



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

석사학위논문

비전문가를 위한 스마트 프로덕트
DIY 도구의 설계



2015년

HANSUNG
UNIVERSITY

한성대학교 대학원

정보통신공학과

정보통신공학전공

이 득 우

석사학위논문
지도교수 조혜경

비전문가를 위한 스마트 프로덕트 DIY 도구의 설계

Designing a Smart Product DIY Tool for Non-Expert
Developers



2014년 12월 일

한성대학교 대학원

정보통신공학과

정보통신공학전공

이 득 우

석사학위논문
지도교수 조혜경

비전문가를 위한 스마트 프로덕트 DIY 도구의 설계

Designing a Smart Product DIY Tool for Non-Expert
Developers

위 논문을 공학 석사학위 논문으로 제출함

2014년 12월 일

한성대학교 대학원

정보통신공학과

정보통신공학전공

이 득 우

이득우의 공학 석사학위논문을 인준함

2014년 12월 일



심사위원장 _____인

심사위원 _____인

심사위원 _____인

국 문 초 록

비전문가를 위한 스마트 프로덕트 DIY 도구의 설계

한성대학교 대학원

정보통신공학과

정보통신공학전공

이 득 우

스마트폰으로 일대 전환기를 맞이한 ICT(Information and Communication Technology)산업은 최근에는 일상생활에서 사용하는 물건에 인터넷을 결합한 스마트 프로덕트(Smart Product)로 진화하고 있다. 그리고 인터넷과 소프트웨어의 발전은 비전문가들도 쉽게 콘텐츠를 제작하는 DIY(Do It Yourself) 문화의 부흥을 가져왔으며, 개인이 만든 창의적인 디지털 콘텐츠는 플랫폼 서비스를 통해 누구든지 세계로 판매할 수 있는 길이 마련되었다.

본 논문은 비전문가가 손쉽게 스마트 프로덕트까지 포괄하는 새로운 개념의 콘텐츠를 제작하게 만들어주는 스마트 프로덕트 DIY 도구에 대한 설계 방식을 담고 있다. 설계는 크게 데스크톱 PC 및 스마트 폰 화면에서 동작하는 멀티미디어와 여러 종류의 하드웨어 제어를 하나의 추상 모델로 일원화시키는 방안, 비전문가들도 손쉽게 콘텐츠를 만들어낼 수 있도록 하는 인터페이스의 설계, 그리고 멀티미디어의 구성요소가 하드웨어의 센서와 액츄에이터와 상호 동작할 수 있도록 하는 인터페이스의 세 부분으로 구분되어 있다. 특히 설계의 검증을 위해 스마트 프로덕트 DIY 도구를 실제로 구현하고, 이의 결과를 정리하였다.

【주요어】 ICT, DIY, 저작 도구, 스마트 프로덕트, 비전문가, 스마트폰, 멀티플랫폼, 멀티미디어, 로봇, 하드웨어, 제어기, 센서, 액츄에이터, 콘텐츠

목 차

I. 서 론	1
1.1 연구의 배경	1
1.2 연구의 목적	3
1.2.1 멀티 플랫폼을 지원하는 스마트 프로덕트 DIY 도구의 설계	3
1.2.2 멀티미디어와 하드웨어의 통합 모델 제시	4
1.2.3 비전문가를 위한 사용자 인터페이스	5
1.3 문제의 정의	7
II. 선행 연구	9
2.1 소프트웨어 기반의 DIY 도구 사례	9
2.1.1 스크래치(Scratch)	9
2.1.2 게임 샐러드(GameSalad)	14
2.2 하드웨어 기반의 DIY 도구 사례	16
2.2.1 레고 마인드 스톰(LEGO MindStorms)	16
2.2.2 로보 플러스(RoboPlus)	19
2.3 시사점과 추가 고려사항	22
III. 핵심 시스템 설계	25
3.1 스마트 프로덕트 DIY 도구의 요구 분석	25
3.2 통합 엔티티 모델의 수립	27
3.3 에디터 인터페이스의 설계	33
3.4 런타임 엔진의 설계	38

IV. 구현 결과	43
4.1 선행 사례와의 비교 분석	43
4.2 멀티 플랫폼 기반의 콘텐츠 편집 및 실행 결과	44
4.3 비전문가를 위한 에디터 인터페이스 구현	47
4.4 스마트 프로덕트 콘텐츠의 제작 및 실행	49
4.5 스마트 프로덕트 DIY 도구의 사용성 평가	51
V. 결 론	52
참 고 문 헌	53
ABSTRACT	57



표 목 차

<표 2-1> 선행 연구를 통해 향후 참고할 시사점들	23
<표 3-1> 액터의 종류와 역할	27
<표 3-2> 프로젝트 구성 요소	33
<표 4-1> 스크래치와 본 연구의 비교	43
<표 4-2> 스마트 프로덕트 DIY 도구에서 지원하는 플랫폼 목록	44
<표 4-3> 멀티 플랫폼 사용자 인터페이스의 규격	45
<표 4-4> 스마트 프로덕트 DIY 도구에 대한 평가 결과	51



그 림 목 차

<그림 1-1> 로봇과 멀티미디어 캐릭터와의 매핑	5
<그림 2-1> 스크래치 프로그램과 구성 요소	10
<그림 2-2> 스크래치의 프로그래밍 인터페이스	10
<그림 2-3> 스크래치 커뮤니티 홈페이지	11
<그림 2-4> 스크래치 프로젝트의 철학	12
<그림 2-5> 게임 샐러드 저작도구 화면	14
<그림 2-6> 게임 샐러드 액터의 구성 요소	15
<그림 2-7> NXT-G 저작도구 화면	16
<그림 2-8> NXG-G 프로그래밍 인터페이스	17
<그림 2-9> 로보플러스 태스크	19
<그림 2-10> 로보플러스 모션 에디터	20
<그림 3-1> DIY 도구의 전체 설계도	26
<그림 3-2> 유형 별 액터의 활용 방안	28
<그림 3-3> 컴포넌트 설계 예시	29
<그림 3-4> 통합 엔티티 모델 클래스 다이어그램	30
<그림 3-5> XML 데이터의 구성	30
<그림 3-6> 사용자 로봇의 XML 데이터 예시	32
<그림 3-7> 룰의 구성	34
<그림 3-8> 조건 설정을 위해 고안한 인터페이스	34
<그림 3-9> 비헤이비어 모델의 구성 요소	35
<그림 3-10> 확장된 최종 엔티티 모델	36
<그림 3-11> 룰 제작을 위해 고안한 사용자 인터페이스	36
<그림 3-12> 스마트 프로덕트 DIY 제작 흐름	37
<그림 3-13> 다수의 룰과 외부 환경의 비교	38
<그림 3-14> 런타임 엔진에서의 룰 실행 로직	39
<그림 3-15> 유니티 통합 인터페이스	40
<그림 3-16> 스마트 디바이스와 하드웨어의 통신	41

<그림 3-17> 이 기종 하드웨어 통신을 위한 메시지 변환	42
<그림 3-18> 로봇 통신 매니저 관련 클래스 다이어그램	42
<그림 4-1> 멀티 플랫폼에서 구동되는 DIY 저작도구	44
<그림 4-2> DIY 도구로 표현된 멀티미디어 콘텐츠 예시	46
<그림 4-3> 액터 조종을 위해 사용자가 제작한 디렉팅 화면	47
<그림 4-4> 콘텐츠 제작을 위한 사용자 인터페이스 흐름	48
<그림 4-5> 사용자가 제작한 콘텐츠를 보여주는 웹 서비스	48
<그림 4-6> 스마트 프로덕트 DIY 콘텐츠 사례	49
<그림 4-7> 최종 소프트웨어 구성	50



I. 서론

1.1 연구의 배경

최근 인터넷을 기반으로 하는 ICT(Information and Communication Technology)산업이 주목을 받고 있다. ICT산업은 대한민국의 새 정부가 선포한 ‘창조경제’의 핵심 근간 기술로서 국가적인 신 성장 산업으로 선정되었으며(문화일보 뉴스, 2013), 2013년에 출간된 유엔 미래보고서 2040(박영숙 외, 2013)에서도, 향후 10년을 지배할 키워드로 모바일 커머스, 소셜 미디어, 빅 데이터, 3D 프린터와 같이 ICT 산업에 관련된 키워드들이 선정될 정도로 미래를 이끌 새로운 산업으로 여겨지고 있다.

ICT 기술의 특징은 플랫폼을 통한 전 세계 사람간의 커뮤니케이션이다. 이는 전문가들의 예상을 뛰어넘어 짧은 시간에 전 세계적으로 큰 파장을 일으킬 수 있는 효과를 만들어주고 있다. 하나의 예로 그 동안 전문가들에게 불가능으로 여겨졌던 해외 음반 시장에 한국어로 만들어진 노래인 강남 스타일이 갑자기 전 세계적인 흥행을 일으키고, 심지어 21억 건의 조회수를 돌파해 32비트의 데이터가 감당할 수 있는 숫자를 넘겨 영상 조회수가 오류나는 재미있는 현상까지 만들어내었다(블로터 뉴스, 2014). 이는 이 전에 어떠한 전문가도 예측하지 못한 특이한 사례였다.

이러한 ICT 기술의 파급력은 개인이 직접 참신한 콘텐츠를 만들어내는 DIY(Do It Yourself)라는 제작 방식이, 이제는 일상생활 영역으로까지 확장되고, 전문가 보다 창의적인 개인의 활동이 앞으로 더 중요해짐을 시사한다. 실제로 대표적인 플랫폼 중 하나인 유튜브(<http://www.youtube.com>)에는 현재 매 분마다 100시간 이상의 분량의 영상이 업로드되고 있으며, 60억 시간 이상의 영상 분량이 시청되고 있다(Youtube Statistics, 2014).

