from an infrared camera that is based on eye tracking implementation user interface developed 3D interactive adventure game “HowSee” was the process of developing skills in the process, an efficient 3D content production methods were presented. unnormalized S3D-based game content production process on how this development occurred in the trial and how to troubleshoot S3D games, etc. to streamline content creation can be a basis for.

**Keywords** : S3D(양안식 3D 입체영상), 3D Contents(3D 콘텐츠), Eye tracking(주시선 인식)

접수일자 : 2011년 07월 14일 일차수정 : 2011년 08월 17일 심사완료 : 2011년 09월 06일

교신저자(Corresponding Author) : 이창남

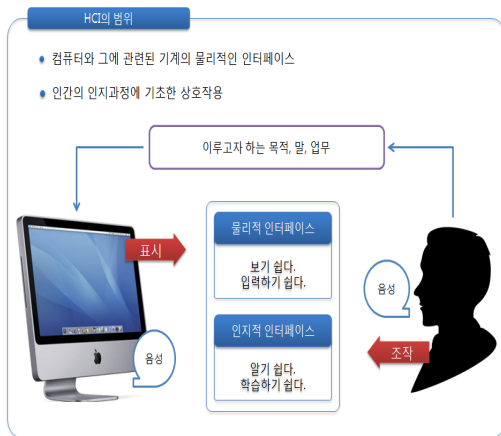
※ 본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음.

(NIPA-2011-C6150-1102-0001)

## 1. 서론

기술의 발달로 3D 입체영상 콘텐츠에 사회적 관심이 높아진데 반해 제작 성과는 매우 저조하다. 또한 3DTV 게임 쪽은 개발의 어려움과 더불어 정착되지 않은 작업 방식으로 인해서 개발이 거의 되고 있지 않는 상황이다. 또한 콘텐츠제작에 대한 3D 입체영상의 제작 방법의 표준화가 되어 있지 않아 제작이 힘들고 제작기간이 길어진다는 문제가 있다[1].

컴퓨터 게임의 경우 사용되는 인터페이스는 주로 마우스와 키보드이다. 이런 인터페이스를 인간적 인터페이스(Human Interface)라고 하며 가장 기본적인 인터페이스 구조에 속한다. 그러나 기술이 발전하면서 점점 인간과 컴퓨터 간의 상호작용에 관한 연구(HCI - Human Computer Interaction)의 중요성이 커져가면서 그와 관련된 인터페이스가 개발되고 있다. 정보의 표현매체가 문자위주에서 SILK(Sound, Image, language, Knowledge)로 전이되면서 인간과 컴퓨터 간의 상호 작용에 관한 연구의 중요성은 더더욱 커지고 있다[2].

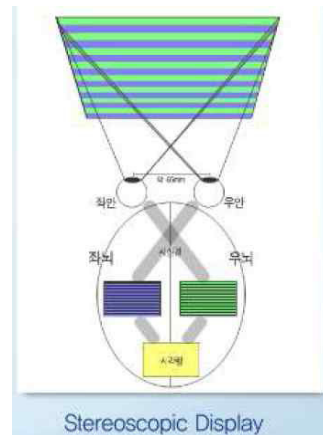


[그림 1] 인간과 컴퓨터 간의 상호작용에 관한 연구의 중요성 및 범위

이러한 인간과 컴퓨터 간의 상호작용에 관한 연구의 중요성에 부합되어 개발된 인터페이스가 바로

3D 주시선 추적 시스템이다. 다양한 3D 입체영상 콘텐츠와 기능성 게임 콘텐츠에 활용 가능한 주시선 추적 시스템을 통해 눈의 주시 정보를 획득하여 게임 개발과 3D 콘텐츠 제어를 위한 정보처리 모형 개발 및 사람의 주시정보를 파악하고 3D 콘텐츠와 기능성 게임 콘텐츠 제어에 적합한 정보처리 모델이다.

이 모델은 현재 가장 주목을 받고 있는 3D 입체영상 구현 방식인 스테레오스코픽 방식으로 양안 시차를 구현하기 위한 2대의 HD 카메라를 이용해서 제작되어 3D 디스플레이에 구현된 콘텐츠를 3D 안경으로 시청하는 방식에 마커와 적외선 카메라라는 입력 장치를 더한 인터페이스이다[3].



[그림 2] 스테레오스코픽 디스플레이 방식

개발된 인터페이스를 기반으로 게임을 제작하는데 있어서 플레이어의 흥미를 유발할 수 있도록 간단한 시나리오가 가미된 어드벤처 기반의 모델하우스를 통해 실제 집안을 돌아다니는 느낌을 유도하는 것이 제작 목적이다.

