

# 실감형 사용자 인터페이스를 위한 XML 기반 저작도구의 구현

## Implementation of an Authoring Tool for Tangible user Interface

서진석\*, 김준호\*, 권덕중\*\*, 김홍준\*\*, 오세웅\*, 김정현\*\*\*, 김창현\*\*\*

동의대학교 게임공학과\*, 동의대학교 컴퓨터소프트웨어공학과\*\*, 고려대학교 컴퓨터통신공학부\*\*\*

JinSeok Seo(jsseo@deu.ac.kr)\*, JunHo Kim(kim.junho@deu.ac.kr)\*,  
Duk-Joong Kwon(rejent@naver.com)\*\*, HongJoon Kim(kingkkk@hanmail.net)\*\*,  
Sei-Woong Oh(osw@deu.ac.kr)\*, Gerard JoungHyun Kim(gkim@korea.ac.kr)\*\*\*,  
Chang-Hun Kim(chkim@korea.ac.kr)\*\*\*

### 요약

실감형 사용자 인터페이스를 사용하는 상호작용을 디자인하고 구현하기 위해서는 장치의 제어, 입력 데이터의 센싱과 보정, 상호작용 디자인, 로우레벨 프로그래밍, 성능 최적화 등과 같은 다양한 분야의 지식이 필요하다. 게다가, 주어진 상호작용 장치와 콘텐츠의 특성에 따라 사용할 수 있는 상호작용 기법을 적절하게 선택 및 조합하기 위해서는 많은 시행착오를 겪어야 하기 때문에, 최상의 사용성을 달성하기 위해서는 많은 노력과 시간이 필요하다.

본 논문은 다양한 장치를 조합하여 제작된 실감형 상호작용 플랫폼을 소개하고 있으며, 위에서 언급한 어려움을 해결하고 실감형 상호작용을 쉽게 구현할 수 있도록 개발된 XML 기반의 저작도구를 설명하고 있다. 더불어, 몇 개의 예시 콘텐츠를 통하여 연구결과의 효용성을 검증하고자 한다.

■ 중심어 : 실감형 사용자 인터페이스 | 상호작용 디자인 | 저작도구 |

### Abstract

The design and implementation of the interaction for tangible user interfaces require in-depth knowledge in many different disciplines, such as device control, sensing and calibrating devices, interaction design, low-level programming, and performance tuning. Many trial and error iterations are needed to determine the proper combination of the interaction techniques while using available interaction devices and considering the characteristics of contents. As a result, it takes too much effort and time to achieve maximum usability.

This paper introduces a tangible user-interface platform, which is fabricated using various hardware devices and an XML-based authoring tool, which is developed in order to relieve content creators of the burden of the above difficulties. Finally, we demonstrate our work by illustrating some example contents.

■ keyword : Tangible User Interface | Interaction Design | Authoring Tool |

## I. 서론

본문실감형 사용자 인터페이스는 직접 손으로 만지

고 느끼면서 상호작용할 수 있다는 장점 때문에 최근 관심이 매우 높아지고 있지만, 아직 다양한 분야의 콘

\* 본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 지역문화산업연구센터(CRC) 지원사업의 연구결과로 수행되었습니다.

접수번호 : #080624-003

심사완료일 : 2008년 07월 16일

접수일자 : 2008년 06월 24일

교신저자 : 서진석, e-mail : jsseo@deu.ac.kr

텐츠 및 플랫폼에서의 활용도가 높지 않다. 기존의 WIMP(Windows, Icon, Menu, Pointing Device) 기반의 상호작용을 주로 사용하는 경우에는 많은 저작도구 및 기술 등이 개발되어, 하드웨어나 컴퓨터 프로그래밍과 같은 전문 지식을 모르는 콘텐츠 디자이너나 기획자도 쉽게 콘텐츠를 설계하고 구현할 수 있는 환경이 구축된 지 오래되었지만, 실감형 사용자 인터페이스 분야의 경우 아직 많은 연구 및 기술 개발이 필요하다.