이렇게 스마트폰과 플랫폼 기술로 일상생활에 퍼져있는 ICT 기술은 최근에는 일반 사물에까지 하드웨어를 부착해 인터넷을 연결하는 스마트 프로덕트(Smart Product) 개념으로 점차 영역을 넓혀가고 있다. 기존 기술이 소프트웨

어와 정형화된 하드웨어인 스마트폰 간의 융합이었다면, 앱세서리라고도 불리는 이 스마트 프로덕트 기술은, 소프트웨어와 스마트폰을 포함한 자유롭고 다양한 하드웨어의 융합으로 요약할 수 있는데, 손목 팔찌, 안경과 같은 다양한 형태로 의료, 건강 등의 산업으로 크게 성장할 것으로 예측하고 있는 상황이다(IT동아 뉴스, 2013.). 이러한 일련의 현상을 종합해보면 향후 ICT 기술 발전을 위해서는 스마트 프로덕트의 관점에서 하드웨어와 소프트웨어의 개념을 이해하고 이를 융합할 수 있는 사고를 만들어주는 교육이 필요하고, 저변 확대를 위해 하드웨어와 소프트웨어를 융합하는 스마트 프로덕트 DIY 콘텐츠를 저학년과 비전문가들도 손쉽게 제작하도록 도와주는 도구와 새로운 플랫폼이 필요할 것으로 예측할 수 있다.

본 논문은 스마트 프로덕트까지 아우르는 넓은 개념의 DIY 도구에 대한 설계 방향을 제시하고, 이 기반에서 만들어진 콘텐츠가 비전문가들도 쉽게 사용이 가능한지 검증해보고자 한다.



1.2 연구의 목적

본 연구의 목적은 스마트 프로덕트의 범주에서 소프트웨어와 하드웨어를 통합한 DIY 도구 소프트웨어를 제작하기 위한 설계 모델을 제시하고, 이 모델 위에서 만들어진 콘텐츠가 스마트폰과 PC를 아우르는 멀티 플랫폼 환경에서 동작이 가능한지 검증하는 것이다. 이를 위해 먼저 멀티미디어로 대표할 수 있는 소프트웨어 환경에서의 콘텐츠 구성과 로봇으로 대표되는 하드웨어 환경에서의 콘텐츠 구성을 분석하여, 이 두 환경을 통합할 수 있는 추상 모델을 제시한다. 그리고 각 환경에서 교육을 위해 제공되고 있는 프로그래밍 교육 사례를 살펴보고 비전문가도 쉽게 콘텐츠를 제작할 수 있는 인터페이스를 연구하고, 이에 대한 반응을 보고자 한다.

1.2.1 멀티 플랫폼을 지원하는 스마트 프로덕트 DIY 도구의 설계

기존의 교육용 로봇 기술은 주로 완제품 형태로 제어기와 센서, 액추에이터를 결합한 형태로 주로 판매되었다. 이러한 제작 방식에 업체 간 협의된 표준이 존재하지 않아 주로 다른 제품들과 유사한 마이크로프로세서를 기반으로 제작되고 있지만, 서로 호환되지 않는다는 문제점이 있으며, 최신 ICT 기술에서 가장 이슈가 되고 있는 장치인 스마트폰과의 연동 부분이 미흡한 상황이다.

반면 멀티미디어의 경우에는 응용 소프트웨어 기술이 활성화되고, 다양한 멀티미디어 콘텐츠 저작도구가 나오고 있으며, 이들은 IL(Intermediate Language) 기반으로 다양한 프로그래밍 언어를 사용할 수 있도록 토대를 제공하고, 이를 기반으로 멀티 플랫폼을 지원하며, 편리한 인터페이스와 안정성 높은 기능을 통해 사용자들이 수준 높은 콘텐츠를 만들어낼 수 있도록 편리성을 제공하고 있다. 하지만 멀티미디어 콘텐츠 저작 도구는 주로 반응형 미디어에 속하는 게임 쪽에 치중되어 있어서, 미래의 스마트 프로덕트에 부합하는 하드웨어와의 연동에 대해서는 아직 미흡한 실정이다.

본 논문에서는 이 두 기술의 융합을 위해 나날이 발전하고 있는 멀티미디어 저작도구의 플랫폼을 활용하되, 이의 확장 인터페이스를 이용하여 로봇 하

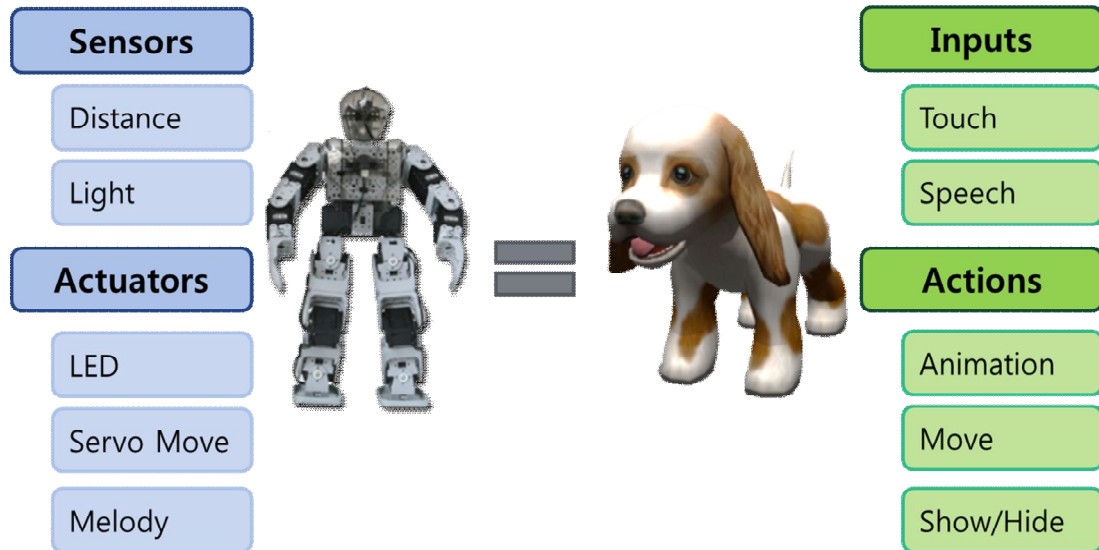
드웨어와의 자유롭게 통신할 수 있는 모델을 제시하고, 또한 흩어져 있는 로봇 규약을 단일 체계로 일원화하는 모델을 만들어 스마트폰에서 다수의 다른 회사 로봇과 통신할 수 있는 토대를 만드는 것을 목적으로 하였다.

1.2.2 멀티미디어와 하드웨어의 통합 모델 제시

멀티미디어에서 콘텐츠를 표현하는 방법은 주로 전체 요소를 배경, 캐릭터 등으로 구분하고 이들의 로직을 구현하는 방법을 사용한다. 여기서 중요한 점은 배경과 캐릭터가 사용자의 프로그래밍 로직에 따라 독자적으로 움직인다는 점이다. 플레이어가 조종하는 주인공 아이, 플레이어를 따라다니는 강아지와 같은 캐릭터들은 시각적인 요소 및 움직임에서 자신의 고유한 성질을 가지고 있지만, 플레이어의 명령에 따라 움직여야 하는 특징을 가진다. 따라서 이를 위해서는 각각 캐릭터에 명령을 내릴 수 있는 프로그래밍 체계가 필요해진다.

하드웨어에서 콘텐츠를 표현하는 방법은 주로 마이크로프로세서를 소유해 독자적인 로직을 수행하는 제어기가 제어기에 부착된 센서와 액추에이터를 조정하는 것으로 표현된다. 일반적인 교구 로봇의 제어기에는 다수의 센서와 액추에이터가 부착될 수 있도록 설계되어 있으며, 추가로 외형을 꾸밀 수 있도록 제조사마다 서로 끼워 맞출 수 있는 다양한 색상과 형태의 블록을 제공한다. 이렇게 블록과 제어기, 센서와 액추에이터를 결합하여 사람의 손동작에 반응하는 강아지 로봇과 같은 하드웨어 콘텐츠가 만들어지게 된다.

이러한 차이로 인해 소프트웨어와 하드웨어 간 융합이 힘들어지게 되는데, 본 논문에서는 이러한 차이를 하나로 통합할 수 있는 추상적인 모델을 제시하고 이를 검증하려고 한다. 본 논문에서 관심을 두었던 것은 두 모델간의 유사성이다. 하드웨어 제어기의 프로그래밍 구조는 특정 센서의 값을 검사하는 조건과 액추에이터를 동작시키는 행동으로 구분되어 있고, 멀티미디어 콘텐츠의 프로그래밍 구조는 사용자 입력 혹은 물체간의 충돌을 감지하는 이벤트와 이 때 특정 작업을 수행하는 비헤이비어로 구성되어 있다.



<그림 1-1> 로봇과 멀티미디어 캐릭터와의 매핑

본 논문에서는 조건과 행동의 범주 내에서 이를 하나의 체계로 통합하는 것이 가능하다고 판단하였고, 추상 모델을 세우고 이를 기반으로 콘텐츠를 만들어 동일한 인터페이스에서 하드웨어와 소프트웨어 제어가 가능한지를 검증하고자 한다.

1.2.3 비전문가를 위한 사용자 인터페이스

본 논문에서는 공통의 모델을 만든 후, 멀티미디어 캐릭터와 하드웨어 제어기를 컨트롤하기 위한 프로그래밍 모델을 어떻게 만들 것인가에 대한 부분도 고려하였다. 지금까지는 텍스트 기반의 프로그래밍 언어를 사용해 멀티미디어 캐릭터와 로봇 제어기를 제어하였는데, 이러한 방식은 초등학생 및 일반인들에게 친숙하지 않고 전문 교육이 필요하다는 문제가 있다.

본 논문에서는 스마트 프로덕트 DIY 도구의 활성화를 위해, 이러한 비전문가들도 쉽게 사용할 수 있는 부분을 고민하였고, 이를 위해 여러 사례들을 살펴보았다. 그래서 초등학생 프로그래밍 교육에 효과적으로 검증된 텍스트가 아닌 특정 메타포를 활용한 비주얼한 프로그래밍 방식을 활용하고자 한다. 비주얼 프로그래밍 방식의 큰 장점은 블록 조립이라는 개념을 사용해 프로그래밍 구문 오류를 미연에 방지해 아이들도 쉽게 프로그래밍에 익숙하도록

만드는 것이다(Mitchel Resnick, 2009).