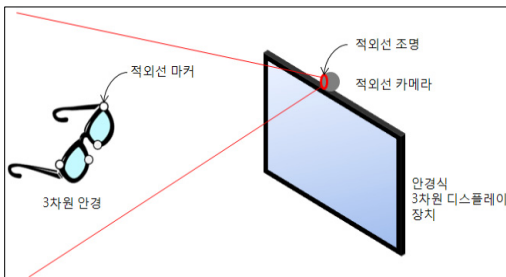
플레이어는 S3D(양안식 3D 입체영상)의 효과를 통해 마치 집안에 존재하는 것 같은 느낌을 들게 되며 주시선 추적 시스템을 이용하여 3D 안경에 달려있는 반구형 마커를 통해 위치 인식을 하여 집안에 존재하는 아이টে를 찾고 스토리를 진행한다.

‘3D HowSee’를 개발하기 위해 게임엔진으로는 OGRE3D를 사용하고 그래픽 제작 툴로는 3D max와 Photoshop CS3를 사용하였다.

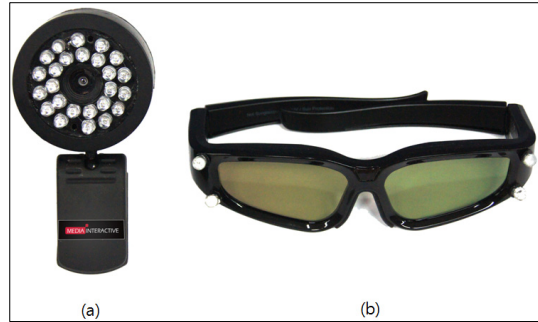
## 2. 본 론

### 2.1 주시선 추적 시스템

[그림 3]과 [그림 4]는 콘텐츠 제어 시스템을 개략적으로 나타낸 구성도이다. 사용자가 착용한 안경의 특정 위치에 4개의 마커가 부착된다. 사용자에게 콘텐츠를 제공하는 영상 기기는 촬영되는 마커의 형상의 변화를 참조로 하여 사용자의 움직임 정보를 획득한다. [그림 3]의 영상 기기는 안경에 부착된 마커로부터 획득한 사용자의 상태 정보(사용자의 위치, 움직임 및 얼굴 시선이 향하고 있는 방향 등에 대한 정보)를 참조로 하여 제어되는 콘텐츠를 디스플레이하기 위한 연산부를 구비한 모든 영상 기기(TV, 모니터가 구비된 데스크톱 컴퓨터나 노트북 컴퓨터, PDA, 스마트폰, 태블릿 등)를 모두 포함한다. 안경에는 가령 4개의 마커가 부착될 수 있는데, 이 경우 안경의 상측 부분과 하측 부분에 각각 2개의 마커가 나뉘어져 부착될 수 있다. 적외선 카메라는 마커가 반사할 수 있는 특정 과장 대역의 빛을 조사하고, 마커로부터 반사되어 되돌아오는 빛을 분석함으로써 안경을 착용한 사용자의 주시선과 움직임 등을 포함한 정보를 분석할 수 있다[4].

