

가상환경에서의 인터랙션을 위한 모바일 시스템용 인터페이스 프레임워크 개발

김 석 환[†] · 조 옹 주^{**}

요 약

가상환경이란 사용자들에게 실감나는 감각적 체험과 인터랙션을 제공하여 마치 실제 공간에 있는 것 같은 느낌을 주는 멀티미디어 환경을 일컫는다. 이런 가상환경에 일반 사용자들에게 친숙한 2차원 인터페이스를 활용하여 사용자 인터랙션을 쉽게 도와주는 연구가 많았는데, 주로 특정 가상환경에서만 사용할 수 있고 일반적으로 적용하기 어려운 문제가 있었다. 이런 점을 개선하기 위해 가상환경을 위한 2차원 사용자 인터페이스 제작 프레임워크 개발 시도가 있었으나 사용성이나 유연성이 부족했다. 본 연구에서는 XML과 스크립트 언어를 활용하고 동적으로 모듈을 적재하고 설정하는 기능과 이벤트 처리 메커니즘을 제공하여 사용성과 유연성을 지원하는 새로운 가상현실 인터페이스 제작 툴킷인 HIVE 프레임워크를 만들었다. 본 논문에서는 먼저 HIVE프레임워크에 대해 설명하고, HIVE를 이용하여 다양한 인터페이스를 개발한 예를 보여주며, 향후 연구방향에 대해 논한다.

키워드 : 가상현실, 가상환경, 프레임워크, 가상계기, 모바일 인터랙션

The Development of Mobile-based User Interface Framework for Virtual Environments

Seokhwan Kim[†] · Yongjoo Cho^{**}

ABSTRACT

Virtual environment is an interactive multimedia space that allows users to experience realistic sensory feedbacks and interactions as they do in real world. There have been some work on the development of two-dimensional user interface for supporting interactions in a virtual environment. However, most of them were designed for specific virtual environments and hence could not be easily deployed to other virtual worlds. To address this problem, some frameworks were developed for building two-dimensional user interfaces for virtual environment, but they had limitations in usability or flexibility. In this paper, we present the HIVE framework, which is designed to help ease the development of two-dimensional user interface for the virtual environment while satisfying the requirements of both usability and flexibility. The HIVE framework utilizes XML and scripting capability, dynamic module loading, XML and script language for usability. It provides module loading, configuration, and event processing mechanism for flexibility. This paper first explains the HIVE framework and describes the applications developed for some virtual environments followed by discussing the future research directions.

Key Words : Virtual Reality, Virtual Environment, Framework, Virtual Instrument, Mobile Interaction

1. 서 론

가상현실이란 “사용자가 컴퓨터를 통해 인공적으로 만들어진 환경과 인터랙션을 통하여 실제인 것처럼 경험하고 몰입할 수 있도록 도와주는 하드웨어와 소프트웨어로 구성된 시스템이다”[1]. 이런 가상현실 시스템에서의 인터랙션은 가

상환경과 사용자가 서로 상호작용할 수 있는 매개체 역할을 한다. 그리고 자연스럽게 쉬운 인터랙션은 사용자의 몰입감을 증대시켜 사용자로 하여금 가상환경 안에 실제 존재하는 느낌을 준다.

가상환경에서의 인터랙션을 지원하는 대표적인 3차원 인터페이스로는 Ray Casting[2], Two-Handed Pointing[2], Aperture Technique[2], Virtual Glove[3] 등이 있다. 그러나 이런 종류의 3차원 인터페이스는 6자유도(6 Degrees of Freedom)의 입력을 지원하므로 초보 사용자들이 자유롭게 사용하기에는 어렵게 느껴지는 경우가 많다. 특히 이런 장

※ 본 연구는 2007년도 상명대학교 소프트웨어·미디어 연구소에서 연구비를 지원받았습니다.

† 정 회 원 : 상명대학교 컴퓨터학과 석사과정

** 정 회 원 : 상명대학교 디지털미디어학부 (교신저자)

논문접수 : 2007년 7월 21일, 심사완료 : 2007년 9월 6일

치들이 사용하기 어렵다는 점은 초보자들로 하여금 가상환경에서의 인터랙션이 어렵다는 선입견을 갖게 한다.

그래서 초보자들을 위해 3차원 입력 장치 대신에 기존의 데스크톱 컴퓨터에서 사용하고 있는 2차원 인터페이스를 활용하여 가상환경에서의 인터랙션을 지원하는 방법이 시도되었다. 그 예로 Virtual Field[4]는 PDA에 2차원 지도를 제공하고 이를 이용하여 사용자는 지정하는 위치로 이동하고 데이터를 수집할 수 있도록 했다. 또한 Virtual Aquaria[5]에서 사용자는 PDA의 2차원 인터페이스를 통해 가상 수족관의 어류를 선택하고 해당 어류에 대한 정보를 얻을 수 있다. Virtual Harlem[6]에서 사용자는 AMIS (Annotator Management Interface System)를 이용하여 메모를 하거나 2차원 지도를 통해 현재의 위치를 알아내는 용도로 2차원 인터페이스를 사용할 수 있다. 이런 프로젝트들에서의 사용 예를 종합해서 보면 가상환경에서 특정 지역으로 이동하거나 텍스트 기반의 정보를 수집 또는 제공할 때 2차원 인터페이스가 다른 장치들보다는 효과적이라는 것을 알 수 있다.

그러나 이런 2차원 인터페이스를 개발하는 것은 많은 노력이 필요하다는 문제점이 있다. 예를 들어 간단한 모바일용 인터랙션 인터페이스를 만들기 위해서는 해당 기기에서 돌아가는 운영체제의 API, 가상환경과 모바일 시스템을 연결할 통신 API, C++와 같은 고수준 프로그래밍 언어, 그리고 가상현실 시스템과 모바일 기기에 대한 이해가 필요하다. 이런 어려움을 해결하고자 연구자들에 의해 VIST[7], Tweek[8], Glass Library[9] 와 같은 프레임워크가 개발되었으나, 이들 역시 모두 다양한 사용자의 요구에 부응하는 다양한 응용프로그램의 제작을 도와주는 유연성에 치중한 나머지 너무 복잡해지거나 혹은 사용하기 쉽게 만드는 것을 강조하여 너무 단순해진다는 단점이 존재했다.