그리고 쉬운 프로그래밍 방법을 제안하여도 많은 양의 명령어를 사용한다면 사용자의 학습흥미를 방해할 여지가 있기 때문에 프로그래밍 언어는 초보자가 학습하기 쉽도록 최소한의 명령어만 사용하는 전략이 적합하다(박동희, 2009). 특히 하드웨어에 해당하는 로봇의 프로그램은 지속적으로 변화하는 실제 환경 속에서 동작하여 예외 상황이 빈번하게 발생되는데 이를 컴퓨터에서 흔히 사용해오던 관습적인 프로그래밍 방식으로 처리하면 모든 예외상황에 대한 예측과 대응을 수많은 인터럽트를 통해 구현해야하므로 매우 복잡해진다(Nils J. Nilsson, 1994).

본 논문에서는 프로그래밍 방식을 최대한 단순화하고, 최소한의 명령만을 사용하되, 사용자들이 흥미를 가질 수 있는 효과적인 콘텐츠를 구성할 수 있도록 인터페이스를 설계하였고, 초보자들의 창의력을 증진시키는 데 효과적으로 입증된 교육 연극 프로그램(박홍재, 2010)을 손쉽게 만들 수 있는 쪽으로 전체 구성을 잡았다.



1.3 문제의 정의

본 연구는 최신 ICT 트렌드에 맞춰 멀티 플랫폼을 지원하는 콘텐츠 환경과, 하드웨어와 소프트웨어를 일원화시키는 추상 모델의 수립, 그리고 비전문가를 위한 사용자 인터페이스를 제시함으로써 스마트 프로덕트 DIY 콘텐츠를 활성화시키고자 하는데 있다.

이러한 목적을 해결하기 위해서는 다음과 같은 연구의 문제가 고려되어야 한다.

가) 하드웨어와 소프트웨어를 어떻게 추상적으로 일원화 시킬 것인가?

나) 스마트 폰과 PC 환경에서 자유롭게 하드웨어와 통신할 수 있는 환경을 어떻게 구축할 것인가?

다) 다양한 규약으로 배포되고 있는 하드웨어를 어떻게 지원할 것인가?

라) 비전문가도 쉽게 사용할 수 있는 사용자 인터페이스를 어떻게 제공할 것인가?

마) 비전문가도 쉽게 사용할 수 있는 프로그래밍 방식을 어떻게 설계할 것인가?

이를 해결하기 위해 다음과 같은 세 가지 연구 목표를 세운다.

첫 번째, 하드웨어와 소프트웨어를 아우르는 통합 모델을 수립한다. 스마트 프로덕트 DIY 도구에서 추구하는 콘텐츠에서 사용할 수 있는 요소들을 분석하고, 특징 있는 요소들을 분리해, 이들을 독립적으로 관리하는 체계를 세운다.

두 번째, 스마트폰과 하드웨어의 통신 체계를 수립한다. 게임 엔진이라 불리는 소프트웨어의 기반을 활용해 PC와 스마트폰 간의 자유로운 멀티 플랫폼 환경을 구현하고, 여기에 로봇 제어기 하드웨어와 통신할 수 있는 방법을 모색한다. 대부분의 로봇은 리모트 컨트롤 기능(remote control)을 지원하기 위해 블루투스 통신 인터페이스를 갖추고 있는데(김진성, 2012), 이를 활용하여 멀티미디어 콘텐츠와 로봇 콘텐츠가 자연스럽게 통신하는 시스템을 구축한다.

세 번째, 비전문가를 위한 사용자 인터페이스를 고안한다. 비전문가들이 흥

미를 가질 수 있도록 스토리텔링이 가능한 시스템을 기반으로 다양한 콘텐츠 제작을 전개할 수 있도록 기반을 마련한다. 이를 위해 멀티미디어 DIY 도구에서 사용하는 구조를 참고하고, 높은 수준의 멀티미디어 콘텐츠가 재생할 수 있도록 게임 엔진이 제공하는 2D/3D 그래픽, 애니메이션, 물리와 같은 여러 기능을 적극 활용한다. 비전문가들이 어려워하는 텍스트 프로그래밍 방식이 아닌 비주얼 프로그래밍 방식으로 기능을 제공하며, 최소한의 명령어만 사용해도 높은 수준의 콘텐츠가 구성될 수 있도록 다양한 액션을 미리 제공한다. 그리고 사용자가 제작한 콘텐츠를 인터넷상에 저장하고, 이를 다른 기기에서 동일하게 불러와 재생할 수 있도록 표준 포맷으로 모든 정보를 저장하고 공유하는 웹 서비스를 구축한다.



II. 선행 연구

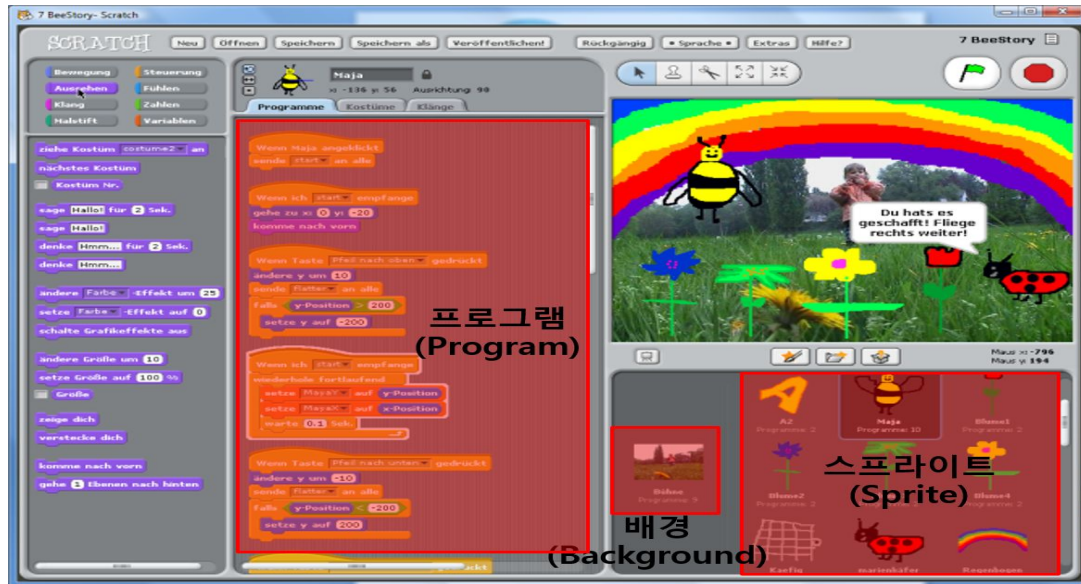
그 동안 비전문가를 대상으로 소프트웨어와 하드웨어의 도메인별로 이의 동작 원리를 이해시켜주는 다양한 시도들이 있어왔다. 이러한 시도들을 분석하여 본 연구에서 지향하는 소프트웨어와 하드웨어가 융합된 스마트 프로덕트 DIY가 나아가야할 방향을 살펴보고자한다

2.1 소프트웨어 기반의 DIY 도구 사례

2.1.1 스크래치(Scratch)

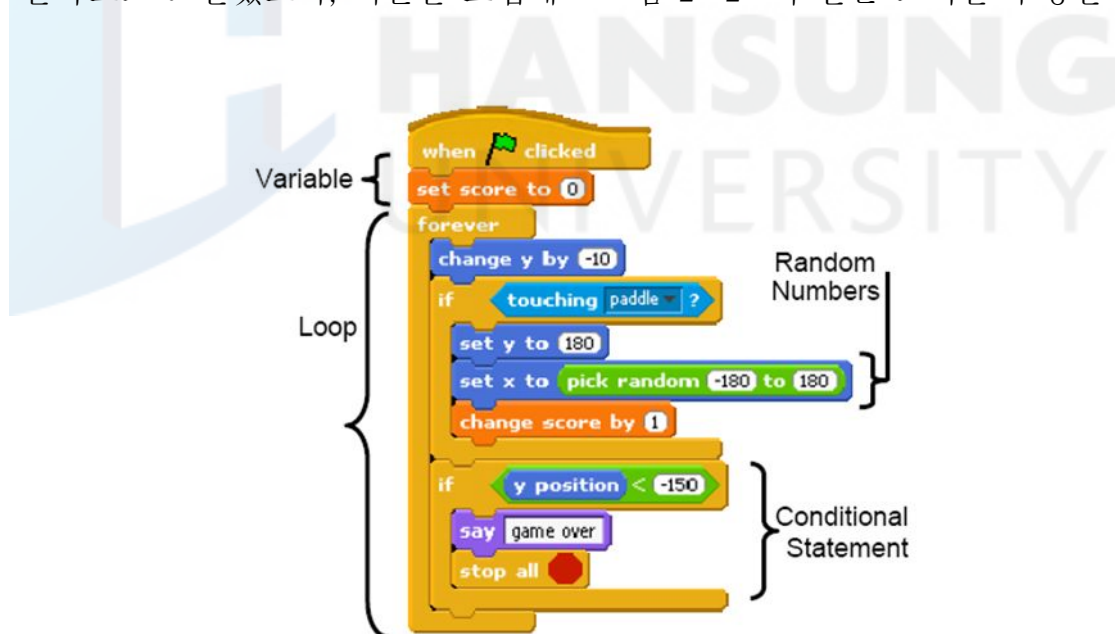
스크래치는 미국 MIT 미디어랩 연구소와 UCLA 대학원 교육정보 연구단이 공동으로 개발한 블록 조립형 멀티미디어 소프트웨어다. (Scratch, 2003) 8세 이상의 어린이들이 스스로 자신만의 반응형 콘텐츠를 만들어 낼 수 있도록 설계되었으며, 특히 디자인과 수학 계산에 대한 아이디어 창출을 장려하고 있다. 스크래치의 특징은 현장에서 언어, 예술, 컴퓨터 영역을 교육할 수 있도록 주제별, 연령별 교육을 제시하고, 실제 수업에서 이를 활용한 결과가 연구, 발표되고 있다. 대한민국에서도 초등학교 교육에 스크래치를 활용하는 사례들이 늘어나고 있는 중이다(서성남, 2011), (함성진, 2011), (문외식, 2013), (블로터 뉴스, 2014.06).

스크래치는 배경(Background)와 스프라이트(Sprite)를 등록해 독자적으로 움직이는 멀티미디어 캐릭터를 지정하고, 각 캐릭터의 로직을 블록으로 프로그래밍 하는 구조를 가지고 있다.