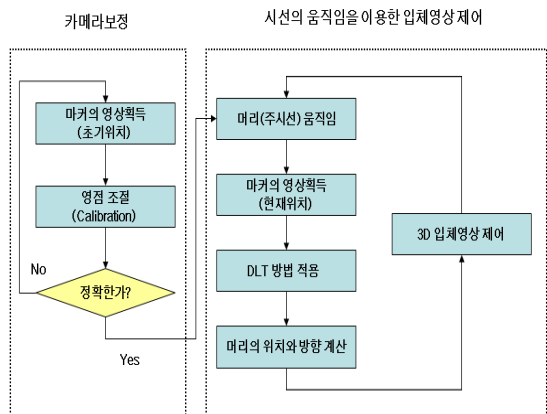


[그림 3] 주시선 추적 시스템의 구성도



[그림 4] 주시선 추적 시스템의 구성  
(a) 적외선 조명을 설치한 적외선 카메라  
(b) 마커가 부착된 3D 안경

### 2.2 Marker tracking 알고리즘 개발



[그림 5] Marker tracking 알고리즘

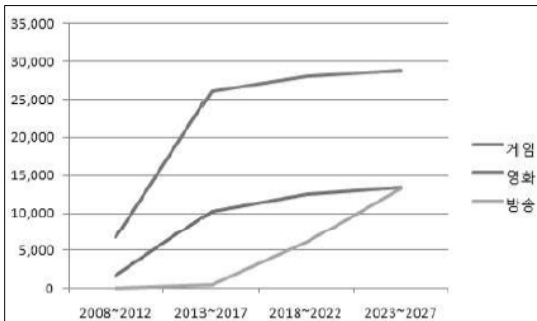
적외선 카메라에서 얻은 영상을 이용하여 마커를 찾기 위한 작업은 [그림 5]와 같은 순서대로 이루어진다. 적외선 조명에 의해 마커가 뚜렷하게 구별되는 명암과 대비를 먼저 조절하여 영상처리상의 에러를 최대한 제거한다. 그리고 출력된 영상을 가지고 이진화를 거쳐 마커를 검색하였다. 이미지 상의 마커를 검색하는 방법은 여러 가지가 있으나 본 논문에서는 윤곽선 정보를 이용하여 마커의 위치를 추출하였다.

[그림 5]에서 찾은 마커의 이미지상의 위치를 이용하여 마커를 착용한 사용자의 3D 좌표와 방향을 알아내기 위한 프로세스는 위의 순서도와 같다. 일

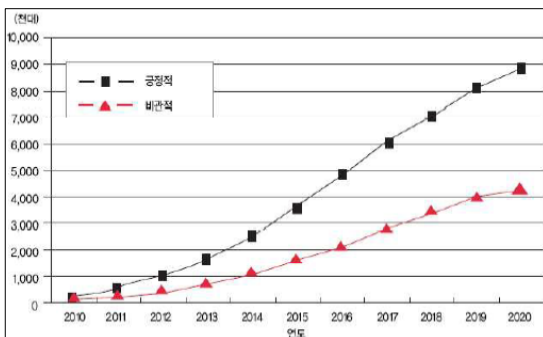
단 4개의 마커 중에서 거리가 가장 먼 2개의 마커가 상단에 있는 마커라는 가정 하에 상단과 하단의 마커를 분별한다. 각 마커의 위치가 분별되면, 이전에 구해놓았던 카메라 내부인자를 가지고 카메라 외부인자 행렬을 계산한다. 그리고 이 행렬에서 마커의 3D 좌표와 방향을 추출하였다[5].

이러한 주시선 추적 시스템을 기반 인터페이스로 한 ‘3D HowSee’이다. 기존의 모델하우스와 마찬가지로 집안을 돌아다니지만 주시선 추적 시스템의 가장 큰 장점인 프로그램과 유저간의 쌍방향 상호작용을 이용해서 시나리오가 존재하는 오리지널 게임을 제작함으로써 기존의 수동적인 콘텐츠가 아닌 유저가 흥미를 느끼면서 자연스럽게 집안의 구조를 파악할 수 있는 엔터테인먼트 요소가 가미된 콘텐츠를 개발하는 데 의의가 있다.

### 2.2.1 3D콘텐츠 개발 전망



[그림 6] 국내 3D 서비스 시장 규모 전망



[그림 7] 국내 3DTV 시장 전망

[그림 6]과 같이 2012년 이후 국내의 3D 서비스 시장 규모는 계속해서 성장할 전망이다. 또한 3D TV 또한 계속해서 보급될 전망이다[6].

그러나 이러한 기대와는 달리 3D TV와 관련된 오리지널 게임은 전무한 상황이고 2D 게임을 3D로 컨버팅한 방식의 게임을 내놓은 상황이다. 3D 컨버팅 방식이 위험이 적다는 장점 때문이다. 그러나 이러한 컨버팅 방식은 위험이 적다는 장점은 있지만 고품로, 저실감이라는 효율 문제와 3D 입체영상 게임 제작과정에서 얻을 수 있는 많은 시행착오와 노하우는 얻을 수 없다는 단점이 있다. 이러한 단점들을 보완하기 위해 이 연구는 체험형 3D 입체영상 콘텐츠 ‘3D HowSee’를 개발을 하면서 발생하는 문제점을 찾고 해결책을 제시 할 것이다.