실감형 사용자 인터페이스 기반 상호작용의 구현이 어려운 대표적인 이유는 아직 표준화 되지 않은 여러 종류의 실감형 인터페이스 장치를 사용하기 위해서는 여러 분야의 전문 지식이 필요하기 때문인데, 그 예로, 장치의 제어 방법, 입력 데이터의 센싱과 보정, 서로 성격이 다른 여러 종류의 모달리티를 기반으로 하는 상호작용의 디자인, 디바이스와 통신하기 위한 로우 레벨 프로그래밍 기술, 각기 다른 데이터 샘플링 속도를 갖는 장치들 간의 성능 최적화 등을 들 수 있다. 게다가, 상호작용을 디자인 및 구현할 때 가장 염두에 두어야 할 높은 사용성을 보장하기 위해서는 많은 노력과 시간이 필요한데, 이는 주어진 장치들과 콘텐츠의 특성에 따라 사용할 수 있는 상호작용 기법을 적절하게 선택 및 조합하기 위해서는 많은 시행착오를 겪어야 하기 때문이다.

이에 본 논문에서는 콘텐츠 제작자들이 위와 같은 하위 수준의 기술적인 전문지식은 사용하지 않고 그들이 원하는 콘텐츠의 완성도에만 전념하면서도 실감형 사용자 인터페이스를 사용하는 상호작용을 쉽게 구현할 수 있는 XML 기반의 저작도구를 소개하고 있다.

본 논문의 2장에서는 저작도구 및 저작 방법과 관련된 연구에 대한 설명을 하고 있고, 3장에서는 콘텐츠가 수행될 실감형 상호작용 플랫폼을 소개하고 있으며, 4장에서는 저작도구의 설계 및 구현 방법을 서술하고 있다. 5장에서는 콘텐츠 제작 결과를 바탕으로 연구결과의 효용성을 보여주고 있고, 마지막으로 6장에서 결론을 맺는다.

## II. 관련 연구

최근에는 증강현실 및 컴퓨터비전 등의 기술의 발전

으로 실감형 사용자 인터페이스 기반의 상호작용이 많이 활용되고 있으며, 특히 편재적 컴퓨팅 환경에 적합한 기술로 많은 관심 받고 있다. [1]에서는 편재적 컴퓨팅 환경이 도래함에 따라 요구되는 인터페이스 방식의 변화를 실감형 사용자 인터페이스를 중심으로 고찰하고 있으며, [2]에서는 상호작용에 과정에 영향을 주는 요소(물리적 형상, 물리적 속성 등)와 사용자 반응, 그리고 시스템의 리액션에 대한 연구를 수행하였다.

가상현실 분야에서는 여러 가지 종류의 모달리티를 제공하기 위해서 다양한 입력 장치를 사용하여 왔는데, VRPN(Virtual-Reality Peripheral Network)[3]은 네트워크에 분산되어 있는 이기종 플랫폼에서 구동되는 각종 물리적인 가상현실 입력 장치(트래커, 3D 마우스, 글러브 등)를 사용할 수 있도록 C++ 언어 기반의 클래스 라이브러리를 제공한다. VRPN은 현재까지도[4] 가상현실분야 뿐만 아니라 다양한 분야의 콘텐츠에서도 활발하게 활용되고 있지만, 컴퓨터 프로그래밍 언어에 익숙하지 않은 콘텐츠 제작자들이 사용하기에는 많은 숙련 기간을 필요로 한다.

"Merigold toolset"[5]은 "Extended Petri-net"[6]이라는 정형화 도구를 사용하여 가상환경 시스템의 상호작용을 다이어그램 형태로 저작할 수 있다. 이산적이면서도 연속적이라는 상호작용의 특성을 고려한 정형화 도구를 제시하였다는 점에서는 의미가 있지만, 저작된 결과인 프로그래밍 코드의 뼈대에 사용자가 직접 프로그래밍을 해야 한다는 단점이 있다.