본 논문에서는 새로운 모바일기기용 2차원 인터랙션 인터페이스 제작 및 실행 프레임워크인 HIVE (Handheld-based Interface development toolkit for Virtual Environment) 에 대해서 알아보고 실제 프로젝트에서 활용한 예를 설명한다. HIVE는 스크립트를 이용하여 가상현실을 위한 2차원 인터페이스를 제작할 수 있는 프레임워크로 기존의 프레임워크들이 보여주었던 단점들을 해결하기 위해 만들어졌다. 본 논문의 2절 관련 연구에서는 기존 프레임워크의 장단점을 분석하고 3절에서 HIVE 프레임워크에 대한 설명을 한다. 4절에서는 HIVE를 이용한 예제 인터페이스에 대해 설명한 후 5절에서 결론과 향후 연구 분야에 대해서 논한다.

2. 관련 연구

PDA는 기존의 데스크톱 컴퓨터보다는 컴퓨팅 능력이나 메모리 모두에서 낮은 성능을 가지고 있다. 하지만 현재의 PDA는 많은 성능 개선을 통하여 멀티미디어 데이터, 사운드 등을 처리하기에 부족함이 없는 성능을 가지게 되었다. 또한 마우스나 키보드와 같은 별도의 입력장치를 사용하지 않고 터치패드를 이용해서 바로 입력을 할 수 있다는 장점

을 가진다. 별도의 입력장치가 없다는 것은 사용자가 몸에 어떤 장치를 부착하지 않아도 되고, 다른 입력 장치를 배울 필요 없이 손가락과 같은 신체의 일부분을 이용해서 편리하게 입력을 할 수 있다는 점을 말한다. 몇몇 프로젝트에서는 휴대용 컴퓨터의 장점을 활용하는 인터페이스를 개발하였지만 대부분의 경우는 특정 가상환경에 종속적이었기 때문에 새로운 가상환경을 개발할 때마다 이에 맞게 인터페이스를 다시 개발해야 하는 어려움이 존재했다 [4,6,10].

이러한 어려움을 극복하기 위하여 모바일기기 기반 애플리케이션 제작을 도와주려는 몇 개의 프로젝트가 시도되었다. 먼저 Tweek은 Java™, Java Beans, XML, CORBA와 같은 기술들을 사용하여 만들어진 인터페이스 디자인 툴킷으로 동적으로 GUI 컴포넌트를 적재해서 응용프로그램을 만들 수 있도록 해주는 메커니즘을 제공한다. Tweek은 범용 애플리케이션을 만들기 위해 디자인되었기 때문에 사용자는 어떤 형태의 GUI 인터페이스도 만들 수 있고, 이를 활용한 예로는 간단한 2차원 맵을 PDA에 디스플레이하고, 사용자는 간단히 맵에서 원하는 위치를 클릭함으로써 이동할 수 있는 응용프로그램이 있다[8]. Tweek은 CORBA 기술을 이용하여 분산 오브젝트간의 통신을 이용해서 운영체제나 프로그래밍 언어에 대해 독립적인 환경을 제공하고, 여러 운영체제와 여러 가지 언어를 함께 사용할 수 있는 유연성을 제공한다. 하지만, C++, JAVA 라는 프로그래밍 언어와 CORBA라는 컴포넌트 기반 개발 기술에 대한 개념의 이해를 요구한다는 점에서 배우기 어려운 것이 단점이다.

이와 반대로 좀 더 쉽게 가상현실에서 필요한 2차원 인터페이스를 구성하는 것을 목표로 만들어진 툴킷인 VIST는 Tweek 보다는 확실히 간단한 방법으로 GUI 컴포넌트를 활용할 수 있도록 해준다. 하지만 너무 간편함을 강조한 나머지 스크립트만으로는 인터랙션에 필요한 어떤 논리적인 프로그램도 만들 수 없으며 이를 위해서는 하위 단계(lower level)인 모듈 차원에서 모든 프로그램의 기능들이 구현되어야 하고 스크립트는 모듈에서 만든 로직을 호출하는 형태로 만들어진다. 따라서 간단한 수정도 모듈 차원에서 이루어지는 경우가 많다.

Glass Library 는 분산 컴퓨팅을 위한 라이브러리로 분산 컴퓨팅 환경에서 사용할 수 있는 PDA (Personal Data Assistant) 의 인터페이스 제작을 지원한다. Glass 라이브러리는 GUI 툴킷을 이용하여 인터페이스를 제작 가능케 함으로서 위에 설명한 프레임워크 보다 좀 더 쉬운 인터페이스 디자인을 가능케 한다. 하지만, 사용자가 구성할 수 있는 인터페이스가 툴킷이 제공하는 템플릿에 한정되는 단점을 가지고 있다. 인터랙션을 위한 이벤트 처리의 경우, 사용자가 인터페이스 요소를 추가할 때마다 툴킷은 각기 다른 정수형의 식별자를 부여하고 사용자가 디자인을 저장하면 툴킷은 식별자를 이용한 C언어 문법의 스위치문을 가지고 있는 C 파일을 생성하고, 파일의 스위치 문에 원하는 C언어 코드를 삽입하여 이벤트 처리 로직을 완성한다. 따라서 사용자는 C

프로그래밍 언어와 같은 고수준 프로그래밍 언어를 배워야 한다는 어려움을 갖는다.

