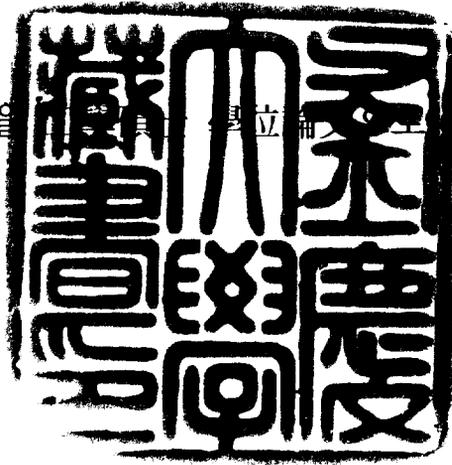


工學碩士 學位論文

웹 기반의 가상환경에 의한 능력별  
영어 가상학교 수업 모형

指導教授 辛 奉 炆

이 論文을 爲한 學位論文 提出함



2002년 2월

釜慶大學校 産業大學院

컴퓨터공학과

朴 淳 喆

이 論文을 朴淳喆의 工學碩士  
學位論文으로 認准함

2001년 12월

主 審 工學博士 鄭 木 童 

委 員 工學博士 徐 庚 龍 

委 員 工學博士 辛 奉 炆 

# 목 차

Abstract .....	vi
제 1 장 서 론 .....	1
제 2 장 VRML 이란 무엇인가? .....	3
2.1 VRML .....	3
2.2 VRML 문법소개 .....	7
2.2.1 노드와 필드 .....	7
2.2.2 DEF와 USE .....	8
2.2.3 Anchor 노드 .....	9
2.2.4 VRML로 구현한 아바타 문법 코드 .....	9
제 3 장 VRML에 의한 Shared Space .....	12
3.1 EAI에 대해서 .....	12
3.2 자바 EAI와 VRML을 이용한 3D 동기화 시스템 .....	18
3.2.1 서버와 클라이언트 동기화 .....	18
3.2.2 아바타 생성 .....	19
제 4 장 웹 기반의 가상환경에 의한 능력별 영어 가상학교 수업모형제안	31
4.1 제안 개요 .....	31
4.2 제안 배경 .....	32
4.3 3D 가상환경의 시스템 흐름도 .....	33

제 5 장 영어 가상학교의 구현방법 .....	35
5.1 A Classroom 설계방법과 교수방법 .....	35
5.2 B Classroom 설계방법과 교수방법 .....	39
5.3 평가교실의 설계방법과 교수자료 .....	43
제 6 장 실험 및 분석 .....	47
제 7 장 결론 및 향후계획 .....	50
참고문헌 .....	52
참고 웹사이트 .....	53
부록 .....	54

## 그림 목 차

그림 2.1 Virtock Technology. Inc 사의 Spazz3D의 authoring tool .....	5
그림 2.2 Cosmo Software사의 CosmoWorld 2.0 .....	6
그림 2.3 자바 애플릿, EAI, VRML을 활용한 3D공간 .....	7
그림 3.1 VRML 모델정의 .....	14
그림 3.2 HTML 파일정의 .....	14
그림 3.3 EAI에 필요한 자바클래스 .....	15
그림 3.4 VRML모델에 접근을 위한 준비 .....	15
그림 3.5 VRML모델에 노드 생성/삭제 .....	15
그림 3.6 VRML모델에 문자열 생성 .....	16
그림 3.7 VRML모델에 이벤트 보내기 .....	17
그림 3.8 VRML모델에 이벤트 받기 .....	17
그림 3.9 초기화면설정 .....	22
그림 3.10 로그인 장면 .....	22
그림 3.11 로그인해서 아바타가 생성되고 실시간으로 3D채팅 하는 장면 ..	23
그림 3.12 mainview 초기화면 .....	28
그림 3.13 mainview에서 로그인 화면 .....	29
그림 3.14 mainview에서 실시간 채팅화면 .....	29
그림 3.15 아바타가 이동하면서 실시간 채팅하는 장면 .....	30
그림 4.1 제안 배경 .....	32

그림 4.2 시스템 흐름도 .....	33
그림 5.1 A Classroom으로 처음 로그인한 화면 .....	36
그림 5.2 수업시작화면 .....	36
그림 5.3 A Classroom의 screen 노드코드 .....	37
그림 5.4 동화상이 진행되고 소리가 나오는 화면 .....	38
그림 5.5 3D 가상공간에서 채팅 하면서 수업하는 장면 .....	39
그림 5.6 B Classroom의 Listen box 코드부분 .....	40
그림 5.7 B Classroom에 로그인 해서 Listen box가 적재되어있는 화면 ...	41
그림 5.8 교사가 학생들에게 직접 명령을 주어 수업하고 있는 화면 .....	42
그림 5.9 학생이 교사에게 응답하며 수업하는 화면 .....	42
그림 5.10 평가교실로 이동했을 때의 화면 .....	43
그림 5.11 평가교실에서 학생들과 평가하기 전에 채팅 하는 장면 .....	44
그림 5.12 3D 가상환경에서 2D HTML화면으로 이동한 화면 .....	45
그림 5.13 3D 공간을 2D URL로 연결시킨 VRML 코드 .....	46

## 표 목 차

표 6.1 3D 가상화면 설계(디자인)은 어떠한가? .....	47
표 6.2 가상학교 수업모형에 대한 흥미는? .....	48
표 6.3 3D 채팅에서 교사와 학생의 상호작용에 대한 이해도는? .....	48
표 6.4 3D 가상학교수업의 학업 성취도는? .....	49
표 6.5 3D 가상학교 프로그램이 원활하게 수행하는가? .....	49

# **Design of Cyber English School Based on Ability Grouping Using Web-based Virtual Environment**

Soon-Cheol Park

Department of Computer Engineering

Graduate School of Industry

Pukyeong National University

## **Abstract**

Since the mid 1990s, the Internet-based platforms and interfaces have had an effect upon every area of our lives. And today the internet is no longer a new high-tech. In the past, every passive information was transmitted in one direction from providers to passive users, but today's users who want the information actively search for it through Web pages and participate in Web-based communities. In this thesis, We tested a new model of multi-user 3D virtual environments based on ability classification using Web-based interfaces. By making use of this technology, many internet-related interfaces have been developed and

applied. And also many applications based on 3D virtual environment have been released and known through the internet. Since the year 1990, many computer scientists have studied the method of 3D Virtual Environment. This was designed and developed using VRML(Virtual Reality Modeling Language) and JAVA EAI. Especially, We expected that this would made the students actively take part in the class through Avatar, an important feature in the 3D Virtual Environments. We also tried to transfer the effective educational materials and provide cyber education using 3D Virtual Environment. In the future, we'll expect to give birth to the X3D standard high technology.

# 제 1 장 서 론

1990년 중반부터 전 세계적으로 불기 시작한 Internet의 열풍은 사회 모든 분야에 엄청난 파급효과를 가져왔다. 인터넷이란 말 자체가 최근에 가장 접하기 쉬운 용어 중에 하나가 되었을 정도이다. 과거에는 정보를 일방적으로 정보공급자 중심의 획일적인 전달 매체를 통해 이루어 졌지만 오늘날은 정보사용자가 직접 찾아서 자기 자신의 정보로 적극적으로 찾아가는 그런 시대로 발전되었다. 그 중에서도 과거 전자우편, 뉴스그룹, 혹은 텔넷, ftp등을 사용하던 문자위주의 정보전달에서 WWW(World Wide Web)을 이용한 문서전달, 동화상의 재생, 음악이나 그래픽 등과 같은 멀티미디어 매체의 활용은 정말로 혁신적인 것이었다. 하지만 이러한 모든 것은 평면적이고 단편적인 문서에 지나지 않았다. 이러한 평면적이고 2차원적인 것에 만족하지 않고 인터넷에서 3차원공간을 활용한 방법의 연구가 시작되었으며 이에 1995년에 탄생한 것이 VRML(Virtual Reality Modeling Language)이다[1]. 한마디로 말하면 VRML은 인터넷을 통해 3차원 그래픽을 표현하기 위한 세계 표준언어이다. 단순히 3차원을 통한 그림을 보여주는 것에서 만족하지 않고 다양한 형태의 표현이 가능하며 여러 가지 기능을 수행할 수 있는 형태의 언어이다. 즉 VRML이란 인터넷과 가상현실의 만남이라고 해도 과언이 아닐 정도로 막강하고 미래의 인터넷의 대표적인 표준언어이다. 그러나 여기에 한가지 문제점이 아직 존재하고 있다. HTML언어와 같이 HTTP 프로토콜이 존재하는 것이 아니라 Netscape 나 Internet Explore와 같은 웹브라우저 속에 Cosmo Player나 Community Client와 같은 응용 프로그램이 plugin형태로 웹브라우저가 호출하면 연결되어 실행된다는 데 있다. 특히 VRML은 독립적인 3D공간에서 호출되고 실행되는 특성 때문에 초기에는 실용화하고 구현하는 데 상당한 어려움이 있다. 이러한 문제점을

해결하기 위한 것으로 VRML97의 탄생과 자바 애플릿을 이용한 EAI(External Authoring Interface)[2]의 개발이다. 이러한 독립적인 공간에서의 시스템구현의 단점을 보완한 자바 애플릿을 이용한 EAI는 지금까지 주춧돌이었던 3D공간 시스템 구현에 더욱 힘을 더해주는 결과를 가져다주었다. 3D 공간 시스템 구현에 있어서 가장 눈에 띄는 것은 다름 아닌 아바타[4]의 생성과 구현이라고 할 수 있다.

본 논문에서는 3D공간에서의 다자간 상호작용을 통한 가상 영어학교를 구현해 보았다. 이러한 가상 영어학교를 제안한 배경에는 21세기의 학교현장에서는 정보교육의 필요성이 강조되고 있으며 학습자의 첨단 멀티미디어 활용욕구가 어느 때보다 증대되고 있다. 이에 따라서 컴퓨터를 활용한 신속한 교육정보의 제공과 능력별 수업프로그램을 활용한 교육활동의 효율성이 재고될 필요가 있다. 특히 3D 가상환경을 통한 다자간 상호작용은 이제 가상학교에서는 반드시 필요한 요소이다.

마지막으로 본 논문의 개략적인 구성에 대해서 설명하겠다. 제2장에서는 VRML의 소개와 배경과 개요 그리고 지정하는 노드 외에 자체적인 확장노드를 추가한 여러 가지 응용 프로그램과 자바 애플릿의 EAI를 이용한 사용자 인터페이스 구현 방법과 정의에 대해서 설명하고자한다. 3장에서는 VRML에 의한 Shared space에 대하여 4장에서는 가상 영어학교의 제안 개요 및 배경에 대해서 학교현장에서 이루어지는 문제점과 향후 발전계획을 원론적인 측면에서 재고하고 수정할 수 있는 방안을 소개하고, 5장에서는 자바 프로그램과 VRML의 코딩방법을 이용해서 가상 영어학교의 구현방법을 구체적으로 설명하겠다. 6장에서는 가상 영어학교의 실험하고 분석한 결과를 소개하고 마지막으로 7장에서는 문제점과 향후 발전계획에 대해서 구체적인 방안을 모색해 보겠다.

## 제 2장 VRML이란 무엇인가?

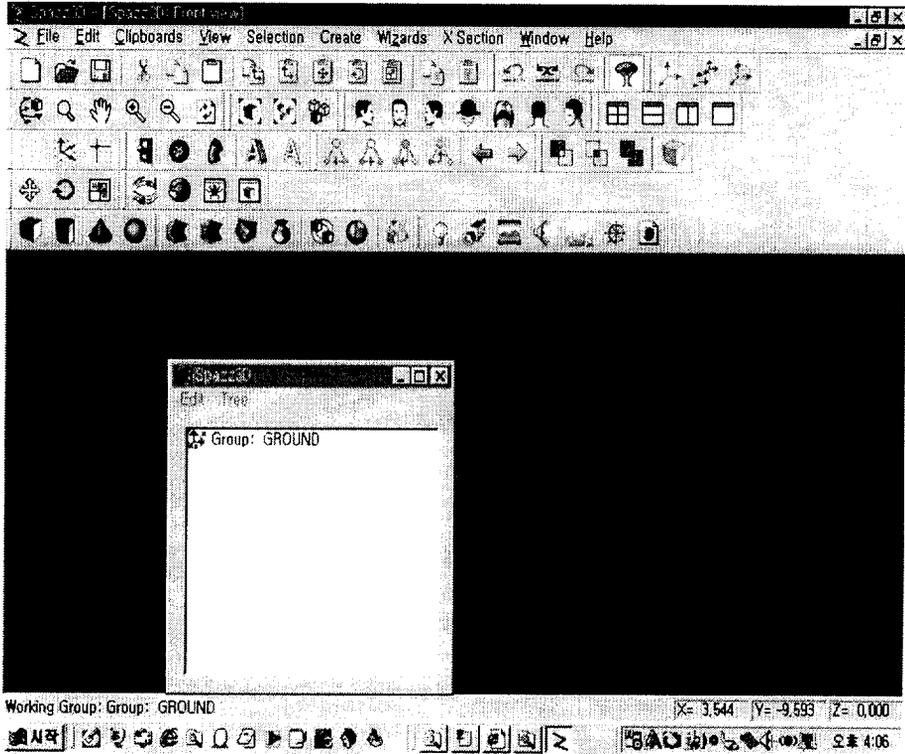
1994년 유럽 웹 컨퍼런스에서 팀 버너리가 웹 표준의 필요성에 대해 주장한 것이 사실상 VRML의 시초이다[9]. 그는 HTML과 유사한 개념의 VRML(Virtual Reality Markup Language)이라는 단어를 만들었다. 따라서 VRML이 만들어지기도 전에 새로운 사상의 표현을 위한 작업이 수행되었고 드디어 VRML2이 탄생하게 된다. 그리고 좀더 많은 정밀한 작업으로 지금 현재는 VRML97까지 세상에 나오게 되었다. 본 논문의 모든 VRML코딩 작업은 VRML97로 이루어져 있다.

### 2.1 VRML(Virtual Reality Modeling Language)

VRML은 다양한 가상 공간을 저작 가능하게 하며 단일 사용자와의 효과적인 상호작용을 지원한다. 차츰 웹사이트를 VRML로 구축한 사례들이 늘어나고 있으며 VRML을 이용한 게임도 볼 수 있다. 그러나, 상호작용 가상 공간 시스템을 구축하기 위해서는 현재까지의 VRML97 사양으로는 아직까지 구현하기가 어렵다. 이런 이유로 상호작용 가상공간시스템을 구현한 제품들은 VRML을 고려하지 않고 독립적인 응용프로그램의 형태로 제품을 개발하거나 고려하더라도 독자적으로 확장된 VRML 사양을 이용하여 개발하고 있다.