<그림 2-1> 스크래치 프로그램과 구성 요소

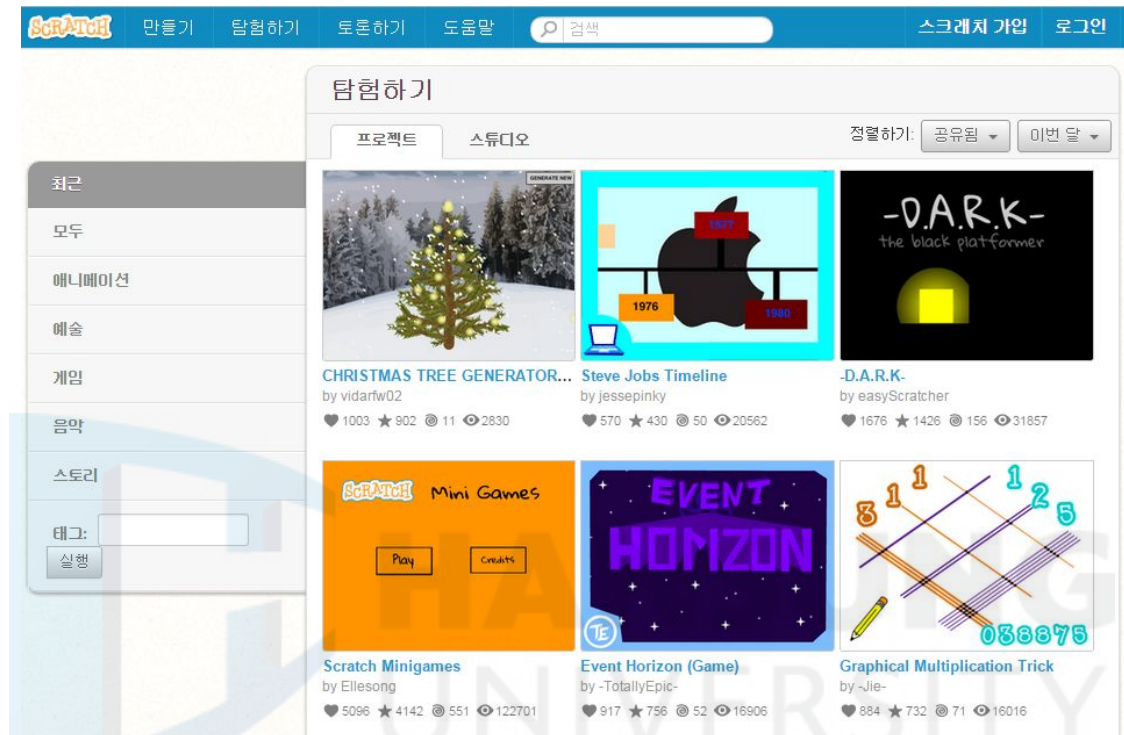
스크래치는 프로그래밍 언어에서 자주 사용되는 요소를 유아들에게 친숙한 블록으로 고안했으며, 이들을 조립해 <그림 2-2>와 같은 로직을 구성한다.



<그림 2-2> 스크래치의 프로그래밍 인터페이스

스크래치의 또 다른 특징은 자체적으로 제작한 커뮤니티 플랫폼을 기반으로 사용자가 만든 콘텐츠를 게시하고, 다른 사용자와 공유하도록 서비스

하는 기능이다(Scratch Explorer, 2014). 이 서비스를 통해 콘텐츠들은 공유되어 다른 사용자가 콘텐츠를 다운로드를 받아 실행하는 것이 가능하며, 다른 사람이 구현한 콘텐츠의 로직을 확인해 공부하고 추가로 수정하는 것이 가능하다.



<그림 2-3> 스크래치 커뮤니티 홈페이지

이 서비스는 스크래치 프로젝트의 철학인 IMAGINE, CREATE, PLAY, SHARE, REFLECT, IMAGE(Mitchel Resnick, 2007)를 실현하기 위해 제작되어졌으며, 2014년 12월까지 약 750만개의 프로젝트가 커뮤니티 서비스에서 공유되고 있는 성과를 거두고 있다.



<그림 2-4> 스크래치 프로젝트의 철학

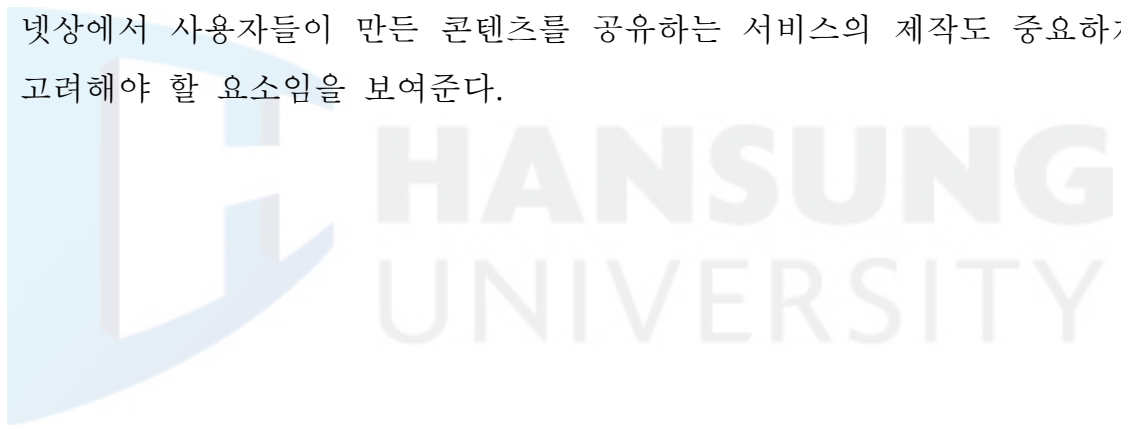
이 외에도 스크래치는 덴마크 LEGO 사와의 제휴를 통해 LEGO WeDo 라는 하드웨어와 결합하여 표현 영역을 멀티미디어에서 하드웨어로 넓힌 것도 또 다른 특징이다. 이를 위해 기존의 멀티미디어 캐릭터 표현에 사용되었던 블록에 WeDo 키트에서 제공하는 하드웨어 모터와 거리 센서의 값을 받아오는 기능을 추가하였다(Scratch WeDo, 2014).

스크래치의 장점은 교육을 위해 설계된 블록 인터페이스이다. 변수, 분기, 반복, Bit 연산과 같이 프로그래밍 언어에서 자주 사용하고 있는 개념을 아동들에게 친숙한 개념인 블록으로 형상화 시켰고, 서로 다른 모양으로 설계해, 잘못된 요소들이 섞이지 않도록 자연스럽게 지도하고 있다. 이 인터페이스를 기반으로 다양한 프로그래밍 교육 교재를 연구하려는 시도 및 유사한 서비스 사례도 등장하게 되었다(김성훈, 2010), (엔트리교육, 2014).

스크래치의 단점을 꼽는다면, 스크래치를 통해 제작된 콘텐츠가 최신 ICT 기술에서 주목받는 안드로이드, iOS와 같은 운영체제 및 디바이스를

지원하지 못하고 있다는 것을 들 수 있다. 그리고 공식적으로는 특정 회사의 제품 하드웨어만 제공하고 있어서 오픈 플랫폼 하드웨어인 아두이노(Arduino)와 연동하는 프로젝트가 별도로 진행되고 있다(S4A, 2014). 그리고 스크래치의 개발 환경이 스킵 스몰토크(Squeak SmallTalk)라는 일반인에게 생소한 언어를 기반으로 제작되었기 때문에, GPL2 라이선싱으로 공개되었음에도 불구하고 일반 개발자들의 참여가 부족한 점도 아쉬운 점으로 들 수 있다(Scratch Licensing, 2014).

스크래치 프로젝트는 일원화된 인터페이스로 하드웨어와 소프트웨어 프로그래밍을 통합했다는 점과, 비전문가도 쉽게 사용할 수 있도록 프로그래밍 인터페이스를 고안하고 이의 효과 검증을 위해 다양한 연구를 수행하는 점에서 본 논문의 연구에 시사 하는 바가 가장 큰 선행 사례라고 할 수 있다. 그리고 비전문가들의 콘텐츠 창작을 활성화하기 위해서는 인터넷상에서 사용자들이 만든 콘텐츠를 공유하는 서비스의 제작도 중요하게 고려해야 할 요소임을 보여준다.



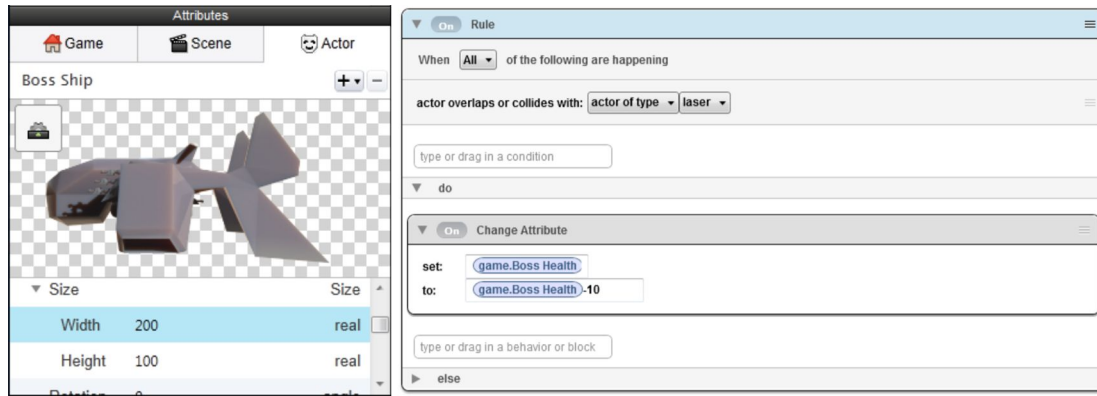
2.1.2 게임 샐러드(Game Salad)

게임샐러드는 게임 콘텐츠 제작을 위해 만들어진 저작 도구다. 스크래치가 저학년 프로그래밍 교육을 목표로 제작되었다면, 게임샐러드는 비전문가들도 퀄리티가 높은 게임을 만들 수 있도록 기획되었고, 실제로 애플 앱스토어 같은 곳에 높은 순위를 기록한 상용 콘텐츠의 사례도 볼 수 있다 (GameSalad Featured, 2014). 게임 샐러드는 씬(Scene)이라는 단위로 콘텐츠 배경을 구성하고 액터(Actor) 라는 단위로 게임에서 플레이하는 캐릭터를 설정한다.