3. HIVE 프레임워크

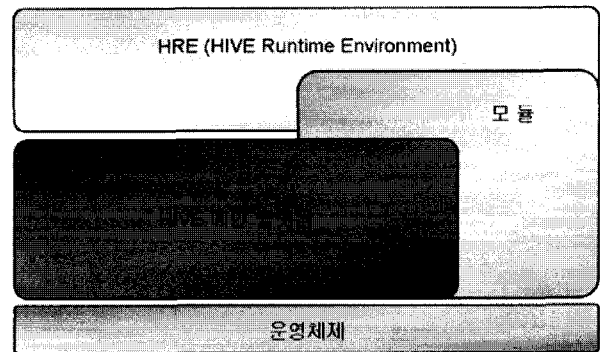
HIVE 프레임워크는 VIST에서의 사용성과 Tweek에서의 유연성을 동시에 만족시키는 것을 목적으로 한다. 프레임워크를 디자인할 때에는 일반적으로 사용성과 유연성이라는 두 가지의 요구사항을 완벽하게 지원하는 것을 목표로 하지만 실제로는 두 가지가 서로 상반되는 경우가 많으므로 어느 한쪽을 치중해서 선택해야 하는 경우가 많다.

HIVE 프레임워크는 사용성과 유연성을 높이기 위해 HIVE 스크립트 언어를 사용한다. HIVE 스크립트 언어는 VIST의 스크립트 언어처럼 간단한 XML 문법을 사용하여 사용성을 높였고, 스크립트 내부에 Lua[11] 스크립트 언어를 사용해서 프로그램을 작성할 수 있게 해주어서 유연성을 높이고 있다. 즉 개발자는 전체 시스템의 기능들을 모듈로 나누어서 개발하고 그 모듈들을 연결시켜 함께 동작할 수 있도록 해주는 스크립트 코드를 작성할 수 있으며, 필요하다면 새로운 기능들을 스크립트 언어를 이용해서 개발해 추가할 수도 있도록 만들었다. 이렇게 하면 자주 변하는 사용자의 요구사항들을 개발하기 어렵고 오래 걸리는 모듈로 만드는 것이 아니라 쉽고 빠르게 만들 수 있는 스크립트 언어로 만들 수 있다. VIST와 같은 프레임워크는 스크립트언어의 역할이 단순히 공유 라이브러리 파일을 적재하고 실행하는 것에 국한되어 있기 때문에 기능의 변경 또는 추가가 필요할 때 모듈을 수정해야 했다. 이와 비교해서 HIVE는 스크립트 상에서 이를 가능케 함으로서 좀 더 쉽게 기능을 변경하고 추가할 수 있다.

HIVE 프레임워크는 개발자가 원하는 C언어 또는 C++언어 기반의 라이브러리를 사용하여 모듈을 제작할 수 있다. 따라서 Glass 라이브러리 보다는 다양하고, Tweek과 같은 프레임워크만큼 다양한 모듈을 제작할 수 있는 유연성을 갖는다. 하지만, Tweek에서 사용하는 CORBA와 같은 컴포넌트 기반 기술을 사용하지 않기 때문에 모듈 개발이 좀 더 쉽다. 또한 HIVE 스크립트는 VIST의 스크립트와 같이 단순히 모듈을 적재하고 실행하는 것도 가능하기 때문에 여러 가지 요구사항에 다양한 모듈을 제작하여 유연하게 대처할 수 있으면서도 사용성을 잃지 않는다.

(그림 1)은 HIVE 시스템의 전체적인 구조를 보여준다. HIVE 시스템은 크게 외부 모듈 부분과 코어 부분으로 나누어진다. 외부 모듈은 스크립트로 구현하기 어려운 인터페이스의 기능들을 운영체제에 기반을 둔 공유 라이브러리의 형태로 만들어 HIVE 스크립트를 이용하여 사용할 수 있도록 만든 것을 일컫는다. 코어 부분은 HRE (HIVE Runtime Environment)와 HIVE 메인 플랫폼 (HIVE Main Platform)으로 구성된다.

HIVE 메인 플랫폼은 운영체제와 가장 밀접한 부분으로 외부 모듈을 메모리에 적재하여 실행시키고, HIVE 스크립트



(그림 1) HIVE 프레임워크 전체 구조

팅 시스템을 구동시키는 역할을 한다. HRE는 앞에서 설명한 바와 같이 HIVE 시스템의 유연성을 높이기 위해 채택된 스크립트 코드를 분석, 저장, 실행하는 역할을 한다. HRE는 그 스크립트로 만들어진 코드를 포함하고 있다가 외부 모듈이나 사용자의 입력을 통해 이벤트가 생성되면 이벤트 핸들러 (Event Handler) 스크립트 코드를 실행하여 이벤트 처리 결과를 사용자에게 보여줄 수 있도록 한다.

3.1 HIVE 메인 플랫폼

HIVE 메인 플랫폼은 바이너리 형태의 실행파일이다. 사용자는 HIVE 메인 플랫폼을 구동함으로써 프레임워크를 시작한다. HIVE 메인 플랫폼은 시작과 동시에 (그림 1)의 윗 부분에 있는 HRE를 실행하여 공유 라이브러리 모듈에 대한 정보를 얻고 이를 이용하여 모듈을 메모리에 적재한 후 실행시키고 관리한다. 여기서 메인 플랫폼은 GUI 모듈을 담아서 보여주는 부모 윈도우의 역할을 한다. 메인 플랫폼은 동적으로 적재한 각 모듈에 대한 정보, 스크립트를 실행하는 HRE에 관한 정보 등을 관리하며 메인 플랫폼 자신과 모듈 그리고 HRE 간의 통신을 가능케 하는 역할을 한다.