## 3. 콘텐츠 제작과정

### 3.1 3D HowSee 개발 환경

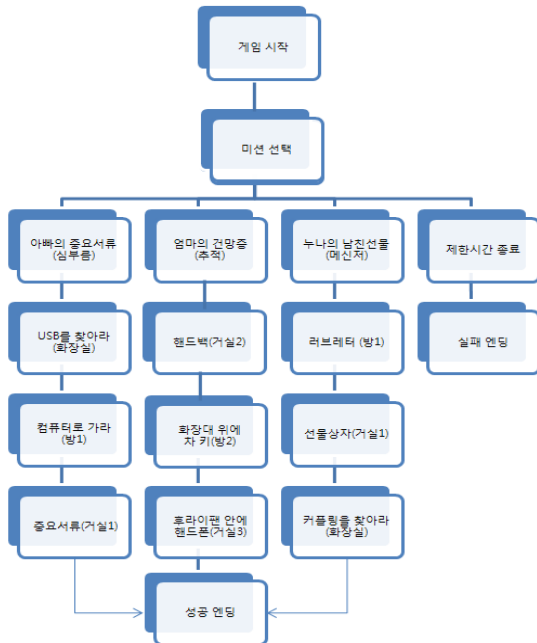
‘3D HowSee’의 플랫폼은 PC이며 입력장치는 적외선 카메라, 안경에 부착 되어있는 마커를 사용한다. 게임 개발 환경은 OGRE3D 엔진 및 기타 프로그램을 사용하였다.

OGRE3D 엔진은 Object-oriented Graphic Rendering Engine의 약자로서 2001년 영국의 Steve ‘sinbad’ Streeting 라는 소프트웨어 엔지니어가 개발을 시작하였다. 객체지향 인터페이스 방식의 설계를 통한 깔끔하고 간결한 엔진 클래스 및 인터페이스 제공과 Direct3D와 OpenGL을 동시 지원함으로써 확장 용이한 프레임워크 구조를 통해, 쉽고 빠른 게임 제작이 가능한 장점을 지니고 있다. 그래픽 데이터를 띄우는데 있어서 좋은 범용성과 쉬운 접근성을 가지고 있다[7]. UI작업을 위해서 사용한 프로그램은 CEGUI이며 2D 기반의 그래픽 자원을 띄우는데 용이하다. 사운드프로그램은 다양한 사운드편집이 가능한 GoldWave를 사용하였다.

### 3.1.1 기획과 시나리오

기존의 3D 입체영상 게임은 기능성 게임에 가깝고 단순한 게임성을 지니는 것에 불과했다. 그러나 3D 관련 전문가들의 의견을 인용해 보자면 다음과 같다. “좋은 3D 콘텐츠를 제작하려면 입체영상 구현방식을 어떻게 할 것인가의 문제보다 기획 부분에서 구현하는데 적합한 스토리를 발굴하는 것이 중요하다[8].”

이것은 2D 제작방식과 3D 입체영상의 제작방식에 있어서 많은 차이가 있음을 말하고 있다. 따라서 기획단계에서 UI부분은 2D로 게임은 3D로 제작하여 2D제작방식과 3D제작방식의 특징과 장단점을 알아보기로 하였다. 또한 모델하우스에 맞추어 시나리오를 제작하여 게임에 반영하기로 하였다.



[그림 8] '3D HowSee'의 시나리오 흐름도

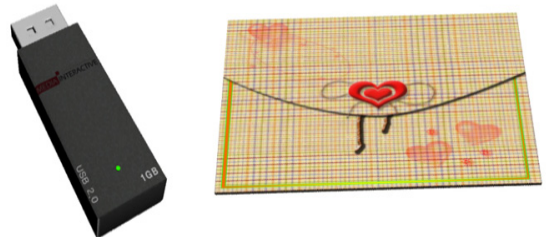
### 3.1.2 스토리텔링

스토리텔링을 가장 잘 구현하기 위하여 3D 입체영상의 효과를 고려할 수는 있으나 3D 입체영상이 주는 깊이감, 몰입감, 입체감을 살리기 위하여 스토리텔링을 인위적으로 구성하는 것에 대해 전문가 대부분은 부정적 의견을 제시한다[9].

이러한 전문가들의 견해를 해결할 수 있는 방법으로 2D 방식의 그래픽 UI 방식을 채택했다. 즉 그래픽 자원은 3D로 개발하여 3D 입체영상의 깊이감, 몰입감, 입체감을 살리고 2D로 UI를 제작함으로써 자연스러운 스토리텔링을 유도한다.

### 3.2 시퀀스 제작

게임 내에 스토리를 전개하기 위해서 만든 것이 시퀀스라는 개념이다. 시퀀스별로 키아이템이 존재하고 플레이어는 집안의 특정 장소에 배치되어 있는 키아이템을 찾게 되면 다음 시퀀스로 넘어간다. 집안 곳곳에 퍼져 있는 아이템을 찾으면서 플레이하는 3D 입체영상 게임의 효과를 인식하게 된다.