<그림 2-5> 게임 샐러드 저작도구 화면

게임 샐러드가 스크래치와 다른 점은 다양한 어트리뷰트(Attribute) 항목을 제공해 액터의 성질을 보다 세밀하게 컨트롤 할 수 있으며, 미리 게임 제작에 많이 사용되는 비헤이비어(Behavior) 목록을 제공하여 액터의 복잡한 움직임들을 손쉽게 사용자가 구현할 수 있도록 설계하였다.



<그림 2-6> 게임 샐러드 액터의 주요 구성 요소.

게임 샐러드는 스크래치와 다르게 일반인들이 완성도 높은 콘텐츠를 제작 하도록 만드는 것을 목표로 하기 때문에, 세부적인 프로그래밍 로직을 만들어 나가도록 유도하기보다 적절한 비헤이비어를 찾아쓰도록 구성되었다. 그리고 최근 ICT 기술에서 이슈가 되고 있는 안드로이드, iOS, HTML5 플랫폼을 지원한다는 점이 큰 특징이다(GameSalad Creator, 2014). 또한 별도의 Education 버전을 만들어 STEM 교육을 위한 프로그램도 제공하고 있으며, 여러 학교에서 교육 프로그램으로 채택하고 있지만 교육 효과등과 같은 결과 자료는 아직 미흡한 상황이다(GameSalad Education, 2014).

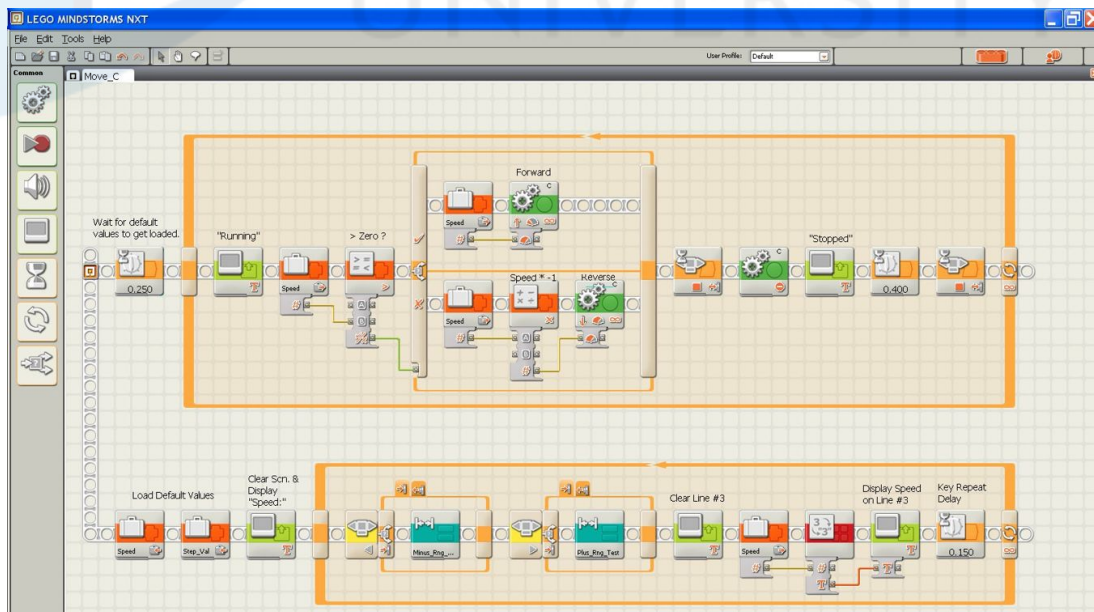
스크래치가 여러 하드웨어와 혼합하는 시도를 하는 반면, 게임 샐러드는 게임 콘텐츠 제작에만 치중되어 있어서 소프트웨어와 하드웨어가 융합하는 새로운 콘셉트의 콘텐츠를 만들 수는 없다는 점이 본 논문에서 개발하고 있는 프로그램과의 차이점이라고 할 수 있다.

게임 샐러드는 쉽게 사용이 가능한 인터페이스로도 높은 수준의 멀티 플랫폼 반응형 멀티미디어 콘텐츠를 만들 수 있다는 것을 증명해 준 사례라는 점에서 본 연구에 시사 하는 바가 큰 사례이다. 손쉬운 인터페이스 설계와 게임 엔진 기술의 적극적인 활용은 비전문가도 창작에 대한 흥미를 가지게 만들고, 스마트폰 마켓에서 흥행하는 높은 수준의 콘텐츠를 만들 수 있음을 실제 사례로 보여주었다는 점에서 본 연구의 전체 구조 설계에 많은 참고가 되었다.

2.2 하드웨어 기반의 DIY 도구 사례

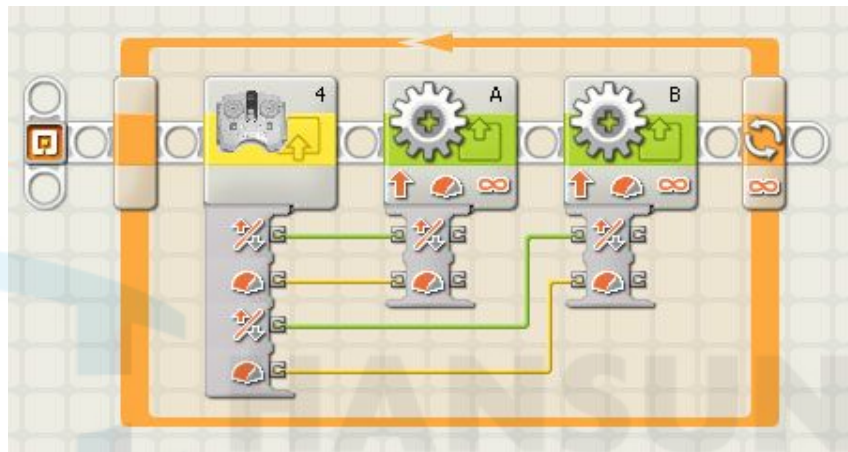
2.2.1 레고 마인드 스톰(LEGO MindStorms)

레고(LEGO)는 일반인에게 블록 장난감 업체로 많이 알려져 있지만, 1998년 전자 제어기와 센서, 모터 등을 결합한 레고 마인드 스톰(LEGO MindStorms) 모델이 출시되면서 하드웨어 교육을 선도하는 업체로도 진출하게 되었고 대한민국에서도 초등학교 교육에 도입되는 등 교육 목적으로 다양하게 활용되고 있다(조이제, 2007), (권이혁, 2010), (김미영 2013), (Lego Mindstorms, 2014). 이후 LEGO RCX, NXT에 이어 EV3 제어기로 꾸준히 업그레이드되었으며, 사용자들이 직접 제어기를 컨트롤 할 수 있도록 Robolab 및 NXT-G와 같은 응용 소프트웨어도 선보이고 있다. 특히 NXT-G와 같은 응용 소프트웨어의 경우 비전문가도 사용할 수 있도록 아이콘 기반의 블록을 제공하며, 순서도를 만드는 형태로 블록을 나열해 로직을 전개하여 제어기를 컨트롤 하는 형태로 구성되어 있다(홍기천, 2009).



<그림 2-7> NXT-G 저작도구 화면

NXT-G에서 제공하는 명령어들은 하드웨어 제어를 위한 크게 센서 제어와 액추에이터 제어를 위한 블록들로 나누어지는데, 텍스트로 표현된 명령어를 전혀 사용하지 않고 아이콘을 사용해 명령어를 표현하게 된다. 이러한 방식을 사용하는 경우 적은 공간에 다양한 명령을 부여할 수 있으나, 복잡한 프로그램을 사용할 수록 점점 가독성이 떨어진다는 단점이 생기게 된다. <그림 2-7>은 특정 센서의 입력을 받아 센서 값에 따라 서로 다른 모터에게 출력을 내리는 예시다.



<그림 2-8> NXG-G 프로그래밍 인터페이스

NXT-G도 스크래치와 유사하게 서로 다른 명령에 대해서는 호환이 되지 않도록 기능이 다른 블록은 서로 맞지 않게 디자인 하여 제공하고 있으며, 이를 통해 비전문가들이 많이 어려워하는 구문 에러를 미연에 방지해주고 있다. 하지만 NXT-G는 하드웨어 제어기를 컨트롤 하는 기능에 특화되어 있기 때문에, 멀티미디어에 관련된 부분은 제어기 액정 화면에 있는 간단한 아이콘만 표시하는 기능만 제공한다. 그리고 오직 NXT 제품군만 호환되며, 프로그래밍 자체에 초점을 맞추기 보다는 제어기를 통해 레고 블록으로 조립된 물체가 제어기를 통해 다이내믹하게 움직일 수 있게 만들어주는 것에 초점을 두고 있다.