3.2 HIVE 스크립트

HIVE를 활용하는 인터페이스 프로그램이 처음 실행될 때, HRE는 사용자가 인터페이스를 정의한 스크립트 코드를 읽어서 HMDL (HIVE Module Description Language)과 HRSL (HIVE Runtime Script Language)로 나누어 처리한다. HMDL은 처음 실행할 때 적재해야 하는 모듈의 이름, 위치와 크기 등에 대한 정보를 가지고 있다. 이 정보들은 HIVE 메인 플랫폼에 전달되어 해당 모듈을 지정된 위치에 배치하고 크기를 조정하는데 활용된다. HRSL은 실행시간에 동적으로 변경될 수 있는 데이터를 저장하고 이벤트 처리를 위한 스크립트이다. HRSL은 사용자 혹은 시스템에서 발생하는 이벤트에 대한 정보와 그 이벤트를 처리하기 위한 Lua 스크립트로 만들어진 이벤트 핸들러로 구성된다. HIVE 프로그램이 실행될 때 이벤트가 발생하면 HRE는 이벤트 핸들러에 해당되는 스크립트 코드를 실행한다.

(그림 2)는 GUI 모듈을 활용하는 HIVE 스크립트의 예제

```

1 <HIVE>
2 <GUI_MODULE file = "Block.dll">
3 <NAME>block</NAME>
4 <POSITION x = "20" y = "10" z = "0"/>
5 <ORIENTATION x = "0" y = "0" z = "0"/>
6 <SIZE x = "200" y = "700" z = "0"/>
7
8 <EVENT type = "MOUSE_LEFT_BUTTON_DOWN">
9 ProcessBtnDown (block.GetX(), block.GetY())
10 </EVENT>
11
12 <EVENT type = "MOUSE_LEFT_BUTTON_UP">
13 ProcessBtnUp(block.GetX(), block.GetY())
14 </EVENT>
15
16 <EVENT type = "MOUSE_LEFT_BUTTON_DOWN_AND_MOVE">
17 ProcessBtnDownAndMove()
18 </EVENT>
19 </GUI_MODULE>
20
21 <SCRIPT>
22 block.ToTopWindow();
23 net.Connect("127.0.0.1", 7500, 15, 14)
24 net.Run()
25 </SCRIPT>
26 </HIVE>
    
```

(그림 2) HIVE 스크립트 예제

이다. HIVE의 외부 모듈은 GUI (Graphical User Interface) 모듈과 일반 모듈로 나누어진다. 이름이 의미하는 것처럼 GUI 모듈은 화면에 나타나는 그래픽 사용자 인터페이스처럼 화면에 시각적으로 보이는 버튼 같은 모듈을 나타내고 일반 모듈은 네트워크 모듈처럼 화면에 보이는 부분없이 실행될 수 있는 모듈을 일컫는다.

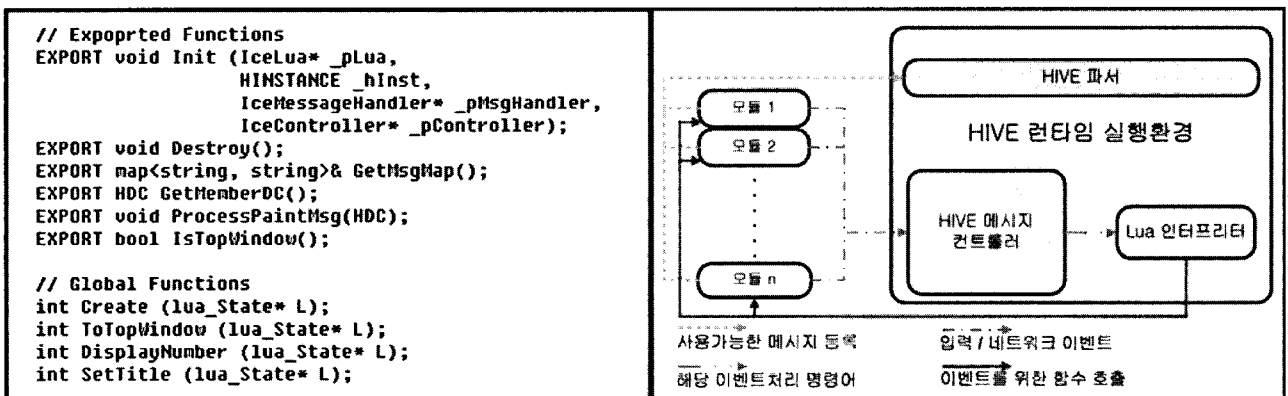
(그림 2)에서 보인 GUI 모듈에서 처음에 나오는 <GUI_MODULE> 태그는 이후에 나오는 하위 XML 태그들의 내용이 GUI 모듈에 관한 것이라는 것을 의미하고, XML 속성 (attribute)인 file은 적재할 모듈의 공유 라이브러리 파일의 이름을 지정한다. 그 다음에 나오는 <SCRIPT> 태그는 모듈이 처음 메모리에 적재될 때 실행되는 부분으로 Lua 스크립트로 작성되어지고 주로 현재 적재되는 모듈의 초기화 작업에 활용된다. <SCRIPT> 태그는 HIVE 스크립트 언어의

어느 곳에서든 지 사용할 수 있는 태그로 내부의 TEXT 데이터 (실제 스크립트의 내용)는 HRE로 전달되는 즉시 실행된다. <POSITION>, <SIZE> 등의 태그는 일반 모듈에서는 활용되지 않고, GUI 모듈의 경우에는 모듈의 위치, 크기, 방향등을 나타내는 값으로 3차원 인터페이스로의 확장을 고려해서 x, y, z의 속성을 갖도록 디자인하였고, <ORIENTATION> 태그 역시 x, y, z축으로의 회전을 염두에 두고 만들었다.