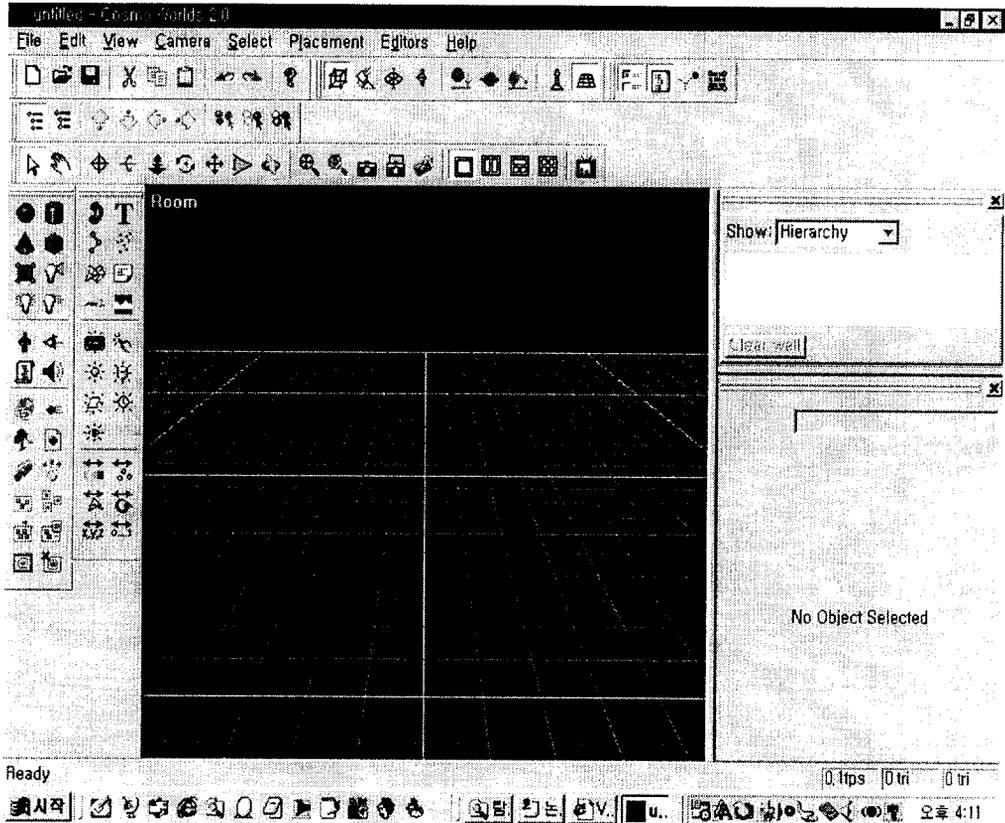
일반적인 VRML 브라우저는 VRML97 사양만을 충족하므로 가상 공간 다중 사용자 환경 시스템의 브라우저가 아닌 이상 동기화 된 시스템은 기대하기 힘들다. 그래서 다중 사용자 환경을 지원하는 시스템들은 특정 응용프로그램을 제공하고 있으며 각각 독자적인 동기화 기술들을 가지고 있어서 시스템간의 호환성이 부족하다. 반면에 VRML의 사양을 만족하면서 확장된 형태의 브라우저 제품도 있어서 매우 다양하다.

확장형 브라우저는 표준 VRML에서 지정하는 노드 외에 자체적인 확장 노드를 추가한 것으로 동기화 시스템 목적에 맞게 추가하였다. 이런 제품들을 구분해서 프로그램과 브라우저 별로 살펴보면 먼저 Circle of Fire Studios사의 Active Worlds2.0과 Chaco사의 Pueblo나 VRScout등의 응용프로그램 형태가 있으며, Blaxxun Interactive사의 Community Server3.0과 Community Clients VRML97 사양을 충분히 지원하는 가장 보편화된 시스템이다. 그리고 Cosmo Software 사의 CosmoWorld2.0(30일 trial version), Virtock Technology, inc 사의 Spazz3D 의 저작도구로서 많이 사용하고 있으며 Trial Version이지만 모든 기능을 30일간 사용할 수 있는 아주 훌륭하고 보편화된 도구들이다. [그림2.1]은 Virtock Technology, Inc 사의 Spazz3D 의 저작도구로서 3D공간을 저작할 때 아주 편리하고 사용자 중심의 인터페이스를 제공한다. 특히 이 저작도구의 가장 큰 특징은 바로 아바타를 저작하고 그 아바타의 행동을 추가 시켜주는 애니메이션기법에 탁월한 성능을 발휘하는 응용프로그램으로 3D 공간을 저작하는데 도움을 준다.



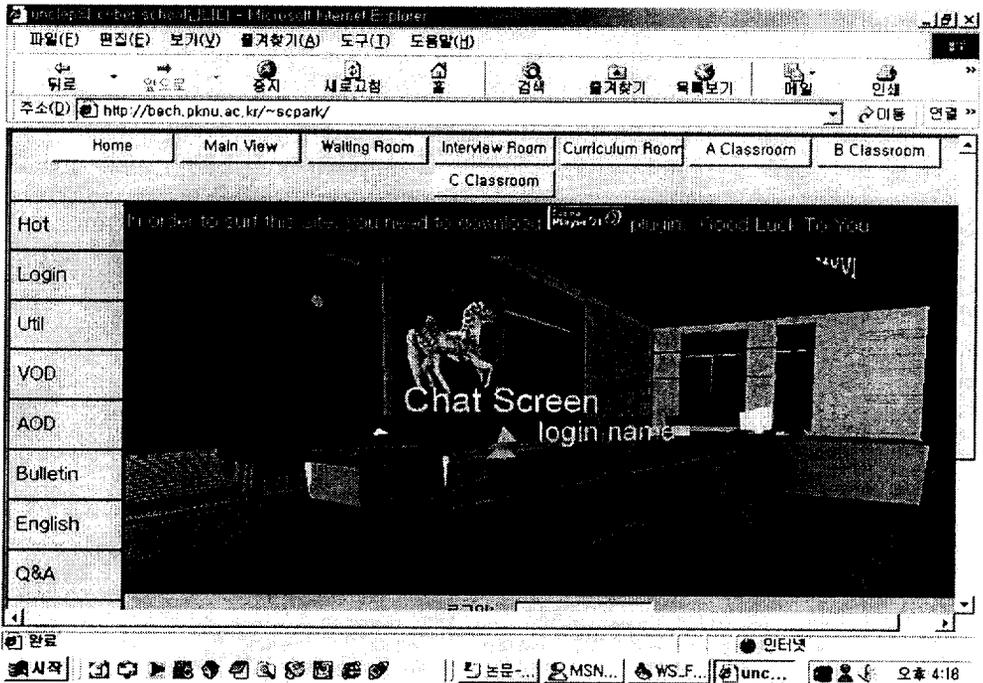
[그림2.1] Virtock Technology, inc 사의 Spazz3D 의 저작도구

그리고 [그림2.2]는 Cosmo Software 사의 CosmoWorld2.0으로서 3D공간의 배경이나 건물 등을 저작하고 외부세계와 서로 상호 작용할 수 있도록 많은 도구 바를 제공해 주고 있다.



[그림2.2] Cosmo Software 사의 CosmoWorld2.0

이러한 저작도구들을 이용해서 가상공간을 아주 쉽고 편하게 제작할 수 있을 뿐만 아니라 필요에 따라 VRML의 프로그램 코드부분을 적절히 수정해서 사용자의 편의에 맞게 디자인해서 멋진 3D공간을 저작할 수 있다. 그 예로 [그림2.3]을 보면 전체 3D공간의 모습과 아바타가 들어간 부분들은 모두 저작도구를 이용해서 저작하고 또한 사용자의 요구에 의해서 여러 부분을 직접 코딩해서 만든 화면이다.



[그림.2.3] 자바애플릿, EAI, VRML을 활용한 3D공간

## 2.2 VRML(Virtual Reality Modeling Language) 문법 소개

### 2.2.1 노드와 필드

VRML 노드는 어떤 기능을 수행하는 부분에 해당한다. 노드의 명칭은 기본적인 기능을 암시해준다. 예를 들어, transform이나 cone등이 있다. 각각의 노드는 일련의 필드를 포함하며, 이것은 기능 요소 값을 포함하고 있다. 예를 들어 cone 노드의 필드는 콘의 높이와 바닥원의 반지름을 정의해준다. 이 두 값은 콘이 반드시 필요로 하는 값이다. 반면에 transform 노드는 cone 노드와는 달리 형상을 표현해주는 노드가 아니라 그룹핑 노드이다. 따라서, 자식 필드라는 필드 중의 하나는 일련의 다른 노드이다. 아래의 코드는 cone 노드와 transform 노드의 정의이다.

```

Cone {
    field SFFloat bottomRadius      1
    field SFFloat height            2
    field SFBool side               TRUE
    field SFBool bottom            TRUE
}

```

```

Transform {
    eventIn      MFNode      addChildren
    eventIn      MFNode      removeChildren
    exposedField SFVec3f     center      0 0 0
    exposedField MFNode      children    [ ]
    exposedField SFRotation  rotation   0 0 1  0
    exposedField SFVec3f     scale       1 1 1
    exposedField SFRotation  scaleOrientation 0 0 1  0
    exposedField SFVec3f     translation 0 0 0
    field        SFVec3f     bboxCenter  0 0 0
    field        SFVec3f     bboxSize   -1 -1 -1
}

```

### 2.2.2 DEF와 USE

만약 사용자가 사각형의 모양으로 배열된 파란 16개의 파란색 구체를 그린다면, 16개의 Transform 노드가 필요할 것이다. 그러나 VRML에서는 DEF/USE를 사용하여 간단히 묘사할 수 있다. 이 두개의 변형 키워드를 이용하면 간단한 노드로 구현이 가능하다. 아래의 코드가 바로 DEF/USE를 사용해서 구현한 코드이다.

```

Shape {
    appearance DEF Red appearance {
        material Material { diffuseColor 1 0 0 }
    }
}

```

```

        geometry Box {}
    }
Shape {
    appearance USE Red
    geometry Cylinder {}
}

```

### 2.2.3 Anchor 노드

HTML 웹 페이지에서, 밝게 반전된 모양, 또는 이미지를 클릭하여 그에 연결된 새로운 페이지로 도달할 수 있다. 하이퍼링크는 HTML 페이지나 이미지, VRML 세계와 같은 다른 매체(media)의 URL이다. Anchor 노드를 가지고 VRML 세계에 하이퍼링크를 둘 수 있다. 다음은 노드의 정의이다.

```

Anchor {
    eventIn          MFNode          addChildren
    eventIn          MFNode          removeChildren
    exposedField     MFNode          Children      []
    exposedField     SFString        description   " "
    exposedField     MFString        parameter    []
    exposedField     MFString        url          []
    field            SFVec3f         bboxcenter   0 0 0
    field            SFVec3f         bboxSize     -1 -1 -1
}

```

### 2.2.4 VRML로 구현한 아바타 코드 문법

```

#VRML V2.0 utf8
WorldInfo {
    title "avatar"
}
//viewpoint 정의
DEF dad_ViewPoint1 Transform {

```

```

    children DEF ViewPoint1 Viewpoint {
        position      0 0 0
        fieldOfView   0.79
        jump          TRUE
        description   "ViewPoint1"
    }
}

// DEF로 정의된 부분을 EAI의 getNode() 함수에서 호출
DEF dad_Wizard_Group Transform {
    children DEF Wizard_Group Group {
        children DEF dad_main Transform {

            children DEF main Group {
                children [
                    DEF dad_torso Transform {
                        children DEF torso Group {
                            children [
                                DEF dad_torso0 Transform {
                                    children DEF torso0 Shape {
                                        appearance Appearance {
                                            material DEF Red Material {
                                                ambientIntensity 0.2
                                                diffuseColor      1 0 0
                                                specularColor    0 0 0
                                                emissiveColor     0 0 0
                                                shininess        0.2
                                            }
                                        }
                                    }
                                }
                            ]
                        }
                    }
                ]
            }
        }
    }
}

//TimeSensor를 이용해서 제어함
DEF Wizard TimeSensor {
    cycleInterval 1
    startTime     0
    stopTime      0
}

```

```

}

//방향값을 제어
DEF Wizard_rot0 OrientationInterpolator {
    key    [ 0, 0.25, 0.5, 0.75, 1 ]
    keyValue    [ 0 0 1  0,
                  0 1 0  1.571,
                  0 1 0  3.142,
                  0 1 0  4.712,
                  0 1 0  6.28319 ]
}

//ROUTE 문을 이용해서 제어함
ROUTE Wizard_rot0.value_changed TO dad_main.set_translation
ROUTE Wizard_rot0.value_changed TO dad_main.set_rotation
ROUTE Wizard.fraction_changed TO Wizard_rot0.set_fraction
ROUTE Script_spazz_init.triggerStart TO Script_Wizard.startAt0
ROUTE Wizard.isActive TO Script_Wizard.activated
ROUTE Wizard.cycleTime TO Script_Wizard.cycle
ROUTE Sensor1_0.touchTime TO Script_Sound1.startAt0
ROUTE spazz_init.cycleTime TO Script_spazz_init.cycle

```

## 제 3 장 VRML에 의한 Shared Space

지금까지 설명한 것은 모두가 VRML이란 독립적인 공간에서 이루어지는 인터페이스에 대한 구현방법과 모델링에 대한 내용이다. 이제부터는 이러한 독립적인 3D 공간에서 외부 세계와의 통신을 하고 공유할 수 있는 수단으로서 어떤 저작도구를 이용하고 또한 무슨 프로그램을 이용해서 다중 사용자들이 원하는 목적을 수행할 것인가에 대한 구체적인 방법들을 제시하고 구현하고자 한다. 먼저 VRML의 모든 노드들의 이벤트를 처리하고 제어하는 데 가장 적합한 언어는 다름 아닌 자바이다. 주로 웹에서 다중 사용자들이 동시에 채팅을 하거나 화상전달, 소리전달과 같은 멀티미디어를 활용할 수 있는 언어로서는 자바 언어 중에서도 주로 애플릿을 이용해서 구현해야 한다. 그리고 VRML과 자바 애플릿의 서로 다른 독립 공간을 매개시켜 주는 것이 EAI인데 이 인터페이스에 대한 개략적인 내용도 소개하겠다.

### 3.1 EAI(External Authoring Interface)

3D환경을 동기화 시키고 실시간으로 아바타를 이용한 3D 채팅을 구현하기 위해서는 우선 자바애플릿과 VRML장면 노드들의 이벤트에 접근하거나 제어할 수 있도록 하는 EAI의 개념이해가 우선 필요하다. EAI라는 것은 External Authoring Interface의 약자로 일종의 자바의 API와 유사한 Interface이다. 이것을 이용해서 각각 독립되어있는 자바애플릿과 VRML의 장면 노드들을 서로 유기적으로 연결시켜주고 동기화 시킴으로써 3D환경에서의 어려웠던 Internet의 웹 상에서의 다중 사용자환경을 구축시켜준다. 그리고 일반적인 웹 브라우저를 사용하면서 VRML 세계를 동기화 하려면 먼저 스크립트 노드에 대한 이해가 필요하다. 스크립트 노드는 VRML에서 동적으로 애니메이션이 가능하게 하는 노드로서 스크립트 노드가 없다면 사

용자의 요구에 즉각적으로 응답하는 가상 공간의 구현은 어렵다. 그러나, 스크립트 노드는 VRML세계 파일에 포함돼서 코딩되거나 따로 URL을 통해서 내장되는 특성 때문에 비록 동적인 가상 공간의 구현은 가능하나 독립적인 세계밖에는 구성할 수 없다. 이를 보완하기 위해서 나온 것이 EAI으로서 스크립트 노드와 구별하자면 특정 VRML 세계 파일과 관련 없이 외부에서 VRML 세계 파일을 저작하게 하는 인터페이스이다.

EAI는 자바 애플릿에 포함되어서 같은 웹 페이지에 포함된 VRML 브라우저와의 동적인 인터페이스를 구축하고 여기에 브라우저에 적재된 특정 VRML 세계 파일에 접근할 수 있는 기능을 제공함으로써 스크립트가 수행하는 모든 기능을 똑같이 지원한다. EAI를 통해서 VRML 세계의 동기화가 가능한 이유는 단지 애플릿으로 구현된다는 것인데, 이것이 바로 동기화를 가능하게 하는 직접적인 원인이 된다.