본 논문에 직접적으로 동기를 부여해 준 연구로는 InTML[7]과 GlovePIE[8]를 들 수 있다. InTML은 C++와 같은 프로그래밍 언어를 사용하지 않고 XML 기반의 마크업 언어를 사용하여 상호작용 기법을 디자인할 수 있도록 도와준다. 데이터플로우(dataflow) 아키텍처를 기반으로 하여, 상호작용 기법을 필터로 정의한 후에 필터의 입력 포트에 장치 데이터를 연결하고, 출력 포트에는 데이터의 처리 결과를 받게 될 대상을 연결하는 방식으로 상호작용을 설계하고 XML 문서로 작성하는 것이다

본 논문의 저작도구도 XML 기반의 마크업 언어를 사용하고 있지만, InTML과는 달리 비전문가도 저작할

수 있도록 GUI 기반의 도구를 제공하고 있으며, 가상현실 콘텐츠에만 국한되지 않고 실감형 사용자 인터페이스와 같은 다양한 모달리티를 지원한다.

GlovePIE(Glove Programmable Input Emulator)는 조이스틱, 게임패드, MIDI 입력 장치, 트랙커, 글러브, 닌텐도사의 위모트 컨트롤러 등과 같은 다양한 입력 장치를 게임과 데스크톱 응용프로그램에서 사용할 수 있도록 제작된 프로그램이다. 데스크톱 응용프로그램은 키보드와 마우스만 지원한다고 가정하고 각종 입력 장치에서 발생하는 입력 데이터를 키보드와 마우스 이벤트로 변경시켜준다. 장점은 이미 개발된 콘텐츠에 새로운 입력장치를 쉽게 추가하거나 변경할 수 있다는 것인데, 본 논문의 저작도구도 키보드나 마우스만 사용할 수 있는 기존의 콘텐츠에 다양한 실감형 사용자 인터페이스 기반의 멀티모달 상호작용을 사용할 수 있도록 하였다. 게다가, 각 상호작용 제작 결과를 조합하여 재사용하기 쉽도록 하기 위하여 XML 기반의 마크업 언어를 정의하고 있다.

비록 제한된 상호작용만 제작이 가능하지만, [9]에서는 실감형 사용자 인터페이스를 사용하여 상호작용을 구현할 수 있는 시스템을 제안하였다. 이 시스템은 콘텐츠 제작자나 프로그래머가 아닌 일반인들도 쉽게 실감형 인터페이스를 사용하여 콘텐츠를 만들 수 있도록 “programming-by-demonstration”[10] 접근 방식을 사용하고 있다.

선행연구인 PVoT[11]에서는 멀티모달 상호작용을 Statecharts[12] 기반의 정형화된 기법으로 명세할 수 있으며, 명세 결과를 그대로 즉석에서 수행시켜 볼 수 있다. 비록 실감형 사용자 인터페이스를 구현할 수는 없지만, 정형화된 도구를 사용하여 복잡한 상호작용의 행위 모델을 쉽게 구현하고 검증할 수 있다는 장점이 있다.

### III. 실감형 상호작용 플랫폼

실감형 상호작용 플랫폼[그림 1]은 여러 가지 센서를 사용하여 제작되었으며, LCD, RFID 리더, 다점 인식

터치스크린, 영상 카메라, 각속도 센서/수신기, 마이크, 스피커 등과 같이 여러 가지 센서 및 하드웨어로 구성되어 있다.

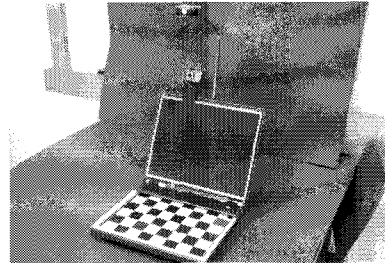


그림 1. 실감형 상호작용 플랫폼

#### 1. LCD

LCD는 PC에서 수행되는 콘텐츠의 수행 화면을 그대로 전송하는 역할을 한다. 터치스크린, RFID 리더 등의 다른 하드웨어 장치들과의 통합을 위하여 노트북 PC와 비슷한 형태로 구성하였다.

#### 2. RFID 리더

실감형 상호작용에 사용되는 모든 도구에 RFID 태그를 부착하여, 현재 사용자가 다루고 있는 물체를 인식할 수 있도록 한다. RFID 태그는 각기 고유의 ID를 가지고 있으므로 동시에 여러 개의 물체 인식이 가능하다. [그림 2]는 “요리조리”라는 예시 콘텐츠에서 사용하고 있는 도구들이다.