[그림 9] '3D HowSee'의 키아이템

### 3.3 그래픽 자원 제작과정

그래픽 자원을 제작함에 있어 기획파트에서는 개발환경과 엔진의 특징 등을 파악해서 확실한 아웃라인을 정해줘야 한다. 특히나 최대 총 폴리곤 개수는 10만 개 정도였고 아무런 제한이 없는 상태에서 한 달여에 걸쳐서 vertex(점) 위주로 작업을 진행했다. 채질 역시 제한이 없었고 최대한 현실의 집에 가깝게 작업한 결과물은 고품질 상태의

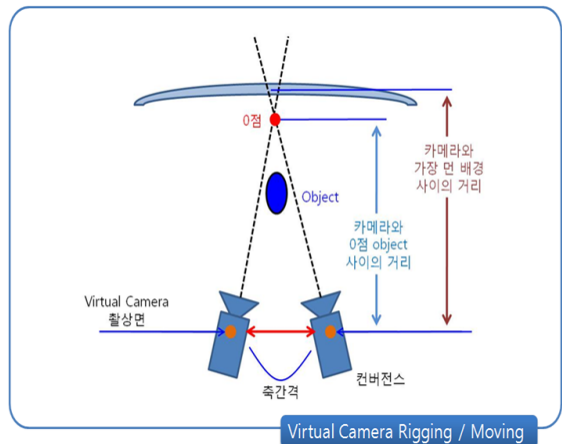
그래픽 자원이다.

고품질 상태의 그래픽 자원은 최적화하기에도 매우 어렵지만 그래픽 품질이 매우 높다는 장점이 있다. 따라서 고품질 작업 방식은 클라이언트가 무겁지 않을 것이라는 가정 하에 진행되는 경우가 많다. 대 폴리곤의 개수, 그래픽리소스가 사용될 용도, 재질 등을 그래픽 디자이너에게 확실히 인지 시켜 작업을 지시하지 않게 되면 예기치 않은 문제점이 발생해서 몇 번이고 재작업을 해야 하는 경우가 발생하였다.

초기에 아무런 제한 조건 없이 그래픽 제작을 했을 때 총 폴리곤 개수는 10만 개 정도였고 아무런 제한이 없는 상태에서 한 달여에 걸쳐서 vertex(점) 위주로 작업을 진행했다. 재질 역시 제한이 없었고 최대한 현실의 집에 가깝게 작업한 결과물은 고품질 상태의 그래픽 자원이다.

고품질 상태의 그래픽 자원은 최적화하기에도 매우 어렵지만 그래픽 품질이 매우 높다는 장점이 있다. 따라서 고품질 작업 방식은 클라이언트가 무겁지 않을 것이라는 가정 하에 진행되는 경우가 많다.

그러나 3D 입체효과를 위해서는 최소 30프레임에서 최대 60프레임이 나와 줘야 한다는 문제점이 발생하였다. 그래픽 품질이 올라갈수록 게임 내에 들어가는 그래픽 데이터가 증가되어 클라이언트가 무거워지고 프레임이 떨어진다. 따라서 그래픽 자원을 4만 폴리곤 이내로 줄이고 vertex(점)위주였던 고품질 상태의 그래픽 리소스를 면(poly)위주의 저 품질 상태로 전환하였다.



[그림 11] 3D 입체효과를 위해 30~60프레임 유지



[그림 10] 제한 없이 제작된 고품질 상태의 그래픽

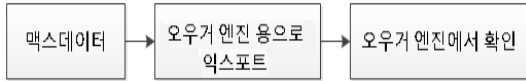
[표 1] 그래픽 품질에 따른 평균 프레임 체크

| 회수 | 저 품질 | 중 품질 | 고 품질 |
|----|------|------|------|
| 1  | 75   | 34   | 5    |
| 2  | 73   | 36   | 4    |
| 3  | 65   | 40   | 8    |
| 4  | 70   | 32   | 4    |
| 5  | 65   | 35   | 2    |
| 6  | 80   | 46   | 3    |
| 7  | 77   | 44   | 4    |
| 8  | 70   | 50   | 5    |
| 9  | 66   | 57   | 5    |
| 10 | 78   | 50   | 6    |

저 품질 작업을 진행한 결과 그래픽 품질은 많이 떨어졌지만 입체효과를 내기에 충분한 프레임이 확보되었다. 수정작업을 거쳐서 제작된 MAX 데이



터는 오우거맥스(OgreMax)라는 추출 프로그램을 거쳐 OGRE3D 엔진에서 사용가능한 추출 파일로 전환된다.