애플릿으로 구현된 EAI는 VRML세계 파일을 동적으로 접근함과 동시에 애플릿이 가지는 네트워크 기능으로 원격 지의 가상 공간 동기화 서버 프로그램과 통신한다. 애플릿은 새로운 사용자가 접속할 때마다 그들을 대신하는 아바타를 생성하는 기능을 가지고 있으며 위의 기능들은 이벤트를 전달함으로써 수행한다. 단일 사용자가 발생시킨 이벤트를 변환 알고리즘을 거쳐 특정 메시지를 생성하여 서버로 보내고 이를 다시 서버에 접속한 나머지 사용자에게 전송하여 원격 지의 사용자에게도 동일한 이벤트를 부여할 수 있게 되는 것이다. 다음의 그림들은 EAI의 중요한 함수와 동기화에 필요한 내용들이다.

```
#VRML V2.0 utf8 :
DEF ROOT Transform {
  translation 0 0 0
  Shape {
    appearance Appearance {
      material Material { }
    }
    geometry Box {
      size 3.0 3.0 3.0
    }
  }
}
```

[그림3.1] VRML 모델 정의

VRML의 특징은 Java EAI가 VRML 공간에 접근할 때는 반드시 DEF문으로 정의된 노드만 접근 가능하다. 첫 줄에 VRML 헤더가 있는 것을 볼 수 있다. 그 다음부터 존재하는 것은 모두 노드들이다.

```
<EMBED SRC="mainroom.wrl">
<APPLET CODE="MainroomChatClient.class" mayscript>
</APPLET>
```

[그림3.2] HTML File 정의

[그림3.2]에 있는 코딩 부분은 VRML세계파일을 HTML에 적재하는 방법으로 먼저 EMBED를 정의해서 VRML세계파일을 적재하고 자바 애플릿과 EAI을 통해서 VRML세계파일을 외부세계와 통신하게 한다.

```
import vrml.external.Browser;
import vrml.external.Node;
import vrml.external.field.*;
import vrml.external.exception.*;
```

[그림3.3] EAI에 필요한 자바 클래스

[그림3.3]은 자바애플릿을 작성할 때 import 시켜주는 4개의 클래스를 나타내주고 있다. 각 클래스마다 VRML세계 파일에 접근할 수 있는 많은 함수가 있는 데 그 중에 중요한 몇 가지 함수만 소개하고자 한다

먼저 [그림3.4]은 VRML세계 파일과 애플릿이 HTML의 같은 프레임에 있을 때와 다른 프레임에 있을 때의 코딩 방법을 설명하고 있다.

```
* VRML Model과 Applet이 같은 Frame에 있을 때
    Browser browser = Browser.getBrowser(this);
* VRML Model과 Applet이 다른 Frame에 있을 때
    Browser browser = Browser.getBrowser(this, "FrameName", 0);
    Node root = browser.getNode("ROOT");
```

[그림3.4] VRML 모델에 접근을 위한 준비

```
Node[] shape;
EventInMFNode addChildren;
EventInMFNode removeChildren;
addChildren = (EventInMFNode) root.getEventIn("addChildren");
removeChildren=(EventInMFNode)root.getEventIn("removeChildren");
shape = browser.createVrmlFromString("Shape (geometry Box { } )");
String [] url1 = "man1.wrl";
browser.createVrmlFromURL(url1,root,"addChildren");
addChildren.setValue(shape);
removeChildren.setValue(shape);
```

[그림3.5] VRML 모델에 노드 생성/삭제

VRML 모델에 노드를 생성하거나 삭제하려면 [그림3.5]처럼 그룹노드의 addChildren, removeChildren 필드를 사용한다. 생성하거나 삭제할 노드에 대한 함수정의는 createVrmlFromString나 createVrmlFromURL 함수로 문자열을 노드로 바꿔주거나 아니면 URL을 VRML세계에 접근시켜준다. 장면 자체를 모두 다른 노드로 생성하거나 삭제에 대한 정의는 loadURL을 이용해서 동기화 시켜준다.

```

//로그인 텍스트 문자열
EventInMFString s_text=null;
String t [] =new String[1];
//텍스트 노드 입력 노드 생성
Node n_text = browser.getNode("TextLogin");
s_text = (EventInMFString) n_text.getEventIn("set_string");
//VRML의 text Node 접근 이벤트
t[0]= line;
s_text.setValue(t);

```

[그림3.6] VRML 모델에 문자열 생성

[그림3.6]처럼 상호작용 가상공간 시스템에서 3D 채팅 시 평면적인 2D 화면에 텍스트 문자열이 나타나는 것이 아니라 3D공간에 문자열이 나타나도록 하기 위해서는 EventInMFString을 이용해서 s\_text 객체를 만들어서 VRML 월드파일의 DEF함수인 TextLogin에 접근하면 멋진 3D공간에 채팅 화면이 실시간으로 이루어지게 된다.

```

translation=(EventInSFVec3f)
avatar.getEventIn("set_translation");
rotation = (EventInSFRotation)
avatar.getEventIn("set_rotation");
float t_xyz[] = new float[3];
float r_xyz[] = new float[4];
translation.setValue(t_xyz);
rotation.setValue(r_xyz);

```

[그림3.7] VRML 모델에 이벤트 보내기

[그림3.7]에서처럼 아바타를 이동시킬 때는 EventInSFVec3f 클래스를 이용하고 회전시키고자 할 때는 EventInSFRotation 클래스를 이용해서 아바타나 자식 노드를 제어할 수 있다.

```

outTranslation = (EventOutSFVec3f) avatar.getEventOut("translation");
outTranslation.advise(this, new Integer(1));
outRotation = (EventOutSFRotation) avatar.getEventOut("rotation");
outRotation.advise(this, new Integer(2));
//VRML장면에서 변화된 반환값을 호출
public void callback(EventOut who, double when, Object which) {
    Integer whichNum = (Integer) which;
    if(whichNum.intValue() == 1)
        float[] val = outTranslation.getValue();
    if(whichNum.intValue() == 2)
        float[] val = outRotation.getValue();
}

```

[그림3.8] VRML 모델에서 이벤트 받기

[그림3.8]에서처럼 선언시 EventOutObserver로 선언한 뒤 이동 이벤트를

받기 위해서는 EventOutSFVec3f 클래스를 이용하고 회전 이벤트를 받기 위해서는 EventOutSFRotation 클래스를 이용하는 데 이러한 VRML 세계파일에서 이루어지는 행동의 방식을 감시하기 위해서 callback() 함수를 이용해서 변화된 반환 값을 계속해서 추적할 수 있다.

## 3.2 자바 EAI와 VRML을 이용한 3D 동기화 시스템

### 3.2.1 서버와 클라이언트의 동기화

처음 사용자가 가상 공간에 접속하게 되면 서버에서는 VRML세계 파일을 클라이언트에게 전송한다. 이때는 단지 HTTP전송 프로토콜에 의해서 비동기로 전송되며 동적으로 갱신이 필요 없는 가상 공간 외형 정보와 행위를 기술한 정보들이 아바타 정보를 제외하고서는 이 시점에 적재된다[2]. 동시에 VRML 브라우저와 통신할 애플릿이 서버로부터 적재되며 이 애플릿은 가상 공간 동기화 서버와 연결되어 이벤트의 실시간 처리를 담당한다. 애플릿은 곧바로 VRML 브라우저와 상호작용 하기 위한 초기화 작업을 수행한다. 아래의 코드는 서버와의 연결을 설정하는 작업이다.

```
try{
    String host = getParameter("host");
    if (host==null)
        host = getCodeBase ().getHost();
    String port = getParameter("port");
    if (port==null)
        port = "9830";
    output.append("Connecting to " + host + ":" + port + "...");
    Socket s =new Socket(host, Integer.parseInt(port));
    I = newDataInputStream(new BufferedInputStream(s.getInputStream()));
    o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
}
```

### 3.2.2 아바타의 생성

시스템에서 가장 핵심사항은 아바타의 생성과 행동이다. 아바타는 사용자의 분신으로서 가상 공간을 사용자를 대신해서 탐험하게 된다. 여러 사용자간에 일어나는 Interaction은 모두가 이 아바타의 생성의 매개로 EAI를 통해 이벤트를 처리할 수 있다. 그리고 채팅공간의 구성은 일반적인 2D의 텍스트 필드에 출력될 뿐만 아니라 3D VRML의 노드 공간에도 모든 클라이언트들이 동시에 동기화 되도록 구현한다. 이들 아바타의 생성과 3D 가상 공간에 텍스트가 출력되도록 하기 위해서는 다음과 같은 내용으로 코딩한다.

```
Browser browser;
```

```
Node root;
```

```
//creatVrmlFromURL을 위한 배열 생성
```

```
String[] url1;
```

```
String[] url2;
```

```
Node[] shape;
```

```
EventInMFNode addChildren;
```

```
EventInMFNode removeChildren;
```

```
    <중략>
```

```
init()함수부분
```

```
browser = (Browser)Browser.getBrowser(this);
```

```
// Root 화면 등록 & 자식노드 등록과 해제
```

```
root = browser.getNode("Mainview");
```

```
url1 = new String[1];
```

```
url1[0] = "man1.wrl";
```

```
url2 = new String[1];
```

```
url2[0] = "man2.wrl";
```

```
addChildren = (EventInMFNode)root.getEventIn("addChildren");
```

```
removeChildren =(EventInMFNode)root.getEventIn("removeChildren");
```

```

root.getEventOut("children_changed").advise(this,new Integer(1));
browser.createVrmlFromURL(url1,root,"addChildren");
//텍스트 로그인노드 입력 노드 생성
Node n_text = browser.getNode("TextLogin");
s_text = (EventInMFString) n_text.getEventIn("set_string");
//텍스트 채팅노드 입력 노드 생성
Node chat_n_text = browser.getNode("TextChat");
chat_text = (EventInMFString) chat_n_text.getEventIn("set_string");
<중략>
try{
    String host = getParameter("host");
    if (host==null)
        host = getCodeBase ().getHost();
    String port = getParameter("port");
    if (port==null)
        port = "9830";
    output.append ("Connecting to " + host + ":" + port + "...");
    Socket s =new Socket(host, Integer.parseInt(port));
    i=newDataInputStream(new BufferedInputStream(s.getInputStream()));
    o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
<중략>
    while(!stop){
        String line = i.readUTF();
        if (lineCount > 50){
            output.setText(lastLine+"\n");
            lineCount =0;
        }
        output.append( line +"\n");
        lastLine = line;
        lineCount ++;
        //로그인
        //VRML의 text Node 접근 이벤트

```

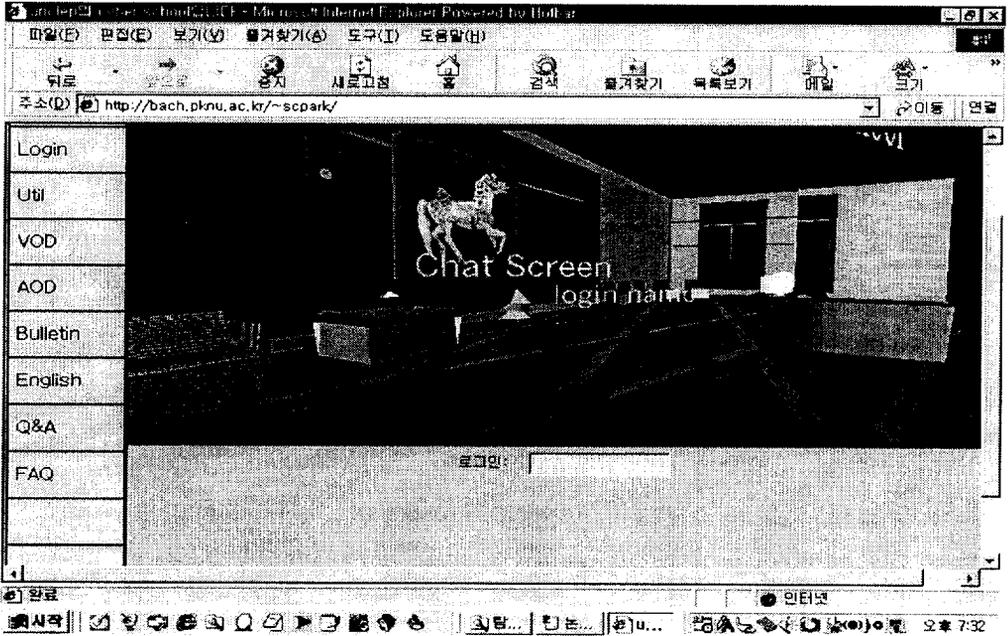
```

t[0]= line;
s_text.setValue(t);
//채팅
//VRML의 text Node 접근 이벤트
chat[0]= line;
chat_text.setValue(chat);
root.getEventOut("children_changed").advise(this,new
Integer(1));
//아바타 생성(모든 client에 생김)
Browser.createVrmlFromURL(url1,root,"addChildren");
}

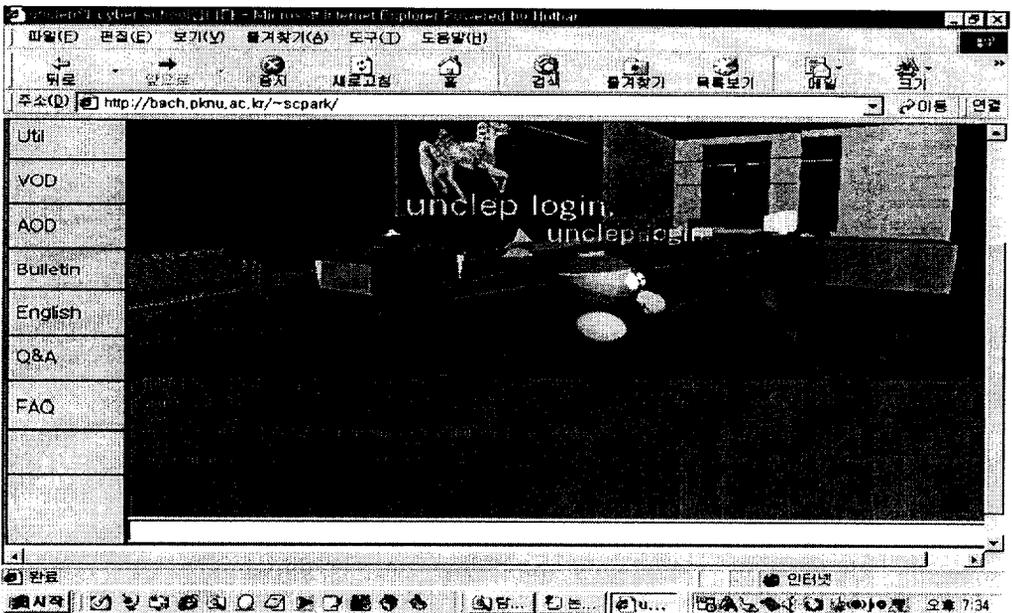
```

<중략> 나머지코드는 뒷면 부록참고

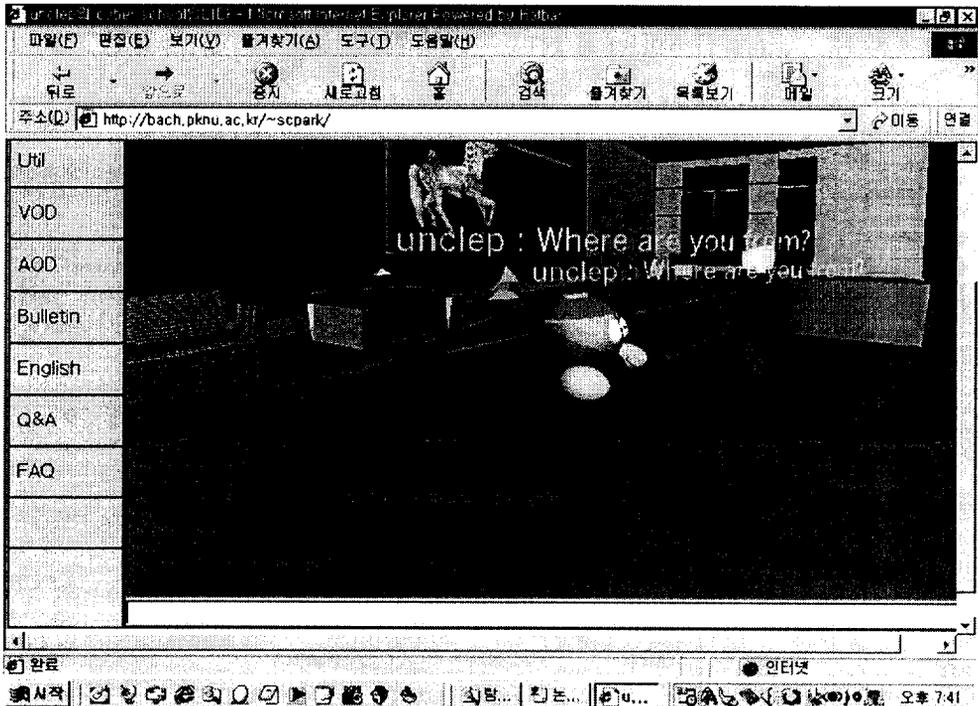
[그림3.9], [그림3.10], [그림3.11]에서처럼 위의 코드를 수행하게 되면 웹 상에서 자바애플릿과 VRML 장면들이 동기화 되어 보여지게된다. 우선 소켓이 연결되고 HTTP 전송 프로토콜에 의해서 VRML 세계파일을 클라이언트에게 전송한다. 이때는 단지 HTTP 전송 프로토콜에 의해 비동기로 전송되며 가상 외형 정보와 동기화에 필요한 정보들이 아바타를 제외하고는 모두 적재되며 사용자가 로그인을 통해서 접속하게 되면 그때부터 아바타가 생성되고 동기화가 드디어 이루어진다. 그리고 VRML 장면 노드와 애플릿이 동기화를 위한 초기화 작업이 수행되고 서버와의 연결을 하게 된다. 그리고 접속하는 모든 클라이언트들에게 모든 화면이 동시에 동기화되어 신원이 확인되고 채팅 할 때마다 아바타의 상단부분에 3D 채팅화면이 나타나서 텍스트 필드가 아닌 그 즉시 VRML 3D 가상환경 속에서 재미있는 채팅을 할 수 있도록 설계하였다.