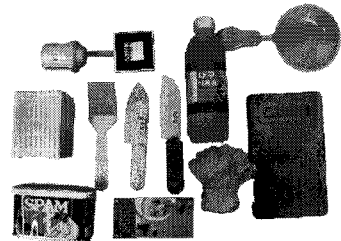


그림 2. RFID 태그가 부착된 실감형 도구들

#### 3. 다점 인식 터치스크린

한 점밖에 인식할 수 없는 감압식 터치스크린의 경우 다양한 상호작용 기법을 사용하기 어렵다. 본 플랫폼에

서 사용된 터치스크린은 광학 유닛을 사용하고 삼각 측량 방식의 원리를 이용하는 다점 인식 터치스크린을 사용하였다.

#### 4. 영상 카메라

2개의 카메라를 사용하여 특정 물체의 3차원 위치 및 방향을 추적할 수 있도록 구성하였으나, 안정적인 컴퓨터 비전 알고리즘이 완성되지 않아 본 플랫폼에서는 1개의 카메라와 ARToolKit[13]을 사용하여 마커가 부착된 물체의 3차원 위치 및 방향을 인식하도록 하였다.

#### 5. 각가속도 센서 수신기

영상 카메라를 이용한 컴퓨터 비전 기술은 주위의 조명 조건에 변화가 발생하거나 사용자의 손 등으로 인식하고자 하는 물체를 가리게 되면 인식하기 힘들다. [그림 3]과 같이 항상 손으로 가려지는 물체와 같은 경우 도구(인형) 안에 각가속도 센서를 부착하여 물체의 움직임을 감지하도록 하였다.

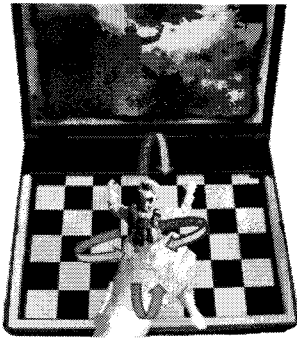


그림 3. 각가속도 센서가 부착된 인형을 이용하여 스카이다이버 게임을 진행하는 화면

### IV. 저작도구의 구현

#### 1. 이벤트

실감형 사용자 인터페이스에 사용되는 장치들은 샘플링 방식에 의해 입력 데이터를 획득하는 경우가 많은 반면, 기존의 WIMP 기반의 콘텐츠는 주로(키보드와

마우스의 경우) 콜백(callback) 방식에 의해 입력 장치로부터의 입력을 감지한다. 원래 샘플링 방식에 의해 입력 데이터를 획득하는 장치로부터 콜백 방식에서 필요한 이벤트를 가공할 수 있도록 각 장치의 입력 데이터를 감시하는 모듈 및 이벤트 관련 클래스를 디자인하였다.

이벤트에는 RFID 태그의 특정 ID 검출, 터치 센서의 접촉 여부 감지 등과 같은 이산적(discrete) 이벤트와 터치센서 접촉면의 변화, AR 마커의 위치 및 방향 변화, 각가속도 센서의 방향 변화 등과 같은 연속적(continuous) 이벤트로 나눌 수 있다. 본 저작도구에서는 실제 장치에서 발생하는 이벤트를 콘텐츠에서 필요로 하는 이벤트로 변환하는 방법을 이벤트의 종류에 따라 3가지로 구분하고 있다.

표 1. 이산적 이벤트에서 이산적 이벤트로의 변환

장치	실제 이벤트	변환된 이벤트
RFID	태그 id 검출	키보드 키의 누름
		마우스 버튼의 클릭
터치 센서	접촉 감지	키보드 키의 누름
		마우스 버튼의 클릭

첫 번째, [표 1]과 같이 물리적인 장치에서 발생하는 모든 이산적 이벤트는 콘텐츠에서도 이산적인 이벤트로 변환된다.