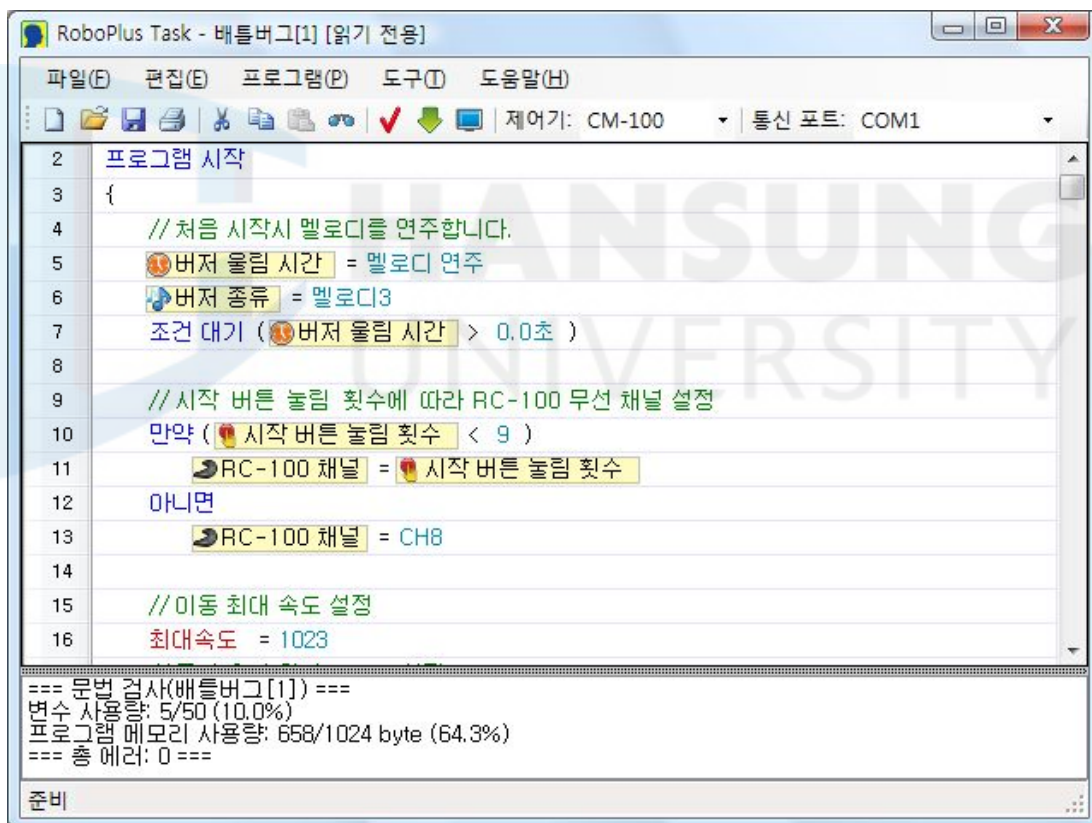
레고 마인드 스톰은 비주얼 프로그래밍의 인터페이스는 비전문가에게 친숙하게 다가가지만 프로그래밍 언어의 제어 구조를 세부적으로 시각화하는 경

우, 오히려 전체적인 구성을 읽고 분석하는 가독성이 떨어질 수 있다는 점에서 본 연구에 시사 하는 바가 크다(Plauska, 2014). 본 연구는 비주얼 프로그래밍 언어의 인터페이스 설계를 프로그래밍의 제어 구조를 그대로 옮기기 보다 비전문가가 쉽게 사용할 수 있도록 간략한 비주얼 프로그래밍 인터페이스 설계를 고안하는 데 중점을 두었다.



2.2.2 로보 플러스(RoboPlus)

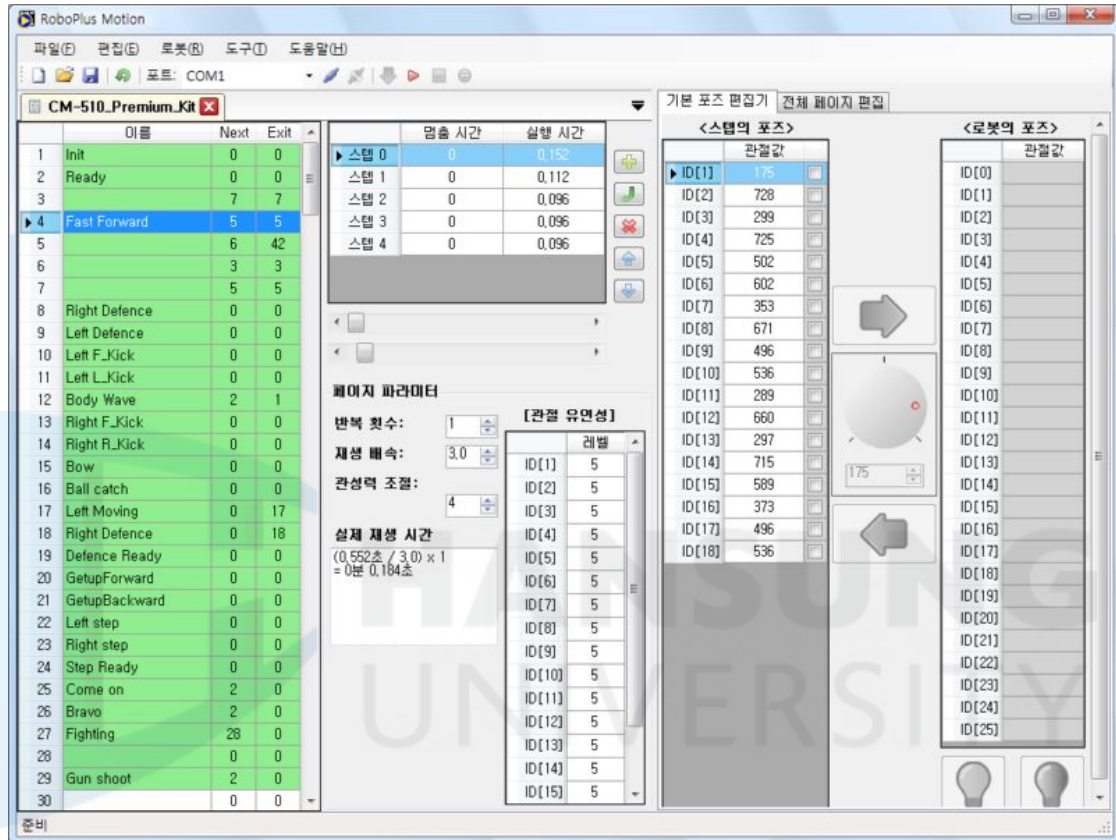
로보 플러스는 한국의 로봇 업체 로보티즈(ROBOTIS)에서 개발한 로봇 전용 프로그래밍 소프트웨어다. 로보 플러스의 특징은 다른 교육용 소프트웨어들이 비주얼한 블록을 메타포로 사용해 구현했다면, 텍스트, 특히 한글을 사용했다는 점이다(로보티즈 로보플러스, 2014). 비주얼한 프로그래밍이 블록의 종류에 따라 다른 슬롯 모양을 사용해 사용자 에러를 미리 방지했다면, 텍스트 프로그래밍의 경우 문법을 검사해야하는 점에서 사용성에 불편함이 있지만, 전체적으로 코드가 만들어졌을 때, 바로 읽을 수 있는 가독성이 뛰어나다는 장점이 있다.



<그림 2-9> 로보플러스 태스크

로보 플러스는 로봇을 컨트롤하기 위해 다양한 프로그램을 제공하고 있는데, 프로그래밍을 통해 제어기를 컨트롤 하는 소프트웨어 ‘태스크(Task)’ 외에

도 로봇을 컨트롤하기 위한 다양한 프로그래밍 소프트웨어를 제공하는데, 다수의 액추에이터를 동기화하여 로봇의 모션을 만드는 ‘모션에디터(MotionEditor)’, 로봇의 각 장치를 점검해주는 ‘매니저(Manager)’ 등이 있다.



<그림 2-10> 로보플러스 모션 에디터

액추에이터를 제어하기 위한 모션 편집은 타임라인 기반의 별도의 프로그램을 제공하여 사용성을 높였다는 점이 특징이다. 다만 프로그램이 다루는 영역이 넓다보니 알아야 하는 명령어가 많고, 텍스트 기반 프로그래밍의 특징인 구문 오류를 해결하는 부분이 있어서 별도의 전문 교육 과정을 거쳐야 사용이 가능하다는 점은 보완이 필요한 점으로 보인다.

로보 플러스는 하나의 프로그램이 초보자용 교육 로봇 키트(OLLO)에서부터 전문가용 로봇 키트(BIOLOID)까지 제어할 수 있도록 설계되어있다는 점에서 본 연구에 시사 하는 바가 크다. 본 연구에서는 로보 플러스의 사례를

바탕으로 초보자용 로봇 키트와 전문가용 로봇 키트 프로그래밍시 고려해야 할 요소들에 대해 파악하였으며, 다양한 종류의 마이크로 컨트롤러에 명령을 전달하기 위해 고려해야 할 데이터 형과 메모리 크기의 다양한 사례를 분석하고 이를 자동으로 변환하는 시스템을 고안하게 되었다.



2.3 시사점과 추가 고려사항

지금까지 소프트웨어 기반의 DIY 도구와 하드웨어 기반의 DIY 도구를 살펴보았다. 이들은 전문가용인가와 초보자 교육용 인가에 따라 소프트웨어 디자인이 크게 달라지는 것을 볼 수 있었다. 스크래치는 8세 이상의 초, 중학생 교육을 목표로 만들어졌기 때문에, 그 연령대에서 관심 가질 수 있는 블록을 디자인하고 프로그래밍 구문 오류로 인한 혼란을 미연에 방지하기 위해 같은 종류의 명령만 결합되도록 디자인하였다. 반면 상용 콘텐츠 생산을 목표로 설계된 게임 샐러드의 사례는 최신 게임 엔진에서 제공하는 미들웨어(Middleware)를 잘 활용한다면 단순한 명령어로도 높은 수준의 콘텐츠를 제작할 수 있다는 가능성을 보여주고 있다. 초급자와 전문가를 모두 아우르도록 설계된 로보폴러스의 경우, 복잡한 프로그래밍 로직을 구현하고, 가독성을 높이기 위해 텍스트 기반 프로그래밍 방식을 채택하였으나, 초급자가 프로그래밍을 잘 못하는 경우 프로그래밍 구문 오류가 발생하고, 이를 해결하기 위해 전문교육과정이 필요한 점이 특징이다. 따라서 비전문가들에게 보다 효과적인 프로그래밍 방식을 구성한다면 텍스트 기반보다는 블록 혹은 아이콘 등의 메타포를 활용한 방식이 보다 효과적임을 유추할 수 있었다.

다만 깊은 프로그래밍 원리에 초점을 두어 세세한 명령어의 동작을 프로그래밍하게 되면 만들 수 있는 결과물에 한계가 발생하게 된다. 이는 스크래치나 NXT-G의 사례에서 볼 수 있듯이 간단한 동작을 실험하고 세부적인 로직의 동작원리를 파악하기 위한 교육적 목적에는 효과적일 수 있지만, 실제로 나오는 콘텐츠의 완성도가 떨어진다는 단점이 있다. 반면 게임 샐러드의 사례는 게임 엔진에서 제공하는 수준 높은 비헤이비어와 어트리뷰트를 정리하여 제공한다면 비전문가들도 높은 완성도의 앱 제작도 가능하다는 점을 시사하고 있다.