[그림3.9] 초기화면 설정



[그림3.10] 로그인 장면



[그림3.11] 로그인해서 아바타가 생성되고 실시간으로 3D채팅을 하는 장면

그리고 아바타의 이동과 회전 이벤트등 실시간으로 진행되고 있는 내용은 가상학교의 핵심적인 기술로서 3D공간의 여러 장면으로 이동하고 이동한 장면에서 한 사용자가 다른 사용자와 필요에 따라서 서로 상호작용이 이루어져야 한다. 그리고 여러 사용자가 3D 가상학교에 접속하게 되면 그 사용자의 신원이 확인되어야한다. 그래서 본 시스템에서는 아바타의 상단에 로그인이 되면 확인할 수 있도록 ID가 텍스트로 나오게 구현하였다. 아래의 코드가 ID를 VRML 장면과 애플릿을 서로 동기화하고 아바타의 이동과 회전을 하는 작업이다.

아래의 EAI 코드부분은 아바타의 이동과 회전을 동기화 시키고 keyDown() 이벤트나 mouseDown() 이벤트를 이용해서 키보드나 마우스로 사용자가 조정할 수 있게 편의성을 제공해주고 있다. 이벤트를 받아서 아

래의 translation.setValue(t\_xyz) 와 rotation.setValue(r\_xyz) 값에 의해 아바타가 조정되고 VRML세계파일에서 여러 장소로 항해할 수 있게 해준다.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import java.util.*;
// VRML 노드와의 메시지 교환을 위해서..
import vrml.external.Browser;
import vrml.external.Node;
import vrml.external.field.*;
import vrml.external.exception.*;

public class ControlChatClient extends Applet implements
Runnable,ActionListener,EventOutObserver {
<중략>
    Node root = browser.getNode("Mainview");
    Node avatar = browser.getNode("dad_main_50");
//위치, 방향 VRML장면으로 이벤트입력
    translation = (EventInSFVec3f)avatar.getEventIn("set_translation");
    rotation = (EventInSFRotation)
    avatar.getEventIn("set_rotation");
//위치,방향 VRML장면으로 이벤트출력
    outTranslation = (EventOutSFVec3f)avatar.getEventOut("translation");
    outTranslation.advise(this, new Integer(1));
    outRotation = (EventOutSFRotation)avatar.getEventOut("rotation");
    outRotation.advise(this, new Integer(2));
<중략>
//VRML장면에서 변화된 반환값을 호출
```

```

public void callback(EventOut who, double when, Object which) {
    Integer whichNum = (Integer) which;
    if(whichNum.intValue() == 1)
        float[] val = outTranslation.getValue();
    if(whichNum.intValue() == 2)
        float[] val = outRotation.getValue();
}

```

<중략>

//아바타 이동 회전 이벤트

```

public boolean keyDown(Event event, int key) {
    switch (key) {
    case Event.UP: //up
        t_xyz[2]-=1.0f;
        break;
    case Event.ENTER: //up,left
        t_xyz[0]-=1.0f;
        t_xyz[2]-=1.0f;
        break;
    case Event.PGDN: //up,right
        t_xyz[0]+=1.0f;
        t_xyz[2]-=1.0f;
        break;
    case Event.DOWN: //down
        t_xyz[2]+=1.0f;
        break;
    case Event.PGUP: //down,left
        t_xyz[0]-=1.0f;
        t_xyz[2]+=1.0f;
        break;
    case Event.END: //down,right
        t_xyz[0]+=1.0f;
        t_xyz[2]+=1.0f;
    }
}

```

```

        break;
    case Event.LEFT:        //left
        t_xyz[0]-=1.0f;
        break;
    case Event.RIGHT:      // right
        t_xyz[0]+=1.0f;
        break;
    case Event.DELETE:     //left rotation
        r_xyz[3]+=1.0f;
        break;
    case Event.HOME://right rotation
        r_xyz[3]-=1.0f;
        break;
    )
    translation.setValue(t_xyz);
    rotation.setValue(r_xyz);
    return true;
    }
//마우스로 아바타를 조정
public boolean mouseDown(Event event, int x, int y) {
    t_xyz[0] = (x-50)/30.f;
    t_xyz[2] = -(y-50)/30.f;
    translation.setValue(t_xyz);
    return true;
}

```

아래의 코드는 VRML 각 장면 노드들에 대한 대략적인 코드로 이루어져 있다. 특히 자바 애플릿에서 VRML장면의 이벤트를 수행하기 위해서 명심해야 할 부분은 VRML 장면의 DEF로 정의된 노드 외에는 접근이 불가능하다. 즉 DEF로 정의된 Mainview노드나 dad\_main\_50노드로 접근이

가능하다.