표 2. 연속적 이벤트에서 연속적 이벤트로의 변환

장치	실제 이벤트	변환된 이벤트
터치 센서	접촉면의 좌표 (sx, sy)	마우스의 좌표(tx, ty)
AR 마커	마커의 위치(sx, sy, sz)	sx, sy, sz 중 2개를 마우스의 위치 값으로 변환
	마커의 방향 (roll, pitch, yaw)	roll, pitch, yaw 중 2개를 마우스의 위치 값으로 변환
각가속도 센서	센서의 방향 (roll, pitch, yaw)	roll, pitch, yaw 중 2개를 마우스의 위치 값으로 변환

두 번째, 물리적인 장치의 연속적인 이벤트를 콘텐츠에서도 연속적인 이벤트로 사용하는 경우는 [표 2]와 같다. 각가속도 센서의 경우 원래 수집되는 데이터는 3차원 방향(roll, pitch, yaw 값)인데 이중 2개를 마우스

커서의 좌표로(tx, tx)로 변환한다. AR 마커로부터 수 집되는 데이터도 마찬가지지만, 이 경우 변환되는 값들의 범위를 지정해주어야 하는데, 한 가지 예로 -30도에서 +30도 범위의 roll값을 0에서 1024 범위의 마우스 커서 tx값으로 변환하는 것을 들 수 있다.

표 3. 연속적 이벤트에서 이산적 이벤트로의 변환

매핑 방법	설명 (sx의 경우만)
범위 이용	sx의 값이 특정 범위를 넘어가면 이벤트 계속 발생
	sx의 값이 특정 범위를 넘어가면 이벤트 한번만 발생
변위 이용	sx의 값이 최근 이벤트 발생 이후로 특정 변위만큼 변화 되었다면 이벤트 발생 (타이머 사용)

마지막으로, 원래 물리적인 장치에서는 연속적인 이벤트이지만 콘텐츠에서는 이산적 이벤트로 사용되는 경우는 [표 3]과 같이 정리할 수 있다. [표 3]은 sx(예를 들어 AR 마커의 x축 상의 위치)에 대해서만 설명하고 있지만, sx, sy, roll, pitch, yaw의 경우도 동일하게 변환 방법을 나눌 수 있다.

## 2. 실감형 사용자 인터페이스 API

실감형 사용자 인터페이스 API는 최종 목표인 저작도구가 사용하는 하위 레벨의 모듈이기도 하고 C++와 같은 프로그래밍 언어에 익숙한 전문가가 사용할 수 있도록 디자인 되었다. API는 각 장치를 초기화 하고, 변환관계를 정의하고, 입력 데이터를 조사할 수 있는 함수를 제공한다.

모든 물리적인 장치마다 별도의 클래스로 구현하였으며, 각 클래스는 해당 장치에서 발생하는 입력 값을 샘플링하여 콘텐츠가 원하는 이벤트로 변환하여 전송한다. 현재 구현된 센서 클래스 MotionSensor, ARSensor, TouchSensor, RFIDSensor이다. 새로운 센서의 추가나 뒤에서 언급할 동적 매핑을 위해서 최상위 센서 클래스인 TUIPhysicalSensor를 정의하였다.

실감형 사용자 인터페이스 API에서 제공하는 C++ 클래스를 직접 사용하는 예는 [그림 4]에서 볼 수 있다. 이 그림은 각가속도 센서의 roll 값을 키보드 키 누름 이

벤트로 변환하는 예를 보여 주고 있는데, 2번째 줄은 roll 값이 45도보다 넘게 되면 “아래 화살표 키”가 눌린 이벤트로 변환하는 것을 말한다.

```
TUIMotionSensor *m_sensor = new TUIMotionSensor;
m_sensor->SetRangeMapRollMax(45,TUI_KEY_RIGHT,true);
m_sensor->SetRangeMapRollMin(-45,TUI_KEY_LEFT,true);
m_sensor->SetRangeMapPitchMax(45,TUI_KEY_DOWN,true);
m_sensor->SetRangeMapPitchMin(-45,TUI_KEY_UP,true);
```

그림 4. 각 실감형 사용자 인터페이스 API의 사용 예

## 3. 입력 장치와 상호작용의 동적 매핑

사용의 편리성을 보장하는 상호작용을 구현하기 위해서는 많은 시행착오를 겪어야 한다. 특정 상호작용을 설계하는 경우, 우선 가장 적절한 입력장치를 선택해야 하며, 실제 발생하는 이벤트로부터 원하는 이벤트로의 변환 과정에 개입하는 여러 가지 변수의 값을 세밀하게 조절해야 할 필요가 있다. 개발된 저작도구는 콘텐츠가 수행중일 때에도 입력장치와 상호작용의 매핑 관계뿐만 아니라 이벤트 변환 방법도 동적으로 변경 가능하도록 설계되었다.