<EVENT> 태그는 type이라는 속성을 가지고 있다. 이 속성은 이벤트의 이름을 의미하고 태그사이에 들어가는 Lua 스크립트는 type에서 정의된 이벤트가 발생되었을 때 자동으로 HRE에서 실행된다. 이 때 이벤트는 주로 외부 모듈에서 많이 발생하게 되는데 모듈을 만드는 개발자들은 모듈이 발생시키는 이벤트의 이름을 공개하면, 스크립트 개발자는 (그림 2)의 8번 줄에서 18번 줄과 같은 형식으로 스크립트를 작성해서 이벤트 핸들러를 만든다. 즉, HIVE 프레임워크는 해당 모듈에서 상용하는 이벤트가 발생했을 경우 위에서 보인 EVENT 스크립트를 코드를 실행하여 메시지를 처리한다.

HRSL에 포함되는 Lua 스크립트 언어는 분기문, 반복문, 함수, 전역 및 지역 변수와 같은 일반적으로 고수준 프로그래밍 언어에서 제공하는 대부분의 기능을 제공함과 동시에 타입 없는 변수 지원과 같은 쉬운 스크립트 언어의 특징을 갖는다. 따라서 사용자는 간단한 기능의 수정이나 추가가 필요한 경우 모듈의 수정 없이 쉽게 HRSL을 수정함으로써 원하는 기능을 실행할 수 있다. 이는 결과적으로 모듈의 재사용성과 유연성을 높여주는 효과가 있다.

예를 들어, 고구려 안악 3호분 프로젝트[12]에서는 사용자의 모습과 그림의 위치를 보여주는 2차원 지도 인터페이스를 제공하는데, 이 가상 환경은 전체 가상공간의 중심으로 위치 데이터를 생성하여 윈도우즈에서 실행되는 지도 인터페이스 프로그램에 네트워크를 통해 전달한다. 이 때 지도 인터페이스를 만드는 모듈은 윈도우즈 시스템의 좌표계 (원점이 좌상단에 위치함)를 사용하기 때문에 서로간의 좌표계를 변환시켜주는 부분이 필요하다. HIVE 프레임워크를 사용하는 응용 프로그램에서는 HIVE 스크립트를 이용해서 좌



(그림 3) HIVE 프레임워크에서 사용되기 위한 모듈의 함수 원형 (왼쪽), 스크립트 언어와 이벤트의 처리를 위한 메시지의 흐름 (오른쪽).

표 변환 공식을 구현하여 추후에 가상 공간이나 GUI 시스템의 좌표계가 달라지더라도 유연하게 대처할 수 있도록 만들 수 있다.

3.3 HIVE 외부 모듈

HIVE 메인 플랫폼에서 지원하는 외부 모듈은 HIVE스크립트를 실행하는 HRE (HIVE Runtime Environment)를 이용해 제어할 수 있도록 디자인되었다. 또한, HIVE 메인 플랫폼에서 외부 모듈을 초기화 시키고 활용할 수 있도록 하기 위해 몇 개의 특정 함수를 지정하고 모든 HIVE 외부 모듈로 하여금 이 함수들을 구현하도록 하고 있다.

(그림 3)의 왼쪽 부분에서는 실제 구현된 HIVE 모듈을 구성하는 함수들의 원형 (prototype)을 보여준다. 이 함수들 중 Init, Destroy, ProcessPaintMsg, GetMsgMap, IsTopWindow 함수들은 GUI 모듈을 만들 때 개발자들이 반드시 구현해야 한다. GUI화면을 활용하지 않는 일반 모듈은 ProcessPaintMsg, IsTopWindow 함수를 제외한 Init, Destroy, GetMsgMap 함수를 구현한다. HIVE 프레임워크에서는 모듈을 개발하는 프로그래머들을 위해 GUI 모듈과 일반 모듈용 기반 클래스(base class)인 “VISUAL_COMPONENT”와 “COMPONENT” 클래스를 제공하고 있는데 이 클래스에 반드시 구현되어야 하는 함수들을 순수 가상함수로 선언하여 구현하지 않는 실수를 사전에 방지하고 있다.

Init 함수는 모듈을 초기화하는데 필요한 부분을 지정한다. 예를 들어, GUI 위젯을 구현하는 모듈의 경우에는 위젯 컴포넌트의 위치, 크기 등에 대한 정보들을 이 함수에서 초기화할 수 있고, 네트워크 모듈과 같은 일반 모듈에서는 네트워크 연결에 사용되는 아이피 주소, 포트 번호 등을 초기화할 수 있다. Destroy 함수는 모듈이 종료될 때 호출되는 함수로 힙 영역이나 파일 같이 모듈에서 사용했던 자원을 반납하는 코드를 포함한다. GetMsgMap은 이벤트를 처리하는데 반드시 필요로 하는 함수로, 미리 해당 모듈에서 처리 가능한 이벤트의 목록을 반환하는 함수이다.

ProcessPaintMsg 함수는 GUI 위젯을 포함하는 모듈이 다시 그려져야 할 경우에 HIVE 메인 플랫폼이 호출하는 함수이다. 마지막으로 IsTopWindow 함수는 여러 개의 GUI 위젯들이 동시에 적재되고 중첩된 경우에 순서를 정해서 이벤트를 전달하거나 그림을 그리는 부분을 위임하기 위해 활용되는 함수이다. 예를 들어 중첩된 부분에 이벤트가 들어왔을 경우 어느 GUI 위젯이 이벤트를 수신하는지 또는 어느 위젯이 그려져야 하는지에 대한 결정을 내리는 것이 모듈간의 우선순위가 없는 HIVE시스템에서는 불가능하다. 이때 프레임워크는 IsTopWindow 함수를 호출하여 어떤 모듈이 가장 위에 있는 GUI 위젯인 지를 판단하고 그 위젯에게 이벤트를 보낸다.