```
#VRML V2.0 utf8
```

```
Background {...}
```

```
DEF Mainview Transform (
```

```
  children      Viewpoint {
```

```
    position    1.5 2 0
```

```
  }
```

```
<중략> ...
```

```
DEF dad_Wizard_Group Transform {
```

```
  children DEF Wizard_Group Group {
```

```
    children DEF dad_main_50 Transform {
```

```
      children DEF main Group {
```

```
        children [ ...
```

```
          geometry Sphere {
```

```
            radius      1
```

```
          }
```

```
        }
```

```
      bboxCenter      0 0 0
```

```
      bboxSize 2.06 2.06 2.06
```

```
      translation     0 0 0.073
```

```
      rotation 1 0 0 0.212
```

```
      scale 1.1 0.766 1
```

```
    }
```

```
    geometry DEF TextLogin Text {
```

```
      fontStyle FontStyle {
```

```
        size 0.5
```

```
        family "SANS"
```

```
      }
```

```
      string "Please Login"
```

```
    }
```

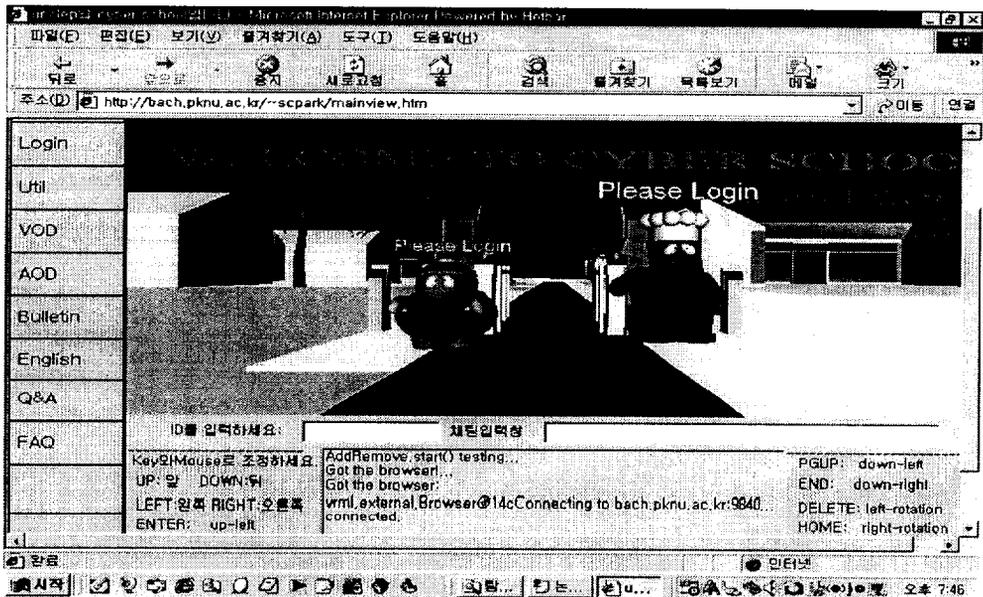
```

}
translation      -1.2 1.7 0
}

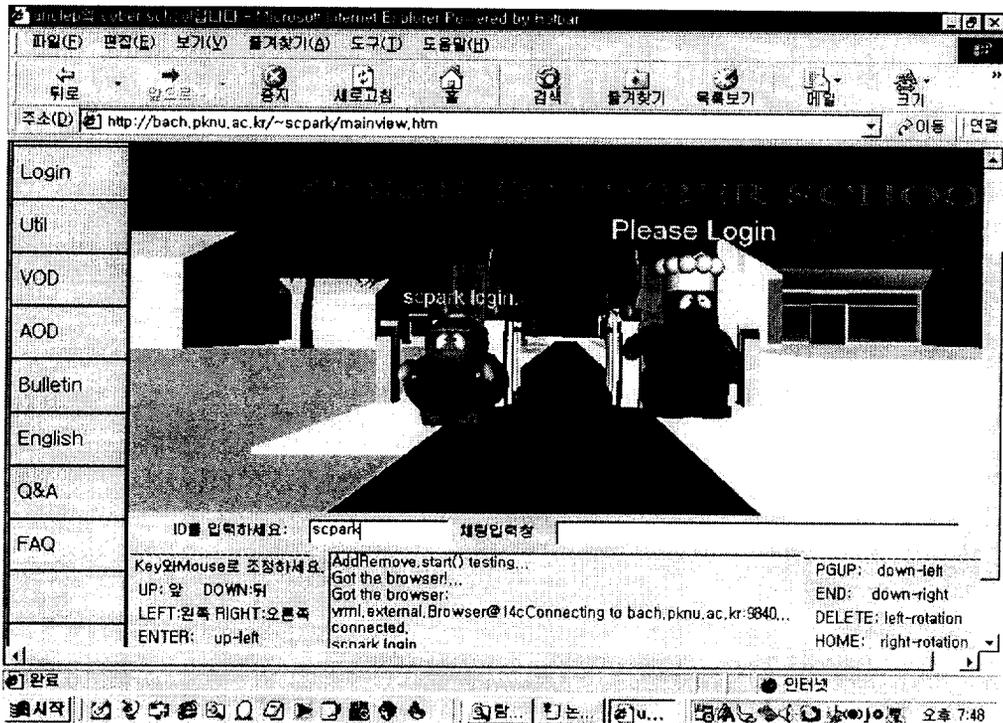
```

<중략>

[그림3.12], [그림3.13], [그림3.14], [그림3.15]이 위의 코드를 VRML 장면과 애플릿 장면을 동기화해서 실행한 화면이다. 이 장면은 가상학교의 전체 보기에 해당하며 가상학교의 6개 Classroom과 배경화면이 등록되어 있고 학생들이 전체 Classroom을 향해하고자 할 때 실행하는 화면이다. 로그인을 통해서 자신의 ID를 확인한 후 여러 학생들과 대화를 하기도 하고 직접 몰입해서 여러 Classroom을 한 눈에 볼 수 있도록 디자인하였다.



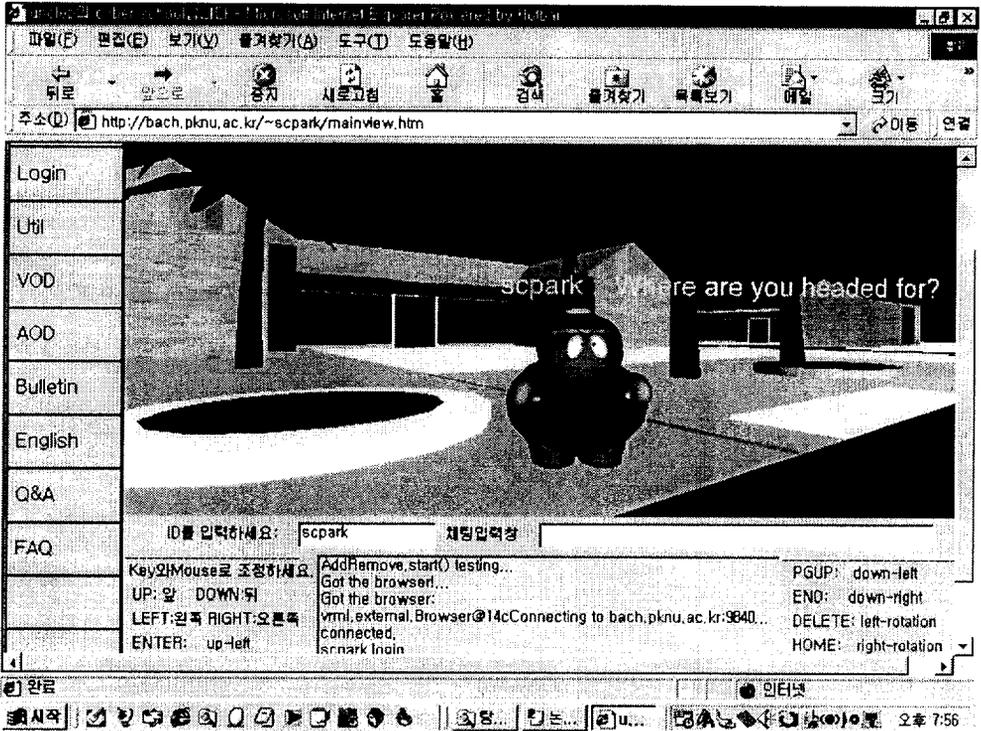
[그림3.12] mainview 초기화면



[그림3.13] mainview에서의 로그인 화면



[그림3.14] mainview에서의 실시간 채팅화면



[그림3.15] 아바타가 이동하면서 실시간 채팅하는 장면

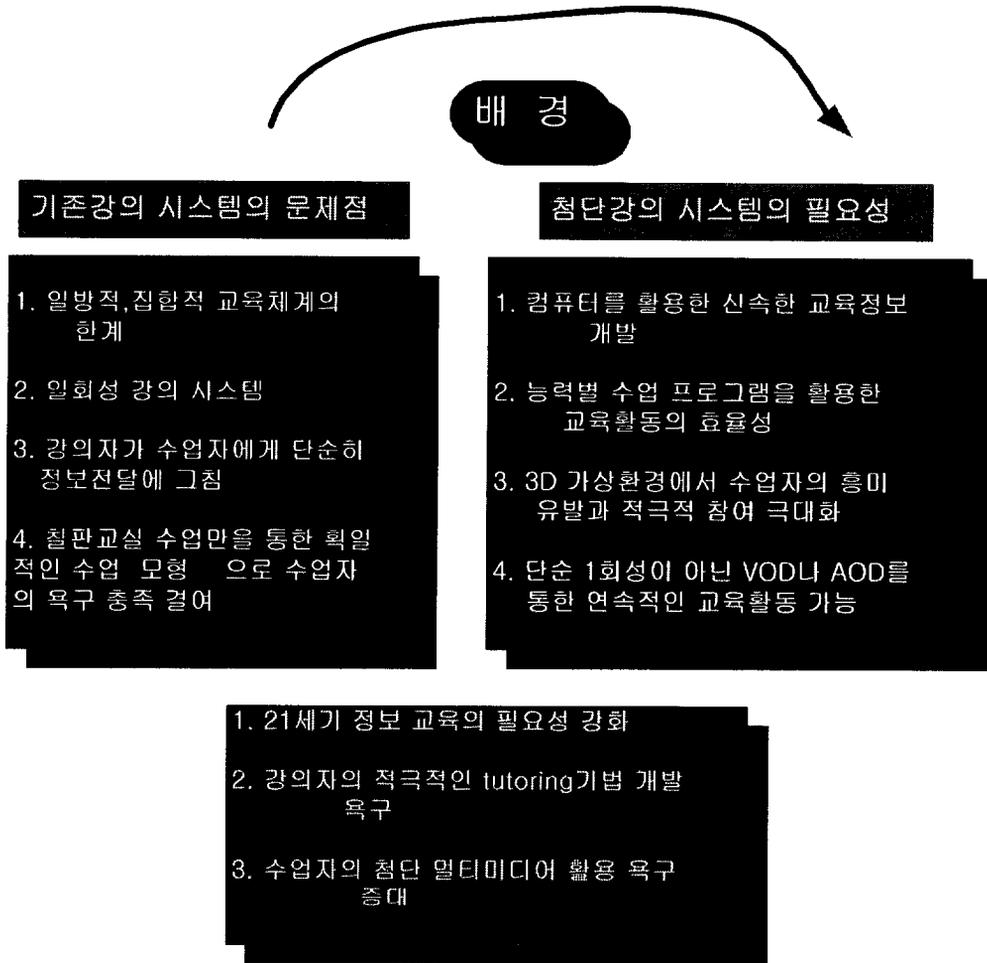
# 제 4 장 웹 기반의 가상환경에 의한 능력별 영어 가상학교 수업 모형 제안

## 4.1 제안 개요

3장에서 보였던 아바타의 동기화 기법, Sever/Client를 구축한 3D 가상 환경에서 실시간 채팅, 그리고 저작도구를 이용한 다양한 교실 3D가상 환경 제공을 통해서 현재의 교육현장에 적용하면 보다 효율적이고 흥미로운 가상 원격 교육을 실시할 것으로 생각된다. 지금까지 교육현장에서 획일적이고 일방적인 수업형태에서 벗어나 학생들의 흥미 유발과 실시간 수업 시 긴장감의 해소 혹은 교실환경에서의 수업에서 벗어나 사용자가 어느 장소에서나 학교의 서버에 접속해서 수업을 받을 수 있는 시스템의 운영은 앞으로 학교 현장에서 반드시 필요하다. 다음은 3D 가상 환경을 통한 수업의 필요성에 대한 개요이다.

- ▣ 효율적인 수업 진행과 다 매체 활용을 통한 강의 시스템 개발.
- ▣ 3D 공간을 활용한 학생들의 수업(강의)에 대한 흥미 유발.
- ▣ 실시간으로 전개되는 Tutoring Material을 통해 수업(강의) 및 교육효율의 극대화.
- ▣ 학생들의 능력에 맞는 집단을 세분화해서 수업의 효율성을 높여줌.
- ▣ 학교 현장에 적절한 프로그램 작성 및 운영(학교현장에 첨단 H/W 설비 부족).
- ▣ 사용자의 편리성을 고려한 환경 시스템 구축.

## 4.2 제안 배경



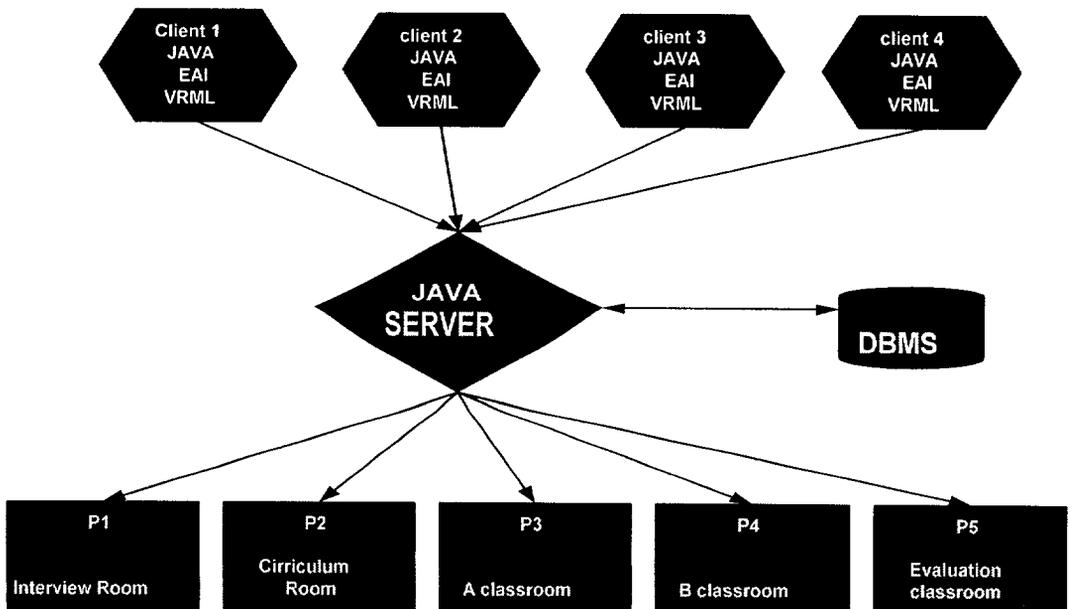
[그림4.1] 제안 배경

현재의 교육적인 현실을 고려할 때 가장 중요한 것이 바로 학생들의 흥미 유발과 교육적인 효율의 극대화를 통한 다양한 다 매체 활용의 강의 시스템 개발이 우선적으로 선행되어야 한다고 생각한다. 가장 큰 교육의 문제점은 학교현장의 업무 중심의 획일적인 교육시스템으로 인하여 일방적, 집합적 교육체계의 한계점을 무시할 수 없다는 것이다. 그리고 강의자의 강의가

단순히 정보전달이나 단편적인 지식전달에 그치거나 아니면 획일적인 지식의 단편화를 초래하고 있다. 그리고 모든 수업자의 획일적인 수업 모형으로 수업자의 욕구 충족이 결여되어 학습의 효과가 엄청나게 감소하는 결과를 초래하게 된다.

그리고 3D 가상 환경을 활용한 강의 시스템이 필요한 배경에는 21세기의 정보 교육의 강화 필요성이 대두되고 있고 소극적인 강의자의 자료개발의 자세에서 벗어나 적극적인 강의자의 강의 기법을 개발하려고 하는 움직임이 지금 교육현장에서 활발하게 이루어지고 있다. 이러한 강의자의 첨단 멀티미디어를 활용한 수업의 개발을 위한 환경변화에 발맞추어 본 논문에서 이러한 3D 가상환경을 직접 구현해서 학교현장에 적용해보고자 제안한다.

### 4.3 3D 가상환경의 시스템 흐름도



[그림4.2] 시스템 흐름도

[그림4.2]에서 처럼 먼저 Java applet이나 application, Java EAI, VRML를 이용해서 아바타를 구현한다. 사용자가 3D Interview Classroom로 이동해서 아바타를 이용한 실시간 영어 채팅을 통해서 Question & Answer방식으로 대화를 한 뒤 강의자가 실시간으로 그 학생의 등급을 정해 준다.

DB를 서버에 구현한다. 대상 수업자들은 본인의 등급을 확인한 후 Main 화면에서 Curriculum 신청 장소로 이동한 후 수업자가 수업하고자하는 Curriculum을 클릭해서 이내용을 서버에 저장되도록 하고 수업자의 최종 신청 단위수와 과목내용의 결과물을 볼 수 있도록 출력한다.

각 등급별로 나누어진 수업자는 본인에게 정해진 A, B Classroom으로 각각 이동해서 본인의 과목을 확인한 후 각 교실로 이동해서 강의자가 서버에 직접 구축한 내용을 3D 가상 환경 교실에서 수업을 실시하면 된다.

수업을 실시한 후 학생들은 3D 가상학교에서 구현되어 있는 평가실로 이동해서 본인의 성취도를 확인하기 위해서 그 등급에 맞는 교육내용을 평가하고 확인하는 작업을 한다.

그리고 대상 수업자들의 보충학습을 위해서 VOD, AOD 활용공간을 구축한다. 직접 수업자가 VOD교실과 AOD교실로 이동해서 수업한 내용을 복습한다.

서버에 구축된 프로그램 내용은 매일 강의자가 갱신시켜준다.

## 제 5장 영어 가상학교의 구현방법

### 5.1 A Classroom 설계 방법과 교수방법

A Classroom의 교수방법은 우선 상위 학생의 10명으로 구성되어있는 교실로서 먼저 로그인을 통해서 10명의 학생이 동시에 접속한다. 모든 학생들의 ID는 로그인을 통해서 교사들은 참여학생을 확인할 수 있다. 그리고 바로 영어 채팅을 통해서 학생들의 주의를 끌 수가 있으며 만약에 학생들이 로그인 만하고 다른 행동을 하고 있으면 교사들이 채팅을 통해서 충분히 확인할 수 있기 때문에 원격적으로 이루어지는 영어 교실에 대한 문제점을 해결할 수 있다고 본다. 그리고 학습방법은 학생들을 확인한 한 뒤 바로 학생들에게 영어로 대화를 나눈 뒤 교사의 통제하에 수업을 진행할 수 있다. 먼저 학생들에게 교사가 준비한 영어로 만 구성된 1~5분 정도의 스크린 영어 교재를 3D안에서 직접 보고 그것에 대한 내용을 교사가 질문과 응답 방식으로 수업을 진행하고 학생들의 답은 채팅창을 통한 공간에서 실시간으로 영어로 답하거나 아니면 3D공간에 있는 정답 box를 클릭 함으로써 학생들의 이해도를 측정할 수 있다. 특히 실시간으로 진행되는 3D환경 수업이기 때문에 교사의 주도하에 집중적으로 지도하지 않으면 산만해지고 효과가 없을 수도 있다. 그래서 교사가 수업 중간에 채팅을 통하여 학생들의 존재를 확인하고 대화하는 시간을 많이 가져야만 이 가상학교의 효과를 증폭시킬 수 있다. 그리고 아바타를 통해서 교육활동이 이루어지기 때문에 학생들이 수업 시 영어 채팅으로 인한 긴장감을 어느 정도 해소하고 좀더 재미있고 효과적인 영어 수업을 할 수 있다. 디자인은 VRML언어를 이용해서 구현하면 된다. 아래의 [그림5.1]과 [그림5.2]가 실제 구현한 A Classroom 3D 가상 수업의 실제이다.



[그림5.1] A Classroom으로 처음 로그인한 화면



[그림5.2] 수업 시작 화면