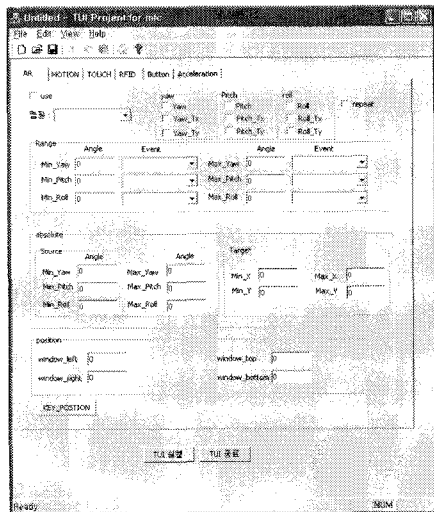


그림 5. 저작도구의 실행 화면

#### 4. 저작도구

[그림 5]는 완성된 저작도구의 실행 화면이다. GUI 환경에서 원하는 입력 장치를 선택한 후 이벤트 변환 관계를 정의하도록 되어있다. 저작된 결과는 XML 기반의 마크업 언어인 TML로[그림 6][그림 7] 저장되며, 저장된 TML을 사용하여 조합 등의 방법으로 재사용이 가능하다.

### V. 구현 콘텐츠

저작도구의 효용성을 검증하기 위하여 총 10개의 콘텐츠를 제작하였다. 제작 의뢰는 C++와 같은 프로그래밍 언어에 대한 지식이 없는 콘텐츠 제작 전문가에게 하였으며, 콘텐츠 제작자는 실제로 사용할 하드웨어 장치에 대한 사전 지식이 없는 상태에서 제작하였다. 본 논문에서는 그중 2개의 콘텐츠에 대하여 설명하고 있다

#### 1. 예시 콘텐츠 1: 요리조리

“요리조리”는 RFID 태그가 부착된 각종 요리 재료와 조리 도구[그림 2]를 주로 사용하는 콘텐츠이다. 콘텐츠의 특성상 여러 물체가 겹쳐 있어도 인식이 가능해야 하고, 소금통과 같이 직접 접촉하지 않은 물체도 인식해야 하기 때문에, RFID 기술을 활용하기에 매우 적합하다.

```
<?xml version="1.0" encoding="utf-8"?>
<tml>
  <sensor type="RFID">
    <button_map id="E004010008B4E559" msg="TUL_KEY_D"/>
    <button_map id="E0040100162B4001" msg="TUL_KEY_H"/>
    <button_map id="E0040100162B4002" msg="TUL_KEY_O"/>
    <!-- 중략 -->
    <button_map id="E0040100162B4008" msg="TUL_KEY_M"/>
    <button_map id="E0040100162B4009" msg="TUL_KEY_J"/>
    <button_map id="E0040100162B417E" msg="TUL_SPACE"/>
  </sensor>
</tml>
```

그림 6. 요리조리 콘텐츠의 상호작용 저작 결과

콘텐츠 제작자는 RFID와 같은 특수한 장치의 작동 원리를 모르므로 콘텐츠에 필요한 각 소품(칼, 도마, 배

추 등)마다 별도의 “키보드 키 누름“ 이벤트를 할당하여 제작하였다. 물론, 제작 과정 중의 테스트나 디버깅은 키보드만 사용하여 진행되었다. 콘텐츠가 완성된 후, 개발된 저작도구를 사용하여 실제로 필요한 각 소품에 부착된 RFID 태그의 ID 인식 이벤트를 콘텐츠 제작자가 정의한 키보드 키 이벤트로 변환되도록 정의하였다. [그림 6]은 저작된 결과물인 TML 파일이다.

#### 2. 예시 콘텐츠 2: 스카이다이버

스카이다이버는 [그림 3]과 같이 인형에 각가속도 센서를 부착한 후, 손으로 인형을 직접 들고 콘텐츠 상의 캐릭터를 조종하는 콘텐츠이다.