[그림 12] 그래픽 데이터 처리 과정

추출 과정에서도 생각하지도 못한 치명적인 문제점이 발견되었다. 바로 OGRE3D 엔진에서 지원되지 않는 재질 문제인 것이다. 초반에 아무 제한 없이 재질을 사용했던 그래픽 데이터를 OGRE3D 엔진용으로 추출 되어 뷰어 상으로 확인해 본 결과, 재질이 표현되지 않는 현상이 발생하였다.



[그림 13] 오우거상에서 재질이 표현되지 않는 문제점

이것이 OGRE3D 엔진의 문제점이다. 게임엔진의 특성을 파악하지 않은 점이 재작업을 하도록 하게 하는 원인이다. 재질에 대한 문제점을 파악해 본 결과 OGRE3D 엔진에서는 기본 재질을 지원한다. 재질 문제뿐만이 아니고 3D 입체 효과를 위해서는 셰이더(Shader) 또한 중요한 역할을 한다. 셰이더를 이용해서 라이트를 주기 위해서는 그래픽 데이터에 어느 정도의 vertex가 필요한데 poly 위주로 되어 있는 로우폴리곤 상태에서는 라이트를 충분히 줄 수가 없다. 즉 poly를 적당히 나눠서 vertex로 공간을 확보해 줄 필요가 있었다. 즉 저 품질을 6만 폴리곤 정도의 중 품질 형태로 재수정할 필요가 있었다.

### 3.4 UI 제작

#### 3.4.1 UI 디자인

‘3D HowSee’는 앞서 했던 설명과 같이 3D 입체영상으로 보여 지는 콘텐츠이기 때문에 UI 디자인에 있어서 신중할 필요가 있다. 그러나 개발 초기에는 화려한 성과물을 보여주고자 하는 욕심에 이를 간과하여 불필요하게 많은 색을 사용한다거나, 형태가 복잡한 디자인을 추구하여 3D 입체화시킨 영상을 보게 되면 불가피하게 생기는 약간의 잔상들이나 흐릿한 효과들이 사용자의 눈을 피로하게 만들었던 적이 있다. 때문에 이후에는 메뉴, 버튼등 대부분의 UI를 Rectangle 형으로 교체하고 사용되는 색상도 한 화면 당 3개 이내로 최대한 절제하였다. 이렇게 단순하게 디자인된 UI는 단순 명료하면서도 원하는 정보를 빠르게 플레이하는 유저에게 전달할 수 있도록 하였다. 또한 디자인된 UI를 클라이언트에 바로 띄울 수 있는 것이 아니고 CEGUI를 이용하여 레이아웃 프로그래밍을 실행하여 OGRE3D 엔진과 호환이 되게 제작하였다 [10].



[그림 14] 흑, 백의 색상만을 사용한 UI 디자인

#### 3.4.2 UI 프로그래밍

‘3D HowSee’를 제작하는데 있어서 핵심적인 부분은 마우스, 키보드 등을 전혀 사용하지 않고 사용자가 원하는 메시지를 프로그램에 입력해야 한다

는 것이었다. 때문에 일반적인 기존의 UI들에서는 소스코드 몇 줄 만으로 이용할 수 있는 메시지 전송 방식을 사용하지 못하고 이와 비슷한 효과를 내는 알고리즘을 직접 제작하였다[11].

우선 화면상의 레이아웃(Layout)안에 고유의 레이아웃 변수를 선언 후 마우스 포인트 모양의 그림을 그 안에 띄우고 메인 레이아웃과 마우스 포인트의 레이아웃이 트리 구조로 서로가 부모-자식 간의 노드처럼 취급되게 하고 이를 트래킹 포인트로 명명하였다.

그 다음에는 트래킹 포인트의 현재 좌표가 매 프레임마다 트래킹 좌표를 인식해 이동하게 처리하고 OGRE 3D, OPENCV, CEGUI가 모두 사용되는 부분이라 서로 다른 엔진들을 호환되게 하는 데는 여러 가지 방법이 필요하다.