```

[VRML코드]
Transform(
  children [
    DEF TS TouchSensor { }
      Shape {...}
      texture DEF SCREEN1 MovieTexture {
        url      "mpg/001.mpg"
        loop      FALSE
        startTime 0
        stopTime  0
      }
    geometry DEF _90 IndexedFace { ... }
    Sound {
      source USE SCREEN1
      maxBack 1000
      maxFront 1000
    }
    ROUTE TS.touchTime TO SCREEN1.startTime
  ]
}

```

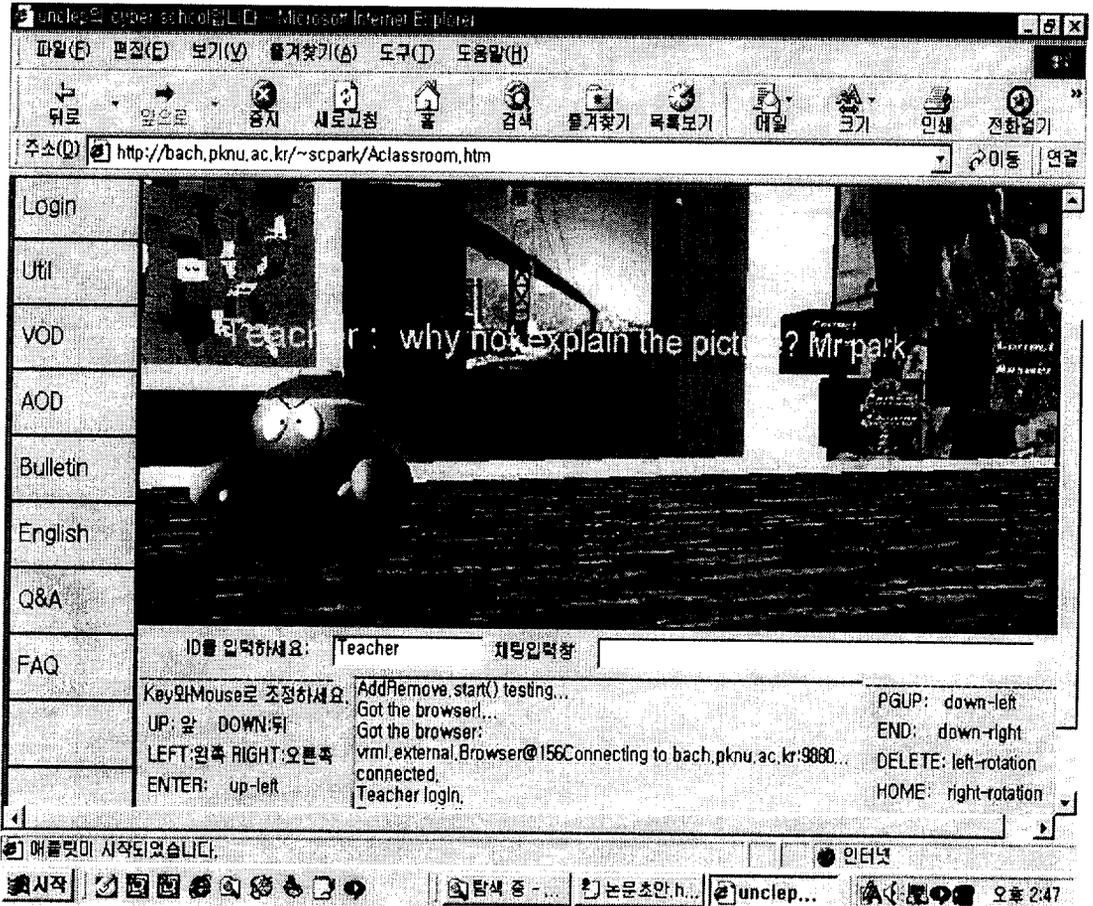
[그림5.3] A Classroom의 screen 노드 코드

[그림5.3] 의 VRML 코드부분은 A Classroom의 실제 수업 모형에서 교실 벽에 위치해 있는 스크린을 제어하는 방법으로 먼저 TouchSensor 노드를 이용해서 사용자가 VRML 세계 파일에 접근 시 스크린에 포함되어 있는 mpg파일을 적재와 동시에 동기화 실행되지 않도록 startTime 과 stopTime을 0값을 주고 loop는 FALSE로 정의한다. 이렇게 코드를 정의하게 되면 사용자가 VRML 월드파일에 접근하더라도 사용자가 스크린을 클릭 할 때까지는 mpg 파일이 실행되지 않는다. 이렇게 구현함으로써 A Classroom의 교실 공간에 적재되어 있는 많은 MovieTexture 노드의 mpg 파일들을 사용자가 원하는 시간에 제어하고 실행할 수 있다. 그 다음의 문제는 mpg 파일이 실행될 때 사용자가 교실공간의 어느 장소에 있는가에 따라서 소리가 동기화 될 수도 있고 아닐 수도 있다. 그래서 이런 문제점

을 해결하기 위해서 Sound노드를 이용한다. Sound노드 안에 maxBack과 maxFront를 이용해서 적절한 수를 입력해주면 사용자가 서버에 접속해서 스크린을 클릭 하게 되면 3D 가상 환경 속에서 mpg파일의 동화상과 소리를 동시에 들을 수 있다. [그림5.4] 와 [그림5.5] 는 실제 수업에서 사용자가 동화상을 클릭해서 수업을 진행하고 있는 화면이다.



[그림5.4] 동화상이 진행되고 소리가 나오고 있는 화면



[그림5.5] 3D 가상공간에서 채팅하면서 수업하는 화면

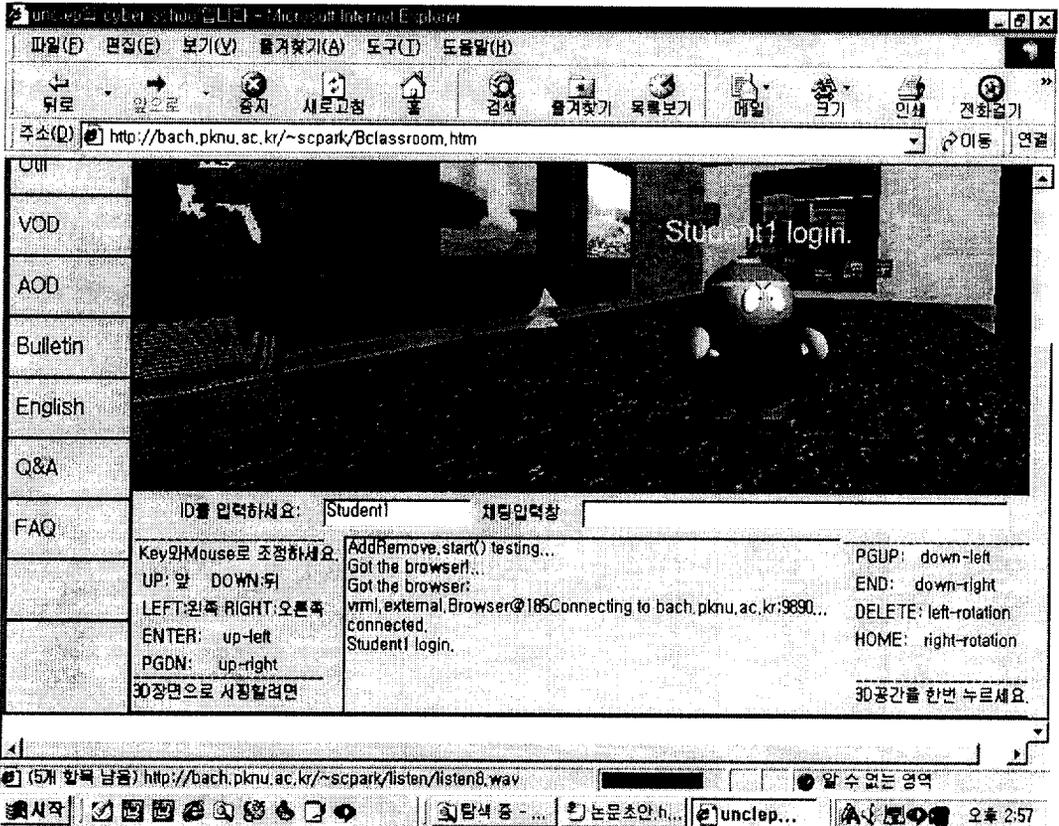
## 5.2 B Classroom 설계 방법과 교수방법

B Classroom의 교실환경은 우선 초급회화(Easy English)이기 때문에 먼저 학생들이 로그인을 통해서 교사와 학생간의 간단한 생활 영어를 구사한다. 이때 교사는 어려운 회화를 구사하는 것이 아니라 평범하고 이해하기 쉬운 문장으로만 구성된 교육내용으로 수업을 진행한다. 그리고 교사 주도하에 학생들로 하여금 Listen1~5까지의 박스를 클릭 하게 하고 학생은 교사의 주도하에 직접 자신이 클릭해서 영어회화의 듣기 내용을 듣고 난 뒤 교사가 직접 영어채팅을 통해서 정답을 알아본다. 학생들은 교사의 질문에

객관식으로 대답하기 위해서는 문제 박스아래에 있는 정답박스를 직접 클릭해서 교사가 확인할 수도 있고 아바타를 통해서 정답을 직접 영어로 표현할 수도 있게 디자인하였다. 아래의 [그림5.6]와 [그림5.7]이 그 화면이다.

```
Transform {
  children[
    DEF TS TouchSensor { }
    Shape {
      texture ImageTexture {
        repeatS TRUE
        repeatT TRUE
        url "listen5.gif")
      }
    geometry Box { }
  }
  Sound {
    source DEF LISTEN5 AudioClip {
      url "../listen/listen8.wav"
      startTime 0
      stopTime 0
      loop FALSE)
      maxFront 1000
      maxBack 1000
    }
  }
]
```

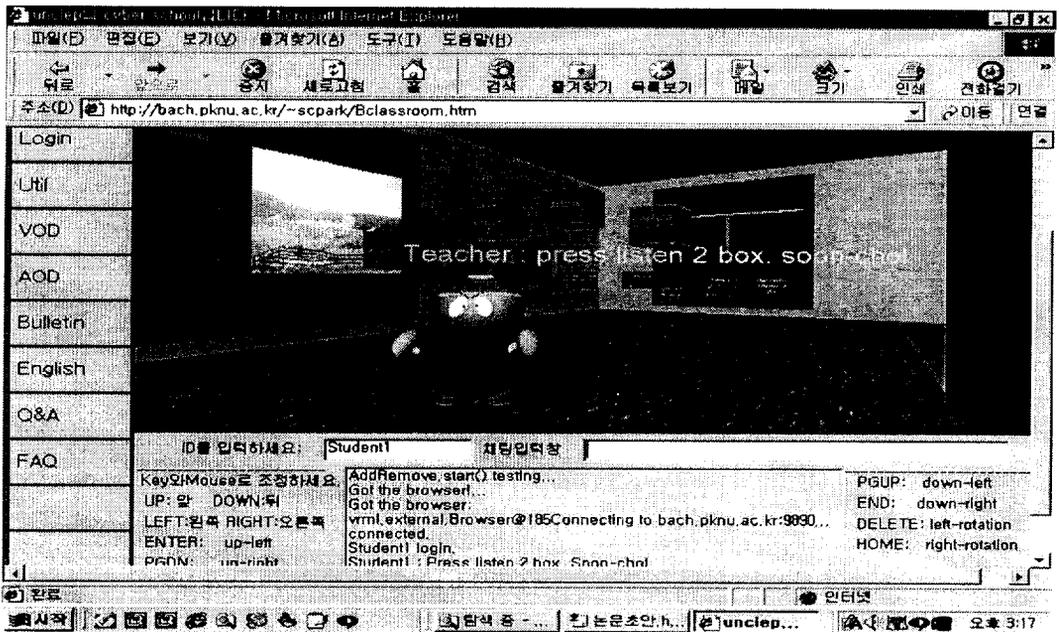
[그림5.6] B Classroom의 Listen box 코드부분



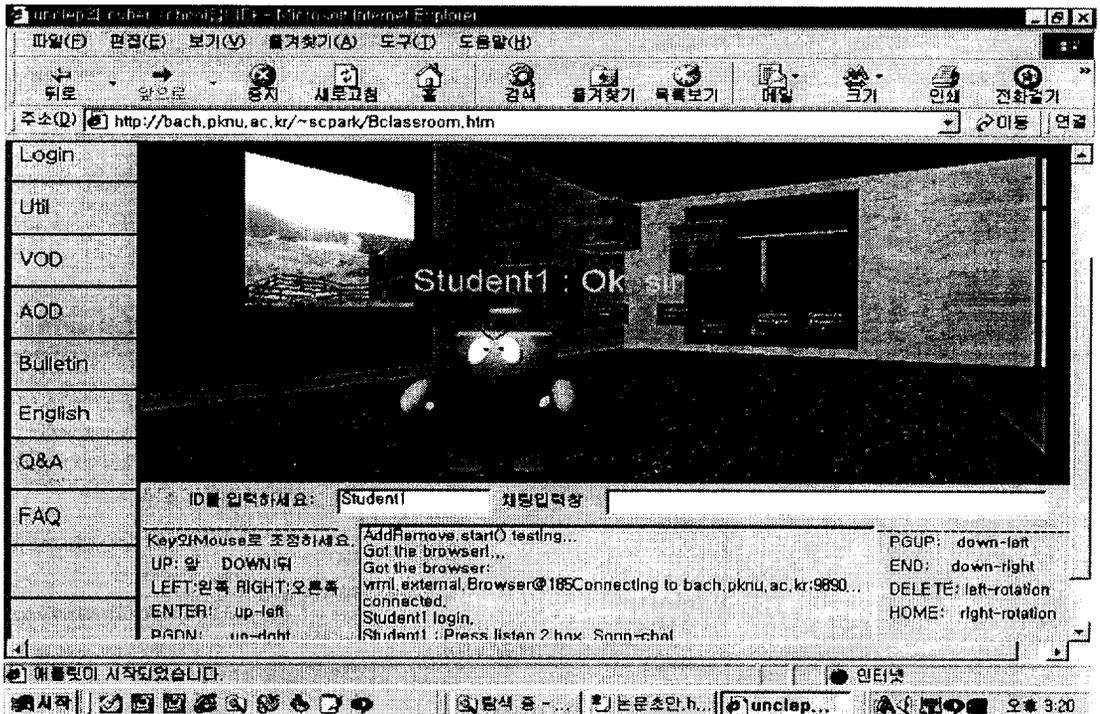
[그림5.7] B Classroom에 로그인해서 3D Listen box가 적재되어있는 화면

[그림5.6]에 있는 Listen box 코드부분은 먼저 TouchSensor 노드로 startTime과 stopTime을 제어할 수 있고 Sound 노드에 포함되어있는 AudioClip 노드를 이용해서 Listen box를 실행할 수 있도록 프로그램 한 내용이다. Listen box 속에 있는 이미지는 imageTexture노드를 호출해서 VRML세계파일에 적재가 된다.

[그림5.7]에서 처럼 student1이 3D 교실환경에 들어오면 그때부터 수업이 진행되는데 교사가 학생들에게 듣기 문제를 직접 클릭하게 해서 이해도를 측정하거나 그에 부수적인 내용을 아바타를 통해서 상호 작용하면서 수업을 진행한다. 그리고 [그림5.8] 와 [그림5.9]은 구체적으로 교사와 학생간의 수업을 진행하고 있는 화면이다.

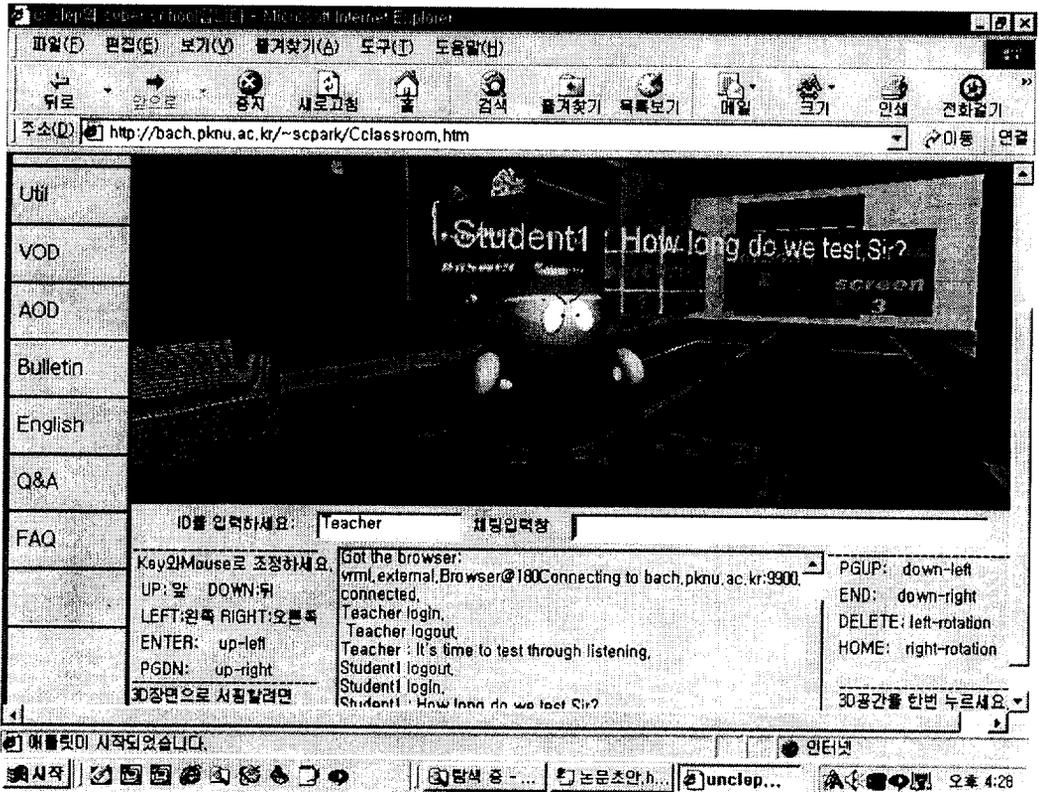


[그림5.8] 교사가 학생들에게 직접 명령을 주어 수업하는 화면



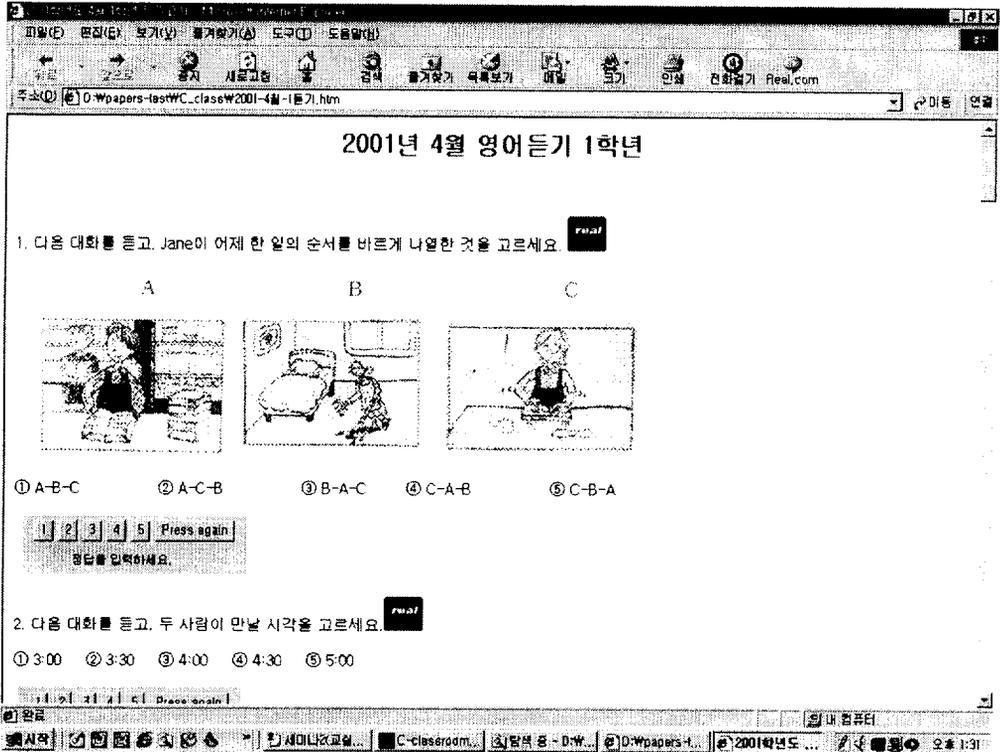
[그림5.9] 학생이 교사에게 응답하며 수업하는 화면





[그림5.11] 평가교실에서 학생들과 평가하기 전에 채팅 하는 화면

[그림5.10] 와 [그림5.11] 는 VRML 세계파일 장면에 아바타와 여러 모양의 3D 화면을 볼 수 있다. 여기에 있는 여러 모양의 3D 화면 구현은 5.1절과 5.2절에 설명되어있는 내용과 동일하다. 다만 3D화면에 나오는 개체를 클릭 하면 지금까지는 3D화면에서 모든 것이 실행되고 구현되었지만 이번의 경우에는 2D의 HTML 파일로 연결되어 동기화 된다. [그림5.12]과 [그림5.13]이 바로 구현되고 실행되는 화면과 코드부분이다.



[그림5.12] 3D 가상환경에서 2D HTML 화면으로 이동한 화면

[그림5.12]는 2D 화면으로 구성된 평범한 웹 페이지로서 학생들을 평가할 때 다양한 그림, 동화상, 소리를 들려주기 위해서는 3D 가상환경만으로는 부족하다. 그래서 학생들이 로그인하여 시험 시작 전에 일어나는 모든 활동은 [그림5.10] 와 [그림5.11] 에서처럼 3D 채팅화면을 통해서 교사와 학생들간의 Interaction이 이루어지고 실제 평가를 할 때는 2D를 통해서 평가할 수 있도록 구현하였다. 다만 정답 입력 부분만 자바 애플릿으로 구현하였다.