위의 요리조리 콘텐츠와 마찬가지로 제작 과정은 진행되었으며, 콘텐츠 제작자는 각가속도 센서가 없는 상태에서 키보드의 화살표 키보드로 작동되도록 콘텐츠를 제작하였다. 콘텐츠가 완성된 후, 저작도구를 사용하여 각가속도에서 획득할 수 있는 3가지 입력 데이터(roll, pitch, yaw)와 다양한 변환 범위를 변경해가면서 테스트 해본 결과 [그림 7]과 같은 상호작용 저작 결과를 얻게 되었다.

```
<?xml version="1.0"?>
<tml>
<sensor type="RFID">
  <button_map id="0101010101" msg="TUL_KEY_SPACE"/>
</sensor>
<sensor type="MOTION">
  <range_map axis="roll_max" value="45"
    msg="TUL_KEY_RIGHT" repeat="true"/>
  <range_map axis="roll_min" value="-45"
    msg="TUL_KEY_LEFT" repeat="true"/>
  <range_map axis="pitch_max" value="45"
    msg="TUL_KEY_DOWN" repeat="true"/>
  <range_map axis="pitch_min" value="-45"
    msg="TUL_KEY_UP" repeat="true"/>
</sensor>
</tml>
```

그림 7. 스카이다이버 콘텐츠의 상호작용 저작 결과

위 그림에서 알 수 있듯이, 최종 선택된 것은 캐릭터의 좌우 이동에 roll 값을, 상하 이동에는 pitch 값을 사용하고 있다. 또한, 각도의 범위는 -45도에서 45도로 하였으며, 각도 값이 범위를 넘은 상태에서는 같은 키보드 이벤트가 계속 발생하도록 하였다.

VI. 결론 및 향후 계획

본 논문은 실감형 사용자 인터페이스를 사용하는 상호작용을 쉽게 디자인하고 구현할 수 있는 저작도구를 설명하였다. 우선, 다양한 장치를 초기화하고 데이터를 수집할 수 있는 실감형 사용자 인터페이스 API를 개발하였으며, 이 API를 이용하여, 각 하드웨어 장치와 콘텐츠간의 이벤트 변환 방법을 정의하는 방식으로 상호작용을 구현할 수 있는 저작도구를 개발하였다.

프로그래밍이나 하드웨어에 대한 사전지식이 없는 콘텐츠 제작자들이 만든 콘텐츠를 저작도구를 통하여 쉽게 실감형 상호작용이 가능하도록 변경할 수 있었으며, 본 논문에서 제시한 예시 콘텐츠의 경우 불과 30여 분 만에 원하는 상호작용을 구현할 수 있었다. 더불어, 콘텐츠 수행 중에도 다양한 상호작용 기법 및 장치를 변경하여 시험해봄으로써 “프로그램 코드 수정, 재캡파일, 콘텐츠 재시작”에 필요한 시간을 절약할 수 있었다.

후행 연구로는 PVoT[11]의 연구에서 볼 수 있는 것과 같이 정형화된 기법으로 상위레벨의 상호작용 저작이 가능하도록 하는 것을 고려할 수 있다. 정형화된 기법이 적용되면, 복잡한 콘텐츠에서 사용되는 많은 상호작용을 모두 개발할 필요 없이, 추상화된 개념의 컴포넌트 단위로[14] 상호작용을 구현 및 재사용할 수 있으며 검증과 디버깅이 수월해질 것으로 기대된다.

참 고 문 헌

[1] 최재희, “유비쿼터스 시대 디자인의 새로운 인터페이스 변화에 관한 연구 -실체적 인터페이스 (Tangible User Interface)를 중심으로”, 한국디자인인문학회지, 제12권, 제1호, pp.59-70, 2006.  
 [2] 최면영, 임창영, “실체적 인터페이스 디자인 시스템에 관한 연구 -탠저블즈의 설계 및 프로토타입 구현을 중심으로”, 디자인학연구, 제17권, 제2호, pp.5-14, 2004.  
 [3] R. M. Taylor II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, “VRPN: a device-independent, network-transparent VR peripheral system,” Proceedings of the ACM