스크래치의 문제점은 소프트웨어에서 최종 완성된 콘텐츠가 데스크톱 웹에서만 동작하고, 현재 ICT 기술에서 주목을 받는 스마트 폰에서는 동작하지 않는다는 점이다. 이와 반대로 상용 콘텐츠 생산을 목표로 설계된 게임 샐러드의 사례는 최신 게임 엔진에서 제공하는 미들웨어(Middleware)를 잘 활용

한다면 여러 종류의 스마트폰 및 웹 브라우저에서 동작 가능하다는 가능성을 보여주고 있다. 이러한 점들을 참고하여 본 논문에서는 스마트 프로덕트 DIY 도구 구성을 위해 멀티미디어의 효과 및 범용성을 늘리기 위해 게임 엔진 기술을 적극 활용하였다.

마지막으로 본 논문에서는 멀티미디어로 대변되는 소프트웨어와 제어기로 대변되는 하드웨어의 융합에 대한 부분을 새롭게 해결해야 하는 문제로 삼았다. 이를 구현한 사례는 스크래치뿐인데, 스크래치는 멀티미디어와 하드웨어를 모두 하나의 추상적인 개념에서 아우른다기 보다는 부가 기능과 같은 형태로 제공하고 있다는 점과, 특정 하드웨어에서 동작한다는 점이 개선해야 할 점으로 보인다. 이의 개선을 위해 사용자들이 하드웨어와 소프트웨어의 차이를 느끼지 않고 로직을 구성할 수 있도록 만드는 것을 본 논문에서는 해결해야 할 핵심 문제로 인식하였다.

<표 2-1> 선행 연구를 통해 향후 참고할 시사점들

참고할 시사점	선행 사례
일원화된 인터페이스로 하드웨어와 소프트웨어를 통합하는 모델 수립의 필요성	스크래치
비전문가도 쉽게 사용이 가능한 비주얼 프로그래밍 방식	스크래치 게임샐러드
비전문가도 흥미를 가질 수 있는 콘텐츠 제작 기능	게임샐러드
스마트 기기까지 아우르는 멀티 플랫폼 설계	게임샐러드
다양한 하드웨어 지원을 위한 단일 통신 인터페이스 설계	로보플러스
사용자 간에 콘텐츠를 공유할 수 있는 부가 서비스	스크래치

이러한 선행 사례 외에도 멀티 플랫폼 콘텐츠 개발을 위해서 추가로 고려해야 할 요소들이 있다. 다양한 규격의 스마트 폰과 데스크톱 PC 환경에서 화면과 사용자 인터페이스가 동일하게 보여지도록 논리적인 해상도를 기반으로 GUI 레이아웃이 설계 되어야 하고, 이를 감안해 버튼, 이미지, 레이블과 같은 사용자 인터페이스에 사용할 기본 요소들이 배치되어야 한다(김성수, 2014).

그리고 다양한 하드웨어 지원을 위해 이기종간 데이터 처리를 위한 어댑터를 설계하고, 이 때 기종 별로 사용하는 데이터 타입 정보를 지정하는 별도의 스키마 구조도 고려해야 한다(신재완, 2010). 그리고 스마트폰에서 사용하는 데이터 형식과 로봇에서 사용하는 데이터 형식을 서로 변환할 수 있는 소프트웨어 모듈도 추가로 설계해야 한다(이평화, 2012).



III. 핵심 시스템 설계

3.1 스마트 프로덕트 DIY 도구의 요구 분석

본 연구에서는 기존의 DIY 도구의 장점을 취하고 단점을 보완해 새롭게 스마트 프로덕트 DIY 도구를 고안하고자 한다. 이를 위해 해결해야 하는 요구 사항들을 다음 항목과 같이 정하였다.

- 가) 일원화된 인터페이스로 하드웨어와 소프트웨어를 통합하는 모델 수립
- 나) 비전문가도 쉽게 사용이 가능한 프로그래밍 방식
- 다) 비전문가도 흥미를 가질 수 있는 설정
- 라) 비전문가도 쉽게 사용이 가능한 사용자 인터페이스
- 마) 스마트 기기까지 아우르는 멀티 플랫폼 도구의 제작
- 바) 다양한 종류의 하드웨어 지원을 위한 단일화 된 통신 인터페이스
- 사) 사용자 간에 콘텐츠를 공유할 수 있는 서비스 설계

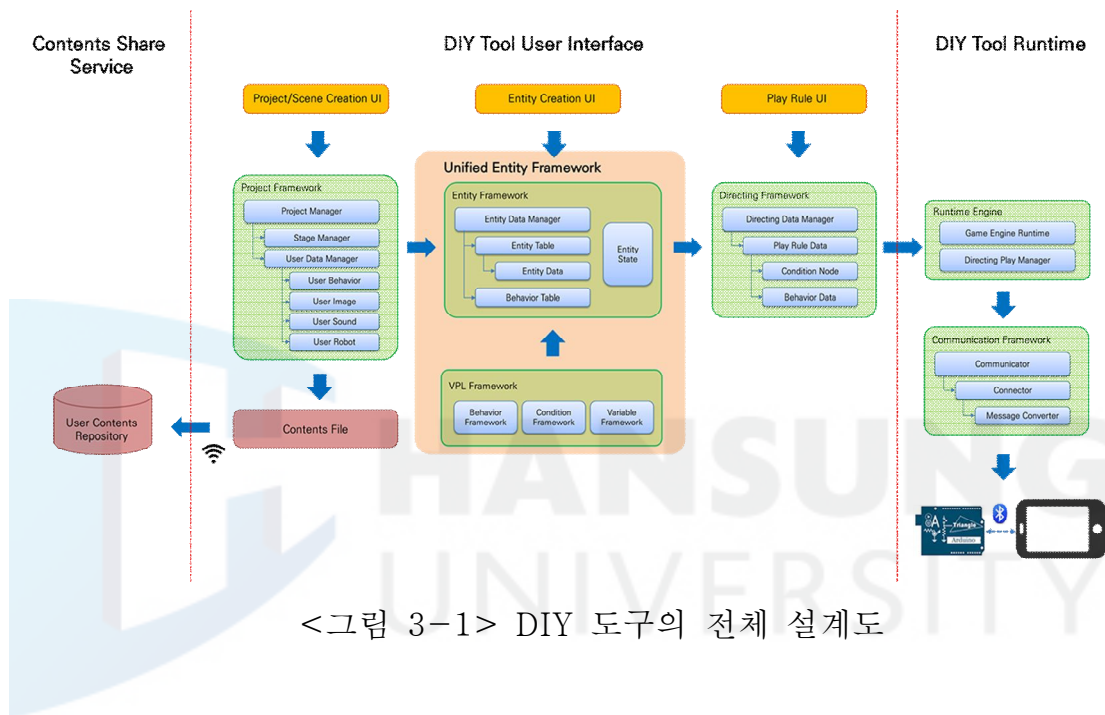
이를 구현하기 위해 본 논문에서는 전체 모델을 통합 엔티티 모델, 사용자 인터페이스, 런타임 엔진의 세 가지 분류로 나누고 분류 별로 독자적인 시스템을 구축하였다.

통합 엔티티 모델에서는 멀티미디어와 하드웨어를 단일 인터페이스 모델로 통합하기 위한 설계 방안을 제시하였다. 그리고 콘텐츠에 관련된 모든 데이터는 XML 형식으로 저장되고 불러들일 수 있도록 직렬화(Serialization)모듈을 추가로 구현하고, 이를 클라우드 서버 시스템과 연동해 인터넷에 연결된 사람들이면 누구나 저장하고 불러들일 수 있도록 서비스까지 구현하였다.

에디터 인터페이스에서는 비전문가도 흥미를 가지고 쉽게 사용할 수 있는 프로젝트 기반의 콘텐츠 설계와, 스마트 디바이스를 고려한 사용자 인터페이스를 고안하고, 기술 소양이 낮아도 엔티티 간의 상호작용 로직을 구현할 수 있도록 조건과 액션이 결합된 룰 기반의 로직 전개 방식을 고안하였다. 그리고 액션의 제작에서 사용자가 복잡함을 느끼지 않도록 두 단계 계층구조로

나누는 설계 방안을 추가로 제시하고 있다.

런타임 엔진에서는 주기적으로 룰을 점검하면서, 조건에 맞는 액션이 실행될 수 있는 모듈을 자체적으로 설계하고, 다 수의 룰이 충돌되는 경우, 이를 해결하기 위한 방안을 제시하고 있다. 또한 런타임에서는 멀티미디어와 다수의 하드웨어 제어가 단일 인터페이스로 통신할 수 있도록 커뮤니케이션 모듈을 탑재하였다. 이를 모두 포괄한 전체 설계도는 <그림 3-1>과 같다.