3.4 HIVE 실행 환경

(그림 3)의 함수 원형에서 Init 함수를 더욱 자세히 살펴

보면 IceLua 라는 클래스의 포인터 타입이 첫 번째 인자로 넘겨지는 것을 볼 수 있다. 이 클래스는 Lua 스크립트 언어에서 제공하는 Lua API[11]라는 C기반 함수들을 HIVE에서 사용할 수 있도록 C++의 클래스로 재구성 한 것이다. IceLua 클래스는 AddFunction, RunScript, PushData, PopData 등과 같은 함수들을 제공해서 HRE에 내장되어 있는 Lua환경과 HIVE를 연결시켜 준다.

예를 들어 IceLua 클래스의 AddFunction함수를 활용하면 모듈에서 C언어로 만든 함수들을 HRE환경에 직접 등록시키는 것이 가능하고, 이렇게 등록된 함수는 HIVE 시스템에서 Lua 스크립트를 통해 활용할 수 있다. 이렇게 스크립트에서 활용 가능한 C언어로 구현된 함수에 인자를 전달하거나 함수의 결과 값을 스크립트 상에서 돌려받을 수 있는 기능이 필요하여, HRE로부터 데이터를 전달받는데 활용하는 PopData, HRE에 데이터를 전달하는 PushData함수를 제공하고 있다.

HRE의 AddFunction이 모듈에서 구현된 함수를 스크립트에서 활용할 수 있는 기능을 제공해주는 반면, RunScript는 모듈 안에서 스크립트 코드를 실행시켜 준다. 이 함수는 스크립트 코드를 인자로 전달받아서 실행시킨다. 이 함수를 이용하면 모듈 내부에서 다른 모듈을 제어하는 스크립트를 실행함으로써 다른 모듈에게 명령을 전달할 수 있다.

HRE에서는 앞서 언급되었던 것처럼, 개발자들이 만드는 모듈에서 구현된 함수들을 스크립트 언어에서 활용할 수 있도록 등록시키는 기능을 제공한다. 그런데 HIVE 메인 플랫폼에서는 다양한 개발자들이 만든 모듈들을 조합해서 함께 사용하는 경우에는 각 모듈에서 구현되어 스크립트에 등록되는 함수들의 이름들이 충돌되는 경우가 발생할 수도 있다. 즉 A와 B라는 모듈에서 GetName이라는 함수들을 동시에 등록시킨다면 이름 충돌이 발생하여 모듈에서 제공하는 함수를 사용할 수 없다.

HRE에서는 이렇게 모듈에서 중복되어 제공될 수 있는 함수들을 관리하기 위해 Lua언어에서 제공하는 테이블[11]이라는 독특한 자료구조를 활용하여 C++의 네임스페이스와 비슷한 기능을 제공한다. 즉 모듈에서 제공하는 함수들을 처음 모듈이 등록될 때 테이블 구조로 묶어 “모듈이름.함수이름”의 형태로 활용할 수 있도록 해준다. 예를 들어 ice라는 이름을 가진 모듈의 get이라는 함수를 등록할 때 ice라는 이름의 테이블을 생성하고 테이블의 멤버로 get 함수를 등록하여 ice.get 이라는 형태로 함수에 접근할 수 있도록 등록한다.

결과적으로 HRE는 전역 데이터의 저장, 사용가능한 함수 리스트의 관리, 함수 호출시 인자 전달과 결과 값 리턴에 필요한 루틴을 모두 실행하고 관리함으로써 인터랙션 처리에 필수가 되는 이벤트를 처리하는 메커니즘을 제공한다.

3.5 HIVE 메인 플랫폼의 이벤트 처리

앞서 기술했듯이 HIVE 메인 플랫폼에서 사용되는 모듈은 크게 GUI 모듈과 일반 모듈로 나누어지고 각각의 모듈

은 사용자로부터 또는 시스템으로부터 이벤트를 받을 수 있다. HIVE 프레임워크는 이 두 가지 이벤트를 모두 같은 형식으로 처리한다. 각각의 이벤트는 모두 이름을 가지고 있는데, 이벤트가 발생했을 때 모듈은 이벤트의 이름을 포함한 미리 정의된 형식으로 이벤트를 포장하여 이를 (그림 3)의 오른쪽에 보이는 HIVE Message Controller로 전송한다.

HIVE Message Controller는 메시지를 받으면 해당 메시지를 처리하는 스크립트의 존재여부를 HRE에 문의한다. HRE는 스크립트가 존재하면 해당 스크립트를 Message Controller에 전달하고 없으면 아무런 응답을 하지 않는다. 스크립트 루틴을 전달받은 Message Controller는 이를 다시 HRE에 내장되어 있는 Lua 실행환경에 전달해서 실행시킨다. 스크립트가 실행될 때 각 명령문들은 각각의 해당하는 모듈에 전달되고 모듈은 이를 실행하여 이벤트 처리를 종료한다.

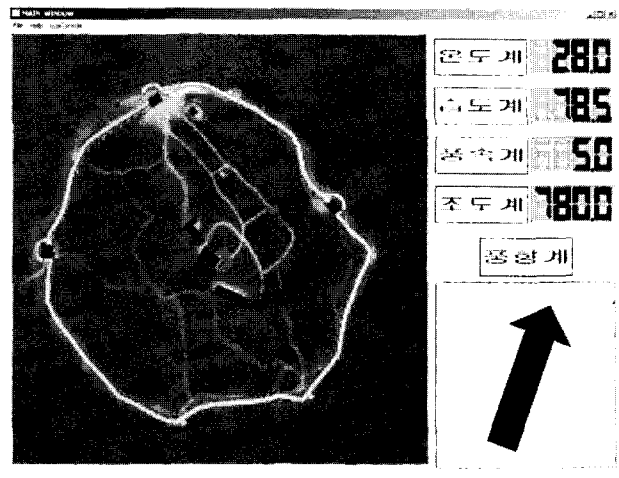
4. HIVE를 이용한 가상 계측기

HIVE 시스템의 유용성을 평가하고자 우리는 HIVE 시스템을 이용하여 디지털 고창 프로젝트[13]의 가상 계측기 인터페이스를 제작했다. (그림 4)는 고창 프로젝트에서 활용된 인터페이스를 보여주고 있다. (그림 4)의 왼쪽은 고창 모양성 탐구학습 프로젝트의 배경이 되는 모양성을 위에서 바라본 모양의 맵뷰 인터페이스이다. 맵뷰 인터페이스는 전체 가상현실의 월드를 위에서 바라본 이미지를 사용자에게 제공하고 사용자가 가상환경에서 이동시 자신이 현재 가상환경에서 어디에 위치하는지를 점으로 표시해준다.