```

Anchor {
  children      Transform {
  children      Shape {
    appearance  Appearance {
      material   DEF _89 Material { }
      texture    ImageTexture {
        repeatS   TRUE
        repeatT   TRUE
        url       "screen1.gif"
      }
    }
  }
  geometry     Box { }
}
translation   8 1 -2.50593
}
url           "http://bach.pknu.ac.kr/scpark/user/public_html/C_class/2001-4-1.htm"
description   ""
parameter    ""
}

```

[그림5.13] 3D공간을 2D의 URL로 연결시킨 VRML코드

[그림5.13]에 있는 코드부분에 가장 특징적인 것은 Anchor 노드를 이용해서 3D 가상환경에서 2D의 웹 문서로 이동할 수 있다는 것이고, URL을 이용해서 가상학교의 서버나 아니면 다른 웹사이트의 웹 문서로 마음대로 향해할 수 있다는 것이다.

## 제 6 장 실험 및 분석

학생들 20명을 대상으로 3D 가상학교 프로그램을 실험 분석하였다. 우선 컴퓨터환경은 펜티엄I 166, 램 32메가, 2메가 그래픽카드, 그리고 인터넷환경은 교육망으로 테스트를 하였다. 그 결과 서버에서 모든 클라이언트에 적재되는 시간이 2분~3분 정도 소요가 되었고 아바타가 적재되고 모든 프로그램이 정상적으로 이루어 졌다. 하지만 수업을 통한 채팅이 시작되고 5분여 정도 지났을 때 각 클라이언트들은 인터넷과 컴퓨터의 속도 때문에 치명적인 에러를 발생시켰다. 우선 가장 큰 문제점은 채팅을 통한 메시지는 원활히 이루어 졌지만 메모리가 많이 차지하는 아바타의 이동이나 3D공간으로 향해할 때는 거의 모든 클라이언트가 에러를 발생시켰다. 그리고 컴퓨터 환경을 펜티엄 2이상, 램 64메가, 8메가 그래픽 카드를 이용했을 때는 각 클라이언트의 에러가 아주 많이 줄어들었고 어느 정도 수업도 가능했다. 다음의 표들은 20명을 상대로 느낀 점을 표본 조사한 결과들이다.

[표6.1] 3D 가상화면 설계(디자인)는 어떠한가?

학생들의 반응	백분율
그저 그렇다	0%
보통이다	20%
좋다	60%
아주 좋다	20%
합계	100%

[표6.1]에서 처럼 “3D 가상학교에 대한 디자인은 어떠한가?” 에 대한 질문에는 “좋다”라고 대답한 학생이 대다수였다. 개선점으로 좀더 단조롭게

설계해서 수행속도를 빠르게 했으면 좋겠다는 응답이 많았다.

[표6.2] 가상학교 수업모형에 대한 흥미는?

학생들의 반응	백분율
그저 그렇다	0%
약간 흥미롭다	20%
흥미롭다	60%
아주 흥미롭다	20%
합계	100%

[표6.2]에서처럼 가상학교 수업 모형에 대한 흥미도를 조사했는데 거의 대다수 학생들이 2D 환경보다는 흥미롭다고 응답을 했다.

[표6.3] 3D 채팅에서 교사와 학생의 상호작용에 대한 이해도는?

학생들의 반응	백분율
그저 그렇다	0%
약간 이해한다	20%
제법 이해한다	80%
전부다 이해한다	0%
합계	100%

[표6.3]에서 처럼 3D 가상환경 채팅에서 교사와 학생들의 상호 이해도를 조사한 결과 채팅공간에서 어느 정도 이해할 수 있다는 대답이 나왔다. 흥미로운 것은 많은 학생들이 가상채팅을 통해서 비록 영어채팅이지만 이해하는 수준이 높았다. 물론 학생들의 수준이 중상이상 정도의 성적을 가진 학생이다.

[표6.4] 3D 가상학교 수업의 학업 성취도는?

학생들의 반응	백분율
효과가 없다	0%
약간 효과가 있다	20%
효과가 있다	80%
아주 많이 효과가 있다	0%
합계	100%

[표6.4]에서처럼 학생들의 수업 성취도를 조사한 결과 학생들의 대다수가 수행속도가 빨라지면 효과가 있다는 응답이었다.

[표6.5] 3D 가상학교 프로그램이 원활하게 수행되는가?

학생들의 반응	백분율
원활하지 못하다	100%
그저 그렇다	0%
원활하다	0%
아주 원활하다	0%
합계	100%

[표6.5]에서처럼 가상학교의 프로그램 수행속도와 원활성에 대한 질문은 모든 학생들이 컴퓨터의 환경과 인터넷의 속도가 원활하지 못하다는 지적이 나왔다.

이런 점 외에 기타 개선점으로는 현재는 채팅을 통한 학습 채팅 사이트 수준이지만 주변 컴퓨터 환경이나 프로그램의 환경이 좋아지면 아주 흥미로울 것 같다는 의견도 있었다.

## 제 7 장 결론 및 향후계획

본 논문은 열악한 현재의 교육여건에서 어떻게 하면 학생들이 영어를 학습하는 데 기존의 획일적이고 칠판위주의 단편적인 수업 방식에서 벗어나 첨단 멀티미디어 매체를 통해서 학생들의 흥미를 유발하고 적극적인 참여를 유도함으로써 학생들의 학업 성취도를 극대화하기 위한 방안이 그 목적이었다. 즉 컴퓨터를 활용한 신속한 교육정보를 개발하고 단순 1회성이 아닌 연속적인 교육활동을 통해서 학생들의 욕구를 충족시키고 앞으로 21세기의 나아가야 할 교수학습방향을 제시하기 위해 노력했다. 가상환경 내에 다중사용자들이 의사교환 및 정보의 전달을 극대화할 수 있도록 하기 위해 아바타를 통한 3D 가상 채팅을 구현했으며 학생들의 흥미유발과 적극적 참여를 유도하기 위해 6개의 3D 가상공간과 가상학교 전체를 향해하고 볼 수 있도록 구성하였다. 그러나 이러한 다중사용자를 지원하는 가상학교의 문제점은 현재 학교현장에 설치되어있는 초고속인터넷 망이나 각 클라이언트들의 플랫폼들이 이러한 3D 가상 공간을 활용한 가상학교를 수용할 만큼의 속도나 환경을 만족시켜 주지 못한다는 것이다. 다만 3D를 활용한 3D 가상채팅 만큼은 실제 현장에서 효과가 있었고 그러한 채팅을 통해 학생들이 거부감을 가질 수 있는 영어 채팅을 어느 정도 흥미롭게 실시가 가능했다. 그리고 VRML97 사양을 가지고 3D를 저작하고 다른 여러 가지의 프로그램을 연결시키고 구현하기에는 아직 현실적인 어려움이 많았다. 특히 문제는 독립적인 VRML 환경에서 자바 EAI를 통해 외부세계와의 연결이 될 때는 속도 때문에 많은 에러를 유발하고 어떤 경우에는 다운되는 일이 많았다. 그러나 다행스럽게도 2002년에 가서는 HTML의 차세대 웹언어인 XML 과 3D를 통합한 X3D가 발표될 예정이다. 이렇게 새로운 버전의 3D 언어가 세계 표준안으로 통합이 된다면 1990년대 인터넷의 폭발적인 인기만큼이나

3D를 이용한 많은 애플리케이션의 개발뿐만 아니라 운영체제도 3D를 이용한 운영체계가 만들어 질 것이라고 믿는다. 현재의 컴퓨터 하드웨어의 발전 속도나 초고속 인터넷망의 발전으로 볼 때 앞으로 몇 년간 3D를 활용한 교수 방법에 대한 연구를 계속한다면 학교현장에서 모든 과목을 3D를 활용한 가상환경 교육이 가능하리라 생각한다.

## [참 고 문 헌]

- [1] 숭실대학교 정보과학대학 멀티미디어 연구실: 다중 사용자용 VRML 시스템의 구조, 1995.
- [2] 동아대학교 컴퓨터 공학과: VRML과 EAI를 이용한 3D 상호작용 공간 시스템, 염창근 박경환, 1998.
- [3] 한국생산기술연구원 첨단생산설비연구팀(Advanced Industrial Facility Team): Proposal for a VRML 2.0 Informative Annex(External Authoring Interface Reference), Chris Marrin Silicon Graphics, Inc, 1997.
- [4] 정보통신연구 제12권 제4호: 3차원 가상공간에서의 채팅서비스, 미래채트 개발,1998.
- [5] 시스템 공학 연구소 감성공학연구부 가상현실 연구실: 다중 참여자를 지원하는 가상 쇼핑 시스템.
- [6] 연세대학교 컴퓨터과학과: Development of 3D Virtual Environment Sever supporting Multi-user Participatio,2000.
- [7] 인하대학교 전자계산공학과, 인하공업전문대학 전자계산기과: Learning/Training System Supporting Real Time Video Chatting on Web-based 3D Virtual Space, 2000.
- [8] 초보자를 위한 VRML2 21일 완성: Chris Marrin & Bruce Campbell, 1997.
- [9] 프로그래머를 위한 자바 레퍼런스: Mike Cohn, Bryan Morgan, Michael Morrison, Michael T Nygard, Dan Joshi, Tom Trinko, 1998.
- [10] JAVA for students: Douglas Bell, Mike Parr, 1999.

## [참고 웹 사이트]

- [1] Cosmo Software: <http://cosmosoftware.com/download>.
- [2] Spazz3d, VRML2.0 Animation Editor: <http://www.spazz3d.com/>.
- [3] Web City Office Tower 3D: <http://www.officetowers.com/>.
- [4] Web3D.co.kr: <http://www.web3d.co.kr/vrml/main.html>.
- [5] Blaxxun interactive:  
<http://www.blaxxun.com/support/developerguide/developer/bindex.html>

## [부 록]

### 1. ChatServer.java

```
import java.net.*;
import java.io.*;
import java.util.*;

public class ChatServer{
    protected Vector handlers;

    public ChatServer (int port) throws IOException{
        ServerSocket server =new ServerSocket(port);
        handlers = new Vector(2,5);
        System.out.println("listening from client...");

        while(true){
            Socket client = server.accept();
            System.out.println("login"+client.getInetAddress());
            ChatHandler c = new ChatHandler(this,client);
            handlers.addElement(c);
            c.start();
        }
    }
}
```

### 2. ChatHandler.java

```
import java.io.*;
import java.net.*;
import java.util.*;

public class ChatHandler extends Thread{
    protected Socket      s;
    protected DataInputStream  i;
    protected DataOutputStream  o;
    protected ChatServer      server;
    protected boolean         stop;

    public ChatHandler(ChatServer server, Socket s) throws IOException{
        this.s =s;
        this.server=server;

        i= new DataInputStream(new BufferedInputStream(s.getInputStream()));
        o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
    }
}
```

```

public void run(){
    String name = s.getInetAddress(). toString();
    try{
        name = i.readUTF();
        broadcast(name+ " login.");
        while(!stop){
            String msg = i.readUTF();
            broadcast(name+" : "+ msg);
        }

    } catch(IOException ex){}
    finally {
        server.handlers.removeElement(this);
        broadcast(name + " logout.");
        try{
            s.close();
        } catch(IOException ex) {}
    }
}

protected void broadcast(String message){
    synchronized(server.handlers){
        Enumeration e = server.handlers.elements();
        while(e.hasMoreElements()){
            ChatHandler c = (ChatHandler) e.nextElement();
            try{
                synchronized(c.o){
                    c.o.writeUTF(message);
                }
                c.o.flush();
            } catch(IOException ex){
                stop = true;
            }
        }
    }
}
}

```

### 3. ManinChatClient.java

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import java.util.*;

```

```

// VRML 노드와의 메시지 교환을 위해서..
import vrml.external.Browser;
import vrml.external.Node;
import vrml.external.field.*;
import vrml.external.exception.*;

public class MainChatClient extends Applet implements Runnable,ActionListener,EventOutObserver{
    //public static final int ADDAVATAR = 0;
    //public static final int REMOVEAVATAR = 1;

    protected DataInputStream    i;
    protected DataOutputStream    o;
    protected TextArea          output;
    protected TextField          input,logtext;
    protected Thread             listener;
    protected int                lineCount;
    protected String             lastLine;
    protected CardLayout         card;
    protected boolean            stop;

    //boolean error =false;

    //자식 노드 등록
    boolean view =false;
    Browser browser;
    Node root;

    //creatVrmlFromURL을 위한 배열 생성
    String[] url1;
    String[] url2;
    Node[] shape;
    EventInMFNode addChildren;
    EventInMFNode removeChildren;
    //로그인 텍스트 문자열
    EventInMFString s_text =null;
    String t [] =new String[1];
    //채팅 텍스트 문자열
    EventInMFString chat_text =null;
    String chat [] =new String[1];

    public void init(){
        card =new CardLayout();