symposium on Virtual reality software and technology, pp.55-61, 2001.  
 [4] <http://www.cs.unc.edu/Research/vrpn/>  
 [5] J. Williams and M. M. Harrison, “A toolset supported approach for designing and testing virtual environment interaction techniques,” International Journal of Human-Computer Studies, Vol.55, No.2, pp.145-165, 2000.  
 [6] K. L. Ko, H. S. Ng, D. M. Grant, and J. Trecat, “Extended Petri net models for fault diagnosis for substation automation,” IEE Proceedings of Generation, Transmission and Distribution, Vol.146, No.3, pp.229-234, 1999.  
 [7] P. Figueroa, M. Green, and H. J. Hoover, “InTml: a description language for VR applications,” Proceedings of the 7th international conference on 3D Web technology, pp.53-58, 2002.  
 [8] <http://carl.kenner.googlepages.com/glovepie>  
 [9] G. A. Lee, C. Nelles, M. Billinghamurst, and G. J. Kim, “Immersive Authoring of Tangible Augmented Reality Applications,” Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.172-181, 2004.  
 [10] A. Cypher, “Watch what I do: programming by demonstration,” MIT Press, 1993.  
 [11] J. Seo, “A structured methodology for virtual reality systems development and interactive support tools,” PhD thesis, Postech, 2005.  
 [12] D. Harel, “STATEMATE: A working environment for the development of complex reactive systems,” IEEE transactions on software engineering, Vol.16, No.4, pp.403-414, 1990.  
 [13] <http://www.hitl.washington.edu/artoolkit/>  
 [14] 이건, 서진석, 김정현, “컴포넌트 기반 프레임워크를 이용한 가상객체의 재사용 기법”, 디지털엔터테인먼트학회논문지, 제1권, 제1호, pp.33-38, 2007.

저자 소개

서진석(JinSeok Seo)

정회원



- 2000년 2월 : 포항공과대학교 컴퓨터공학과(공학석사)
- 2005년 2월 : 포항공과대학교 컴퓨터공학과(공학박사)
- 2005년 9월 ~ 현재 : 동의대학교 게임공학과 조교수

<관심분야> : 저작도구, E-러닝, 기능성 게임

김준호(JunHo Kim)

정회원



- 2000년 2월 : 포항공과대학교 컴퓨터공학과(공학석사)
- 2005년 2월 : 포항공과대학교 컴퓨터공학과(공학박사)
- 2008년 3월 ~ 현재 : 동의대학교 게임공학과 전임강사

<관심분야> : 컴퓨터 그래픽스, 리얼타임 렌더링

권덕중(Duk-Joong Kwon)

준회원



- 2002년 3월 ~ 현재 : 동의대학교 컴퓨터소프트웨어공학과 학사과정

<관심분야> : 컴퓨터 그래픽스

김홍준(HongJoon Kim)

준회원



- 2003년 3월 ~ 현재 : 동의대학교 컴퓨터소프트웨어공학과 학사과정

<관심분야> : 저작도구

오세웅(Sei-Woong Oh)

종신회원

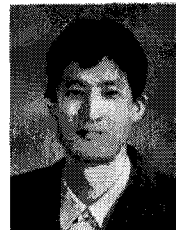


- 1987년 2월 : 한양대학교 전자공학과(공학석사)
- 1998년 2월 : 오사카대학교 정보공학과(공학박사)
- 2004년 9월 ~ 현재 : 동의대학교 게임공학과 부교수

<관심분야> : 네트워크 가상현실, 온라인게임, 유비쿼터스 컴퓨팅

김정현(Gerard JoungHyun Kim)

정회원



- 1989년 2월 : 미 Univ. of Sourthen California(공학석사)
- 1994년 3월 : 미 Univ. of Sourthen California(공학박사)
- 1996년 3월 ~ 2006년 2월 : 포항공과대학교 컴퓨터공학과 조교수/부교수

수/부교수

- 2006년 3월 ~ 현재 : 고려대학교 정보통신대학 부교수

<관심분야> : 가상현실, 혼합현실, HCI, 컴퓨터 음악

김창현(Chang-Hun Kim)

정회원



- 1979년 2월 : 고려대학교 경제학과(학사)
- 1993년 2월 : 일본 Tsukuba대학교(공학박사)
- 1995년 3월 ~ 현재 : 고려대학교 정보통신대학 교수

<관심분야> : 3차원 게임 엔진, Mesh Processing, 3D Shape Reconstruction