마지막으로는 원하는 오브젝트가 트래킹 포인트에 반응하도록 해야 했는데, 초반에 이 부분에서는 서로의 좌표 값을 이용해 충돌 체크 하는 방법을 사용 했지만, 매 프레임마다 여러 오브젝트와 트래킹 포인트의 좌표 값을 읽어가며 거기에 충돌체크 부분까지 프로그램을 구동하게 하였으나 프레임수가 현저하게 떨어지면서 다른 부분에서의 연산까지 지장을 주는 바람에 다른 방법을 찾아야 했다. 그래서 사용한 방법이 오브젝트의 포인터 값과 트래킹 포인트의 포인터 값을 읽어와 서로간의 포인터가 충돌할 때에만 연산이 이루어지도록 프로그래밍 하는 것이었다. 이를 이용해 트래킹 포인트가 해당 오브젝트 위에 일정 시간 이상 머물러 있다면 윈도우 마우스의 클릭 이벤트와 비슷한 방식의 메시지 전달로 프로그램이 반응하도록 제작하였다.

### 3.5 프로그래밍

#### 3.5.1 3D 입체효과 기술구현 방법과 해결방법

스테레오스코픽 방식의 핵심은 2대의 카메라를 교차시켜 3D 효과를 발생시키는 것이다. 즉 카메라의 좌표와 위치, 거리, 포커스를 어디다 두어야 하는지가 최대 관건이었고 구현의 핵심 사항이다.

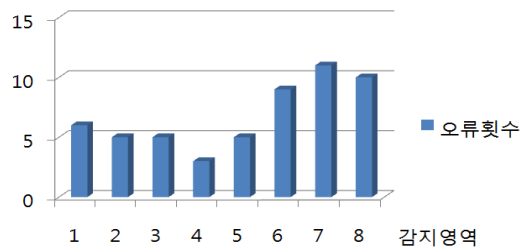
인터페이스 상의 적외선 카메라는 한 대 뿐이므로 코딩으로 노드를 만든 뒤 가상의 두 대의 카메라를 붙여서 제작하였다. 위치와 거리 설정을 수 없이 바꿔가며 최적의 3D 효과를 위해서 시행착오를 거듭했고 2대의 카메라 위치를  $+ - 0.0268893f$ 로 설정을 했을 때 최적의 3D 효과가 나오는 것을 확인하였다. 그리고 스테레오스코픽 방식대로 두 대의 카메라를 좌우를 한 번씩 출력하게 하였으나 이 방식은 치명적인 결함이 있었다. 바로 안정된 프레임이 유지할 수 없다는 것이었다. 그 결과 화면의 싱크 역시 불규칙하고 싱크가 튀는 현상이 있었다.

이 문제를 해결하기 위해서 고려한 것이 출력 방식을 스테레오스코픽 방식에서 Side By Side 방식으로 우회하는 방법이다. Side By Side 방식은 좌우 동시에 같은 화면을 출력하는 방식이고 평균 프레임은 45~60 프레임이고 조화율 또한 안정적이다.



[그림 15] Side By Side 출력 방식

#### 오류횟수



[그림 16] 출력방식에 따른 프레임 저하 오류 횟수



### 3.5.2 주시선 트래킹 구현 방법과 해결 방법

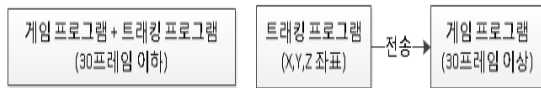
프로그램 상에 트래킹 함수를 사용하면 매 프레임마다 항상 트래킹을 호출하는 부분과 애니메이션이나 사운드 등 게임 코드적인 부분이 동시에 실행되어 120 프레임이 나와야 안정적인 3D 효과를 체감할 수 있지만 10~20 프레임 내외로 떨어진다. 현재 시중에 나온 보통의 캠은 30프레임 정도로 호출을 받기 때문에 120프레임이 나올 수가 없다. 최대 나올 수 있는 프레임은 테스트를 통해 확인할 결과 30 프레임 정도이다.

이것을 해결하기 위해서 기존에 한 프로그램으로 통합되어 있던 트래킹 시스템과 게임프로그램을 분리시킨다. MFC에서 사용하는 윈도우메시지 출력 방식을 응용해서 다른 프로그램으로 메시지를 보내는 방법을 연구한 뒤 게임 프로그램 내에선 트래킹 프로그램에서 넘겨주는 x, y, z 값을 가져와 게임을 구동시킨다. 그 결과 30 프레임 이상이 확보되어 트래킹 시스템과의 연동에 문제가 발견되지 않는다.

한다. 그렇지 않으면 개발 과정에서 발견되는 문제 뿐 만이 아니라 재작업이라는 문제에 봉착하였다. 그래픽은 여러 시행착오 결과 중 품질 상태의 그래픽 품질이 주시선 인터페이스 기반의 3D 입체영상 콘텐츠를 개발하는데 가장 적합하다.