(그림 4)의 오른쪽은 가상계측기들을 보여준다. 가상 계측기는 온도계, 습도계, 조도계, 풍속계, 풍향계로 이루어져 있다. 이 중 온도계, 습도계, 조도계, 풍속계는 모두 디지털 타입의 숫자를 디스플레이하여 사용자는 각각의 계기가 제공하는 정보를 얻는다. 풍향계는 화살표를 디스플레이하여 사용자가 현재 바람이 어느 쪽에서 불어오는지에 대해 알 수 있도록 해준다.

HIVE 시스템의 사용성과 유연성을 살펴보기 위해, 고창 프로젝트의 인터페이스를 구현하면서 다른 프로젝트를 위해 만들어졌던 모듈 두 개를 재활용하고 새로운 모듈을 제작하였다. 기존의 모듈들은 디지털 고구려 프로젝트[12]의 맵뷰에 사용하였던 모듈로 가상공간에서의 사용자의 위치를 2차원 지도 위에 실시간으로 보여주는 맵뷰(MapView) GUI 모듈과 가상공간과 네트워크를 통해 데이터를 주고받을 수 있는 네트워크 모듈이다. 이 두 가지 모듈 모두 기존의 프로젝트만을 위해 개발되었던 것이어서 모듈을 수정하지 않고도 새로운 프로젝트에 쉽게 적용시킬 수 있는지를 알아보기 위해 채택했다.

우선 디지털 고구려에서의 인터페이스는 단순하게 지도만을 보여주고 사용자의 위치를 실시간으로 확인할 수 있도록 해주었기 때문에 맵뷰와 네트워크 모듈을 제외한 다른 모듈을 활용하지 않았다. 하지만, 고창 프로젝트의 맵뷰는 다른 가상계기들과 함께 인터페이스를 구성하는 일부분으로 활용



(그림 4) HIVE 프레임워크를 이용한 예제 응용프로그램

되었다. 이는 모듈을 동적으로 적재하는 부분을 기술한 HIVE 스크립트에서 사용하는 모듈들을 추가하고 위치와 크기에 대한 정보를 다시 지정하여 기존의 모듈을 수정하지 않고도 쉽게 활용할 수 있었다. 네트워크 모듈의 경우, 기존의 디지털 고구려 프로젝트와 디지털 고창의 월드의 크기가 다르기 때문에 맵뷰에서 사용자의 좌표를 올바르게 표시하기 위해서 좌표 변환을 다른 형태로 하는 것이 필요했다. 이 역시 기존 네트워크 모듈에서 가공되지 않은 원시 데이터 값 (Raw data)를 받아서 HIVE 스크립트 상에서 그 값을 다시 계산하여 맵뷰 모듈로 전달해 줌으로서 기존 네트워크 모듈을 재활용할 수 있었고, 결과적으로 응용프로그램을 좀 더 쉽게 개발할 수 있게 하여 사용성의 우수함을 확인할 수 있었다.

온도, 습도, 풍속, 조도를 표현하는 가상계기들은 모두 한 개의 모듈을 활용하였다. 디지털 디스플레이 모듈은 디지털 숫자위에 표시되는 이름과 숫자 값을 인자를 통해 설정할 수 있도록 제작했다. 따라서 동적으로 네트워크를 통해 가상현실 시스템으로부터 데이터를 받으면 HIVE 스크립트를 실행하여 표시하는 값을 변경한다. 풍향계는 실시간으로 가상환경에서 알려주는 바람의 방향을 사용자층이 되는 학생들이 쉽게 볼 수 있도록 화살표를 활용해서 보여준다. HIVE 모듈 개발자는 모듈 개발시 C언어 또는 C++언어 기반의 모든 라이브러리를 이용하여 HIVE 프레임워크가 요구하는 함수를 포함한 공유 라이브러리의 형태로 제작한다. 예를 들어 위문단에 기술한 온도, 습도 등을 표현하는 가상계기와 풍향계는 고창 프로젝트를 위해 새로 제작한 모듈로 Windows native API를 이용했다. 이를 이용하여 우리는 기존 프레임워크들이 기본으로 제공하는 텍스트 필드, 버튼과 같은 기본적인 GUI 컴포넌트가 아닌 (그림 4)의 오른쪽에 보이는 디지털 타입 숫자 디스플레이와 화살표 디스플레이 같은 완전히 새로운 GUI 컴포넌트를 개발할 수 있었다. 개발자는 자유롭게 여러 가지 라이브러리를 이용할 수 있고 이를 이용하여 다양한 모듈을 제작할 수 있다. 따라서 이는 사용자가 원하는 어떤 모듈도 제작할 수 있게끔 하여 전체

시스템의 유연성을 높여준다.

HIVE 프레임워크를 이용해 인터페이스를 제작한 결과, 맵뷰에서의 좌표 변환 등과 같은 간단한 데이터 변환 작업이나 가공 작업은 기존의 모듈들이 재사용성을 감안해서 만들어진 것들이 아니라도 어느 정도까지는 스크립트를 수정해서 적용시킬 수 있었고 이는 결과적으로 모듈의 재활용성을 높여주어 사용성을 증대시키는 것을 확인할 수 있었다. 또 그 밖에 새로운 모듈이 필요한 경우 자유롭게 원하는 모듈을 제작함으로써 다양한 요구사항의 변경에 유연하게 대처할 수 있는 것은 확인할 수 있었다.