```

```

setLayout(card);
Panel login = new Panel(new BorderLayout());
Panel bottom = new Panel();
bottom.setBackground(Color.orange);
logtext =new TextField(15);
logtext.addActionListener(this);
bottom.add(new Label("로그인:")); //로그인 화면 출력
bottom.add(logtext);
login.add("Center",bottom);
Panel chat =new Panel(new BorderLayout());
//로그인에서 엔터를 치면 이화면으로 이동
chat.add("Center",output = new TextArea());
output.setEditable(false);
output.setBackground(Color.green);
chat.add("South",input= new TextField()); //채팅화면 부분 등록
input.setEditable(false);
input.addActionListener(this);
lineCount=0;
add(login,"login");
add(chat,"chat");
card.show(this,"login");
// VRML장면 등록
System.out.println("AddRemove.start() testing...");
output.appendText("AddRemove.start() testing...\n");
browser = (Browser)Browser.getBrowser(this);
output.appendText("Got the browser!...\n");
output.appendText("Got the browser: \n" + browser);
// Root 화면 등록 & 자식노드 등록과 해제
root = browser.getNode("Mainview");
url1 = new String[1];
url1[0] = "man1.wrl";
url2 = new String[1];
url2[0] = "man2.wrl";
addChildren = (EventInMFNode)root.getEventIn("addChildren");
removeChildren = (EventInMFNode)root.getEventIn("removeChildren");
root.getEventOut("children_changed").advise(this,new Integer(1));
//browser.createVrmlFromURL(url1,root,"addChildren");
//텍스트 로그인노드 입력 노드 생성
Node n_text = browser.getNode("TextLogin");
s_text = (EventInMFString) n_text.getEventIn("set_string");
//텍스트 채팅노드 입력 노드 생성
Node chat_n_text = browser.getNode("TextChat");
chat_text = (EventInMFString) chat_n_text.getEventIn("set_string");

```

)

```

public void start(){
    listener = new Thread(this);
    stop=false;
    listener.start();
}
public void stop(){
    if (listener !=null){
        stop=true;
        listener=null;
    }
}
public void run(){
    try{
        String host = getParameter("host");
        if (host==null)
            host = getCodeBase ().getHost();
        String port = getParameter("port");
        if (port==null)
            port = "9830";
        output.append ("Connecting to " + host + ":" + port + "...");
        Socket s =new Socket(host, Integer.parseInt(port));
        i=new DataInputStream(new BufferedInputStream(s.getInputStream()));
        o=newDataOutputStream(newBufferedOutputStream(s.getOutputStream()));
        output.append("\nconnected.\n");
        input.setEditable(true);
        input.requestFocus();
        output.setEditable(false);
        output.requestFocus();
        while(!stop){
            String line = i.readUTF();
            if (lineCount > 50){
                output.setText(lastLine+"\n");
                lineCount =0;
            }
            output.append( line +"\n");
            lastLine = line;
            lineCount ++;
            //로그인
            //VRML의 text Node 접근 이벤트
            t[0]= line;
            s_text.setValue(t);

            //채팅
            //VRML의 text Node 접근 이벤트

```

```

        chat[0]= line;
        chat_text.setValue(chat);

        root.getEventOut("children_changed").advise(this,newInteger(1));
        browser.createVrmlFromURL(url1,root,"addChildren");
    }
} catch (IOException ex){
    ex.printStackTrace(System.out);
}
}

//아바타 추가/제거 함수로 정의 할수 있음
/*public void excuteAdd(){
    try{
        while(!stop){
            String line = i.readUTF();
            if (lineCount > 50){
                output.setText(lastLine+"\n");
                lineCount =0;
            }
            output.append( line +"\n");
            lastLine = line;
            lineCount ++;
            addChildren.setValue(shape);
        }
    } catch (IOException ex){
        ex.printStackTrace(System.out);
    } finally {
        listener = null;
        try {
            o.close();
        } catch(IOException ex){
            ex.printStackTrace();
        }
    }
}

public void excuteRemove(){
    try{
        while(!stop){
            String line = i.readUTF();
            if (lineCount > 50){
                output.setText(lastLine+"\n");
                lineCount =0;
            }
            output.append( line +"\n");

```

```

        lastLine = line;
        lineCount ++;
        addChildren.setValue(shape);
    }
} catch (IOException ex){
ex.printStackTrace(System.out);
} finally {
    listener = null;
    try {
        o.close();
    } catch(IOException ex){
        ex.printStackTrace();
    }
}
}*/
public void actionPerformed(ActionEvent e){
    Component c = (Component) e.getSource();
    if (c== logtext){
        //VRML의 text Node 접근 이벤트
        //t[0]= logtext.getText();
        //s_text.setValue(t);

        //TextArea 부분에 출력
        String loginname= logtext.getText();
        loginname= loginname.trim();
        if (loginname==null|| loginname.length()==0){
            return;
        }
    try{
        o.writeUTF(loginname);
        o.flush();
        card.show(this,"chat");
    } catch(IOException ex){
        ex.printStackTrace(System.out);
    }
} else if (c == input){
    try {
        o.writeUTF(input.getText());
        o.flush();
    } catch(IOException ex) {
        ex.printStackTrace();
        stop = true;
    }
    input.setText(" ");
}

```

```

        }
        //return true;
    }
    public void callback(EventOut value, double timeStamp, Object data) {
        Integer whichNum =(Integer) data;
        if (whichNum.intValue() == 1) {
            EventOutMFNode kids = (EventOutMFNode) value;
            shape = kids.getValue();
        }
    }
}

```

#### 4. ControlChatClient.java

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import java.util.*;

// VRML 노드와의 메시지 교환을 위해서..
import vrml.external.Browser;
import vrml.external.Node;
import vrml.external.field.*;
import vrml.external.exception.*;

public class ControlChatClient extends Applet implements Runnable,ActionListener,EventOutObserver{
    protected DataInputStream    i;
    protected DataOutputStream    o;
    protected TextArea          output;
    protected TextField          input,logtext;
    protected Thread             listener;
    protected int                lineCount;
    protected String             lastLine;
    protected boolean            stop;
    //자식 노드 등록
    boolean view =false;
    Browser browser;
    Node root;
    Node avatar;
    //로그인 텍스트 문자열
    EventInMFString s_text=null;
    String t [] =new String[1];
    //위치,방향 함수 초기화
    Node[] node_list;
    EventInSFVec3f translation = null;
    float t_xyz[] = new float[3];

```

```

EventInSFRotation rotation = null;
float r_xyz[] = new float[4];
EventOutSFVec3f outTranslation = null;
EventOutSFRotation outRotation = null;
public void init(){
    //로그인 장면 등록
    logtext =new TextField(15);
    logtext.addActionListener(this);
    add(new Label("ID를 입력하세요:")); //로그인 화면 출력
    add(logtext);
    //채팅 장면 등록
    add(new Label("채팅입력창"));
    input= new TextField(50); //채팅화면 부분 등록
    input.addActionListener(this);
    add(input);
    //로그인,채팅,EAI 출력화면
    output = new TextArea();
    output.setEditable(false);
    add(output);
    output.setBackground(Color.yellow);
    lineCount=0; //output 출력 카운트 초기화
    // 위치 : xyz 세개의 값이 필요함.
    t_xyz[0] = 0.0f;
    t_xyz[1] = 0.0f;
    t_xyz[2] = 0.0f;
    // 방향 : xyz 와 변수값, 4개의 값이 필요
    r_xyz[0] = 0.0f;
    r_xyz[1] = 0.1f;
    r_xyz[2] = 0.0f;
    r_xyz[3] = 0.0f;
    // VRML장면 등록
    System.out.println("AddRemove.start() testing...");
    output.appendText("AddRemove.start() testing...\n");
    browser = (Browser)Browser.getBrowser(this);
    output.appendText("Got the browser!...\n");
    output.appendText("Got the browser: \n" + browser);
    // Root 화면 등록
    Node root = browser.getNode("Mainview");
    Node avatar = browser.getNode("dad_main_50");
    //텍스트 노드 입력 노드 생성
    Node n_text = browser.getNode("TextLogin");
    s_text = (EventInMFString) n_text.getEventIn("set_string");
    //위치,방향 VRML장면으로 이벤트입력
    translation = (EventInSFVec3f) avatar.getEventIn("set_translation");
    rotation = (EventInSFRotation) avatar.getEventIn("set_rotation");
    //위치,방향 VRML장면으로 이벤트출력

```

```

        outTranslation = (EventOutSFVec3f) avatar.getEventOut("translation");
        outTranslation.advise(this, new Integer(1));
        outRotation = (EventOutSFRotation) avatar.getEventOut("rotation");
        outRotation.advise(this, new Integer(2));
    }
    //VRML장면에서 변화된 반환값을 호출
    public void callback(EventOut who, double when, Object which) {
        Integer whichNum = (Integer) which;
        if(whichNum.intValue() == 1) {
            float[] val = outTranslation.getValue();
        }
        if(whichNum.intValue() == 2) {
            float[] val = outRotation.getValue();
        }
    }
    //서버로 연결을 위한 쓰레드 형성
    public void start(){
        listener = new Thread(this);
        stop=false;
        listener.start();
    }
    public void stop(){
        if (listener !=null){
            stop=true;
            listener=null;
        }
    }
    public void run(){
        try{
            String host = getParameter("host");
            if (host==null)
                host = getCodeBase ().getHost();
            String port = getParameter("port");
            if (port==null)
                port = "9840";
            output.append ("Connecting to " + host + ":" + port + "...");
            Socket s =new Socket(host, Integer.parseInt(port));
            i= new DataInputStream(newBufferedInputStream(s.getInputStream()));
            o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
            output.append("\nconnected.\n");
            input.setEditable(true);
            input.requestFocus();
            output.setEditable(false);
            output.requestFocus();
            while(!stop){
                String line = i.readUTF();
                if (lineCount > 50){

```

```

        output.setText(lastLine + "\n");
        lineCount =0;
    }
    output.append( line +"\n");
    lastLine = line;
    lineCount ++;

    //VRML의 text Node 접근 이벤트
    t[0]= line;
    s_text.setValue(t);
}
} catch (IOException ex){
    ex.printStackTrace(System.out);
}
}
//Key나 Mouse로 아바타 조정
public void paint(Graphics g) {
    g.drawString("-----" ,1,40);
    g.drawString("Key와Mouse로 조정하세요.", 5, 50);
    g.drawString("UP: 앞 DOWN:뒤",8,70);
    g.drawString("LEFT:왼쪽 RIGHT:오른쪽",8,90);
    g.drawString("ENTER: up-left",8,110);
    g.drawString("PGDN: up-right",8,130);
    g.drawString("-----" ,1,140);
    g.drawString("3D장면으로 서핑하려면", 1,150);

    g.drawString("-----" ,540,40);
    g.drawString("PGUP: down-left",550,50);
    g.drawString("END: down-right",550,70);
    g.drawString("DELETE: left-rotation",550,90);
    g.drawString("HOME: right-rotation",550,110);
    g.drawString("-----",550,140);
    g.drawString("3D공간을 한번 누르세요.",550,150);
}
public boolean keyDown(Event event, int key) {
    switch (key) {
        case Event.UP: //up
            t_xyz[2]-=1.0f;
            break;
        case Event.ENTER: //up,left
            t_xyz[0]-=1.0f;
            t_xyz[2]-=1.0f;
            break;
        case Event.PGDN: //up,right
            t_xyz[0]+=1.0f;
            t_xyz[2]-=1.0f;
            break;
    }
}

```

```

        case Event.DOWN: //down
            t_xyz[2]+=1.0f;
            break;
        case Event.PGUP: //down,left
            t_xyz[0]-=1.0f;
            t_xyz[2]+=1.0f;
            break;
        case Event.END: //down,right
            t_xyz[0]+=1.0f;
            t_xyz[2]+=1.0f;
            break;
        case Event.LEFT: //left
            t_xyz[0]-=1.0f;
            break;
        case Event.RIGHT: // right
            t_xyz[0]+=1.0f;
            break;
        case Event.DELETE://left rotation
            r_xyz[3]+=1.0f;
            break;
        case Event.HOME: //right rotation
            r_xyz[3]-=1.0f;
            break;
    }
    translation.setValue(t_xyz);
    rotation.setValue(r_xyz);
    return true;
}

/*public boolean mouseDown(Event event, int x, int y) {
    t_xyz[0] = (x-50)/30.f;
    t_xyz[2] = -(y-50)/30.f;
    translation.setValue(t_xyz);
    return true;
}*/

public void actionPerformed(ActionEvent e){
    Component c = (Component) e.getSource();
    if (c== logtext){
        //TextArea 부분에 출력
        String loginname= logtext.getText();

        try{
            o.writeUTF(loginname);
            o.flush();

        } catch(IOException ex){
            ex.printStackTrace(System.out);
        }
    }
}

```

```

        stop =true;
    }
} else if (c == input){
    try {
        o.writeUTF(input.getText());
        o.flush();
    } catch(IOException ex) {
        ex.printStackTrace();
        stop = true;
    }
    input.setText(" ");
}
}
}

```

나머지 VRML코드화면은 너무 많아서 여기에서 생략합니다.

## 설문지 작성

영어 가상학교에서 1시간 동안 수업한 뒤 설문을 작성하여 주시기 바랍니다. 여러분이 웹 기반의 3D 가상학교를 체험한 뒤 다음과 같은 질문에 솔직히 답변 해주시기 바랍니다.

1. 이 가상학교의 3D 가상화면 설계(디자인)는 어떠한가?

- ① 그저 그렇다. ( )
- ② 보통이다. ( )
- ③ 좋다. ( )
- ④ 아주 좋다. ( )

2. 이 가상학교의 수업 모형의 흥미도는?

- ① 그저 그렇다. ( )
- ② 약간 흥미롭다. ( )
- ③ 흥미롭다. ( )
- ④ 아주 흥미롭다. ( )

3. 3D 화면 채팅에서 교사나 학생들의 상호작용에 대한 이해도는?

- ① 그저 그렇다. ( )
- ② 약간 이해한다. ( )
- ③ 제법 이해한다. ( )
- ④ 거의 다 이해한다. ( )

4. 3D 영어 채팅에서 3D 공간에서 몰입(immersion)이 되어 수업의 성취효과가 있는가?

- ① 효과가 없다. ( )
- ② 약간 효과가 있다. ( )
- ③ 어느 정도 효과가 있다. ( )
- ④ 아주 많은 효과가 있다. ( )

5. 3D 가상학교에서 수업 시 프로그램이 에러 없이 원활하게 수행하는가?

- ① 원활하지 못하다. ( )
- ② 그저 그렇다. ( )
- ③ 원활하다. ( )
- ④ 아주 원활하다. ( )

6. 3D 가상학교의 개선해야할 점을 적어주세요.

( )

## 감사의 글

이렇게 작으나마 보람있는 결실을 보게끔 해준 신봉기 교수님께 진심으로 감사의 글을 올립니다. 부족한 저를 조금이라도 나은 글이 되도록 음으로 양으로 도와주었을 뿐만 아니라 어려운 코드부분을 정리해주고 충고해주신 점 감사하게 생각합니다.

그리고 여름 방학 때 문제가 해결되지 않아 고민하고 있을 때 프로그램 방향과 조언을 주셨던 정목동 교수님, 서경룡 교수님에게도 진심으로 감사를 드리고 기회가 있으면 소찬이라도 대접하고 싶습니다.

학위 과정동안 동고동락하며 어려움이 있을 때 서로 도움을 주고 아낌없이 사랑을 주었던 우리 2년차 4명에게도 고마움을 전하고 또한 우리 2년차의 부족한 점을 선배로서 채워 주셨던 박창수님께 깊은 감사의 뜻을 전합니다.

오늘이 있기까지 칠순 평생동안 오직 당신의 자식만을 위하고 희생해오신 저의 모친에게 정말 40년 평생동안 한번도 ‘고맙습니다’ 는 말을 한 적이 없는 데 이 지면을 통해 진심으로 저희 모친에게 고마움을 전합니다.

마지막으로 오늘의 제가 있기까지 뒤에서 저의 수족이 되고 눈이 되어주었던 홍식이 준현이 엄마에게 작지만 제가 처음으로 이루어 낸 작은 결실을 바칩니다.

2001년 12월

박 순 철 올림