OGRE3D 엔진은 좋은 호환성과 범용성을 가지고 있지만 기본 재질 밖에 지원되지 않는다는 단점을 안고 있다. 프로그래밍은 프레임 유지가 관건인데 최소 30프레임 이상이 나와야 최소한의 3D 입체영상 효과를 기대할 수 있다. 프레임이 제대로 나와 주지 않을 때는 Side By Side 출력방식으로 문제를 해결 할 수 가 있었다.

‘3D HowSee’의 제작 경험은 스테레오스코픽 방식으로도 충분히 가능성이 있다는 것을 알게 해주었다. 물론 많은 시행착오가 있고 앞으로도 이러한 경험들과 문제해결 방식은 스테레오스코픽 제작방식 표준화의 밑거름이 될 것이라 믿는다.



[그림 17] 주시선 트래킹 해결 방법

## 4. 결 론

기술이 발달되고 인간과 컴퓨터 간의 상호작용에 관한 연구의 중요성이 커져가면서 개발된 주시선 인터페이스는 3D 입체효과를 극대화시켜주고 유저와 소프트웨어 간의 상호작용을 가능하게 해준다. 이러한 인터페이스를 기반으로 ‘3D HowSee’를 제작함에 있어서 아무런 개발 경험과 노하우가 알려지지 않아서 많은 문제들이 발생 하였다.

기획은 주시선 인터페이스의 특징과 OGRE3D 엔진의 특징, 그래픽 3D 폴리곤의 개수, 재질 등 개발 제반 사항은 반드시 파악하여 기획을 해야

## 참고문헌

- [1] 김주현, “3D헤딩슈터 입체영상 콘텐츠 개발”, 한국컴퓨터게임학회논문지 Volume2, NO.24 June 2011, pp. 1-2, 2011.
- [2] 윤세한, 인지공학 - What is HCI, 삼성경제연구소, 2010.12.16.
- [3] 김주현, “3D헤딩슈터 입체영상 콘텐츠 개발”, 한국컴퓨터게임학회논문지 Volume2, NO.24 June 2011, pp. 3, 2011.
- [4] C. J. Lim, Y.M. Lee, Yun-Guen Jeong “Development of eye tracking interface for 3D contents control”, Vol. 24, No.1, pp.13~20, March 2011.
- [5] C. J. Lim, Lee Hun, Yun-Guen Jeong “User interface Based on eye tracking Implementation and Its Analysis”, pp. 3, 2011.
- [6] 김용관, “3차원 입체영상 산업 시장 현황과 기술 표준화 전략에 관한 연구”, pp. 3-5.
- [7] 이대현, “게임엔진론-게임엔진 개요 및 OGRE3D 엔진 소개” Vol .1, pp. 25-40, 2008.
- [8] KOCCA포커스, “국내 3D 입체영상 제작의 현황과 미래”, pp. 7, 2010.

- [9] KOCCA포커스, “국내 3D 입체영상 제작의 현황과 미래”, pp. 17, 2010.
- [10] 이대현, “게임엔진론-게임엔진 개요 및 OGRE3D 엔진 소개” Vol .1, pp. 41-42, 2008.
- [11] 남문초, “GUI 디자인 표준화에 관한 연구”, pp.4-6p. 2011.



임 창 주 (C. J. Lim)

KAIST 산업공학과 졸업  
KAIST 산업공학과(HCI/VR) 석사  
KAIST 산업공학과(HCI/VR) 박사  
한국과학기술기획평가원 선임연구원  
펜실베이니아주립대학교 컴퓨터공학과 초빙연구원  
한국산업기술대학교 게임공학과 부교수

관심분야 : HCI/VR, 교육용 Game Design



최 승 훈 (Seung-Hoon Choi)

한국산업기술대 게임공학과

관심분야 : 3D 프로그래밍



이 창 남 (Chang-Nam Lee)

한국컨텐츠진흥원 게임아카데미 졸업  
한국산업기술대학교 게임공학과

관심분야 : 게임기획, 게임시나리오



정 성 택 (S. T. Chung)

KAIST 전기 및 전자공학과 졸업  
KAIST 정보 및 통신공학과 석사  
KAIST 전기 및 전자공학과 박사  
한국산업기술대학교 컴퓨터공학과 부교수

관심분야 : 영상처리, 헬스케어, 기능성 Game Design



송 장 섭 (Jang-Seop Song)

한국의국어대학교 물리학과 졸업  
한국의국어대학교 물리학과 석사  
(주)비술 개발팀장

관심분야 : 영상계측, 초고속영상해석, 3차원 영상분석,  
광응용 계측기술