5. 결론 및 향후 연구방향

본 논문에서 우리는 가상현실과의 인터랙션에 사용될 수 있는 2차원 인터페이스 개발을 위한 HIVE 프레임워크와 이를 활용한 예제 애플리케이션을 소개하였다. 가상환경에서 제공되는 인터페이스들은 그 사용법이나 제공하는 기능들이 일반 사용자가 쉽게 적응하기 어려워 기존의 데스크톱 인터페이스를 제공하는 작은 휴대용 장치들을 이용해 인터페이스로 활용하는 연구가 진행되어 왔다. 단 이런 인터페이스의 개발은 가상환경과 휴대용 컴퓨터 프로그래밍에 관한 지식을 요하고 따라서 시간이 많이 걸리고 어려운 작업이 되는 경우가 많다. 또 초기에 개발된 가상 인터페이스들은 주로 특정 가상환경에만 적합하도록 만들어져서 재활용성이 떨어지는 경우가 많았다.

HIVE 시스템에서는 Lua라는 스크립트 언어와 XML 문법을 활용한 HIVE 스크립트 언어를 제공하여 유연성과 사용성을 제공한다. HIVE를 이용하여 여러 개의 인터페이스를 만들어보았을 때 기존의 모듈이 특정 가상환경을 위해서 만들어졌을 때에도 스크립트를 통해서 데이터를 재가공하여 재활용성이나 유연성을 높이는 것을 확인하였다. 또한 HIVE시스템을 이용해서 기존의 모듈의 내용을 고치지 않더라도 쉽게 인터페이스를 작성할 수 있도록 해주는 사용성도 확인할 수 있어 그 유용성이 입증된다.

현재 HIVE 시스템은 초기 단계에 있다. 그 예로 HIVE시스템의 모듈을 보면 지도를 보여주는 맵뷰, 가상환경과 인터페이스를 네트워크를 통해 연결해주는 모듈 등 응용 프로그램의 단위 모듈로 존재하는 것들이 많다. 향후 HIVE 시스템에서는 모듈들을 특정 응용 프로그램의 단위 모듈이 아니라 일반적인 가상환경에서의 인터랙션 스킴 (scheme) 단위-예를 들면 내비게이션(navigation), 선택(selection), 조작(manipulation)-로 제공할 수 있는 프레임워크로 발전시키고 이의 유용성을 평가할 것이다.

참 고 문 헌

[1] P. Frederick Jr, "What's Real About Virtual Reality?," IEEE Computer Graphics & Application, Nov / Dec, 16,

1999.
 [2] D. Bowman, E. Kruijff, J. LaViola, I. Popoyrev, "3D User Interfaces: Theory and Practice," Addison-Wesley Professional, pp. 139-182, 2004.
 [3] M. Folgheraiter, G. Gini, D. Vercesi, "A Glove Interface with Tactile feeling display for Humanoid Robotics and Virtual Reality system," Proc. International Conference ICINCO 2005, pp. 13-17, 2005.
 [4] 조용주, 박경신, "가상필드에서 초등학생들의 과학점 탐구를 도와주는 멀티미디어 보조도구," 한국정보처리학회 논문지 B, pp. 143-150, 2005년 4월.
 [5] G. Wetzstein, P. Stephenson, "Towards a Workflow and Interaction Framework for Virtual Aquaria," VR for Public Consumption, IEEE VR Workshop, 2004.
 [6] K. Park, J. Leigh, B. Carter, J. Brody, J. Sosnosok, "Distance Learning Classroom Using Virtual Harlem," Proceedings of Seventh International Conference on Virtual System and Multimedia, pp. 489-498, 2001.
 [7] 김석환, 정다영, 조용주, "가상환경에서의 사용자 인터랙션 지원을 위한 가상 계측 단말기 스크립팅 툴킷," 제 23회 한국정보처리학회 춘계학술발표대회 논문집 제 12권 1호, pp. 155-158, 2005.
 [8] P. Harting, A. Bierbaum, C. Cruz-Neira, "Virtual reality interfaces using Tweek," International Conference on Computer Graphics and Interactive Technique, pp. 278-278, 2002.
 [9] P. Guimaraes, B. Gnecco, K. Zuffo, "Graphical interaction devices for distributed virtual reality system," Proceedings of ACM SIGGRAPH international conference on Virtual Reality continuum and its application in industry, pp. 363-367, 2004.
 [10] M. Wloka, E. Greenfield, "The Virtual Tricorder: A Unified Interface for Virtual Reality," Proceedings of the ACM Symposium on UIST, pp. 39-40, 1995.
 [11] R. Ierusalimschy, "Programming in Lua," Lua.Org, 2005
 [12] Y. Cho, K. Park, S. Park, H. Moon, "Designing Virtual Reality Reconstruction of the Koguryo Mural," CDVE 2005. LNCS, vol. 3675, Springer, Heidelberg, pp. 194-201, 2005.
 [13] K. Park, H. Cho, J. Lim, Y. Cho, S. Kang, S. Park, "Learning Cooperation Skills in a Tangible Moyangsung," HCII, Lecture Notes in Computer Science 4563, pp. 689-698, 2007.



김 석 환

e-mail : compedian@gmail.com
2007년 상명대학교 소프트웨어학부(학사)
2007년~현재 상명대학교 컴퓨터과학과
석사과정
관심분야: 가상현실, HCI



조 용 주

e-mail : ycho@smu.ac.kr
1993년 일리노이대학 컴퓨터과학과
(공학사)
1997년 일리노이대학
전기전자컴퓨터과학과(공학석사)
2003년 일리노이대학 컴퓨터과학과
(공학박사)
2004년~현재 상명대학교 미디어학부 조교수
관심분야: 가상현실, HCI, 인터랙티브 컴퓨팅