<그림 3-1> DIY 도구의 전체 설계도

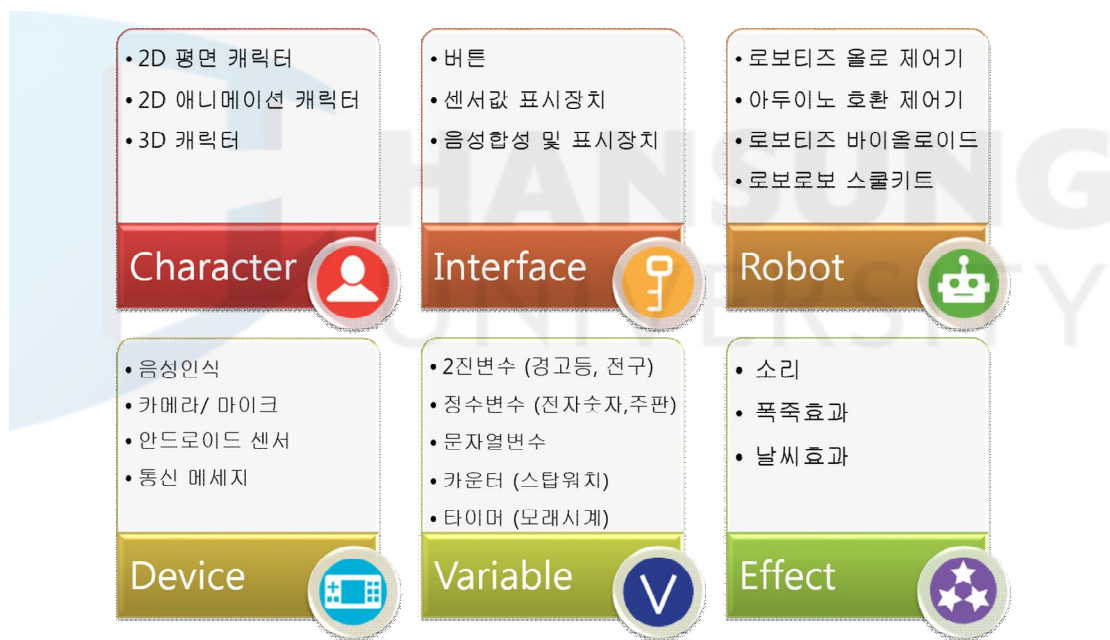
3.2 통합 엔티티 모델의 수립

스마트 프로덕트 DIY 도구를 설계하는데 있어서 가장 먼저 착수한 부분은 멀티미디어 상에서 독립적인 로직으로 움직이는 객체(Object)와 하드웨어에서 독립적인 로직을 수행하는 제어기를 하나의 모델로 일원화 시키는 작업이다. 이를 위해 엔티티(Entity)라는 추상 모델을 정의하고, 엔티티에서 이들이 만들어질 수 있도록 명세서에 해당하는 엔티티 데이터(Entity Data)를 설계하였다. 엔티티 모델로 정의된 가상의 객체는 사용자 인터페이스에서는 액터(Actor)로 표현되며, 스마트 프로덕트 DIY 콘텐츠의 용도에 따라 캐릭터(Character), 인터페이스(Interface), 로봇(Robot), 디바이스(Device), 변수(Variable), 이펙트(Effect)의 <표 3-1>과 같이 여섯 가지로 분류하였다. 또한 스마트 프로덕트 DIY 콘텐츠 내에서 이렇게 구분된 액터의 사용 예시는 <그림 3-2>과 같이 분류가 가능하다.

<표 3-1> 액터의 종류와 역할

액터	역할
캐릭터 (Character)	멀티미디어에서 독립적으로 움직이는 객체를 표현할 때 사용한다. 주로 시각적인 모습을 표현하고 애니메이션을 수행하고, 일정 거리를 이동하거나 물리 시뮬레이션을 수행한다.
인터페이스 (Interface)	사용자의 입력을 받거나 로봇의 현재 상태를 표시하는 용도로 사용된다. 다른 물체들의 상호작용이 없이 독립적인 공간을 보장 받는다.
로봇 (Robot)	하드웨어의 제어기와 제어기에 부착된 센서와 액추에이터가 결합된 하나의 독립 기기를 표현할 때 사용한다. 예를 들어 제어기와 센서, 액추에이터 그리고 추가적인 블록을 사용해 강아지를 표현했다면 이 강아지는 하나의 로봇 엔티티가 된다.
디바이스	멀티미디어와 하드웨어 로봇 이외에 스마트 디바이스에서 제공하는 기능을 활용하기 위해 추가로 제공하는 기능을

(Device)	정의할 때 사용한다. 스마트폰 및 PC에 부착된 음성인식, 카메라, 마이크, 키넥트 등이 범주에 속하며, 디바이스를 이용하여 사용자는 간단한 코딩만으로 높은 수준의 콘텐츠를 구현할 수 있게 된다.
변수 (Variable)	데이터를 저장하는 용도로 사용되며, 룰 기반의 제어 시스템에서 조건을 확인하는 용도로 사용된다. 프로그래밍에서 사용하는 변수와 동일한 기능을 가지고 있다.
이펙트 (Effect)	시각적인 효과를 증대시키기 위해 입자 시뮬레이션에 기반을 둔 멀티미디어 효과를 준다. 간단한 코딩만으로 화려한 멀티미디어 콘텐츠의 구현이 가능하다.



<그림 3-2> 유형 별 액터의 활용 방안

이렇게 분류된 엔티티는 타입에 따라 독자적인 기능을 수행하는데, 본 논문에서는 엔티티의 확장성과 다양성을 표현하기 위해 컴포넌트(Component) 모델을 기반으로 전체 모델 구조를 설계하였다. 컴포넌트 모델은 단일 기능을 컴포넌트라 불리는 의존성이 없는 객체를 생성해 이를 엔티티에 연결하여 구

현하는 설계 방법이다(Scott Bilas, 2002).

의존성이 없는 기능을 분류한 후 이를 각각 컴포넌트로 정리하고, 이를 엔티티에 부착하게 되면 엔티티는 부착된 컴포넌트에 따라 여러 기능들이 수행된다. 예를 들어 멀티미디어 캐릭터인 여자 아이 엔티티는 캐릭터(Character) 엔티티 컴포넌트를 부착한 후, 3차원 공간에 시각적으로 보여주기 위한 Entity3DChar 컴포넌트, 2차원 애니메이션을 위한 Entity2DAnim 컴포넌트, 물리적인 움직임을 위한 Rigidbody 컴포넌트 그리고 캡슐 모양의 충돌 영역 지정에 위한 Capsule Collider 컴포넌트를 추가로 부착하여 완성하게 된다.

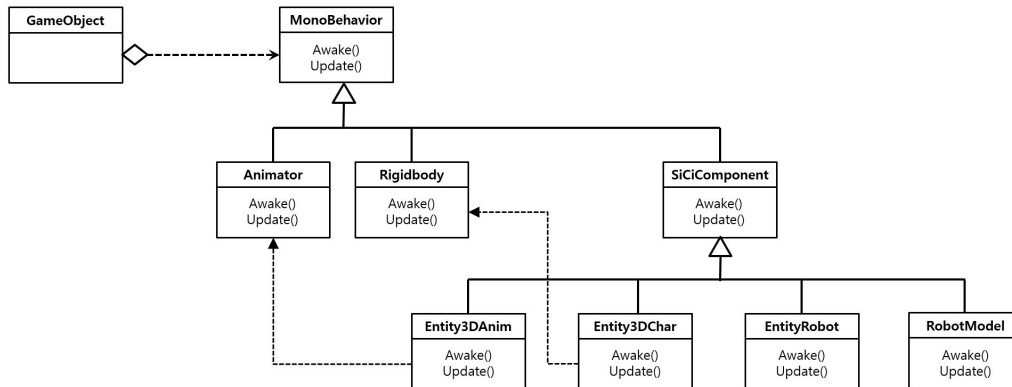
<그림 3-3> 은 여자아이 캐릭터를 표현하기 위해 엔티티와 컴포넌트 정보를 활용한 예시이다. 여기에 향후 캐릭터에 2차원 애니메이션이 아닌 3차원 애니메이션으로 변경해 구현하고 싶다면 Animator라는 컴포넌트를 추가하면 되고, 캐릭터에 물리적인 현상을 없애고 싶다면 Rigidbody 컴포넌트와 Capsule Collider 컴포넌트를 제거하면 된다.



<그림 3-3> 컴포넌트 설계 예시

<그림 3-4>는 GameObject라 불리는 기반 클래스로 기본 객체를 정의한 후 엔티티를 규정하는 클래스를 기반 클래스에 연결하고, 엔티티 별로 지원하

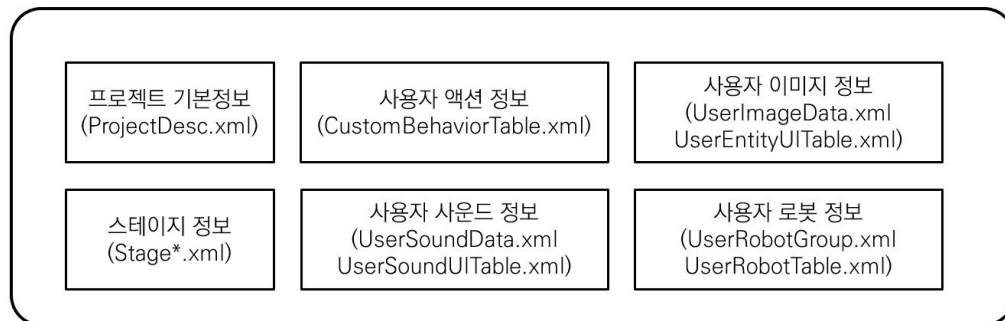
는 다른 컴포넌트를 부착하는 모델의 설계를 나타낸다.



<그림 3-4> 통합 엔티티 모델 클래스 다이어그램

이렇게 설계된 통합 엔티티 모델은 데이터에 의해 관리되도록 직렬화 모듈을 추가로 설계하였다. 직렬화 모듈은 XML 기반으로 만들어졌으며, 각각의 구성요소들은 직렬화 모듈에 따라 저장되고 불러들여진다. <그림 3-5>는 직렬화 모듈이 저장하는 XML 데이터의 전체 스키마를 보여주고 있으며, <그림 3-6>은 이 중 하나인 사용자 로봇 정보의 XML 데이터를 구성한 예시다.

프로젝트 XML 데이터



<그림 3-5> XML 데이터의 구성

```

<?xml version="1.0" encoding="utf-8"?>
<ArrayOfEntityData>
  <EntityData>
    <TypeName>Gatekeeper</TypeName>
    <SortPriority>6</SortPriority>
    <Guid>725313694</Guid>
    <SceneParent>UI</SceneParent>
    <Category>CUSTOM_Robot</Category>
    <CategoryForUI>Robot</CategoryForUI>
    <MaxGenerate>1</MaxGenerate>
    <HideInList>>false</HideInList>
    <CannotPicking>>false</CannotPicking>
    <CannotSelect>>false</CannotSelect>
    <Compnents>
      <EntityComponentData>
        <Name>RobotUCR</Name>
        <OptionParams>
          <OptionParam>
            <Name>Name</Name>
            <Value>Gatekeeper</Value>
          </OptionParam>
          <OptionParam>
            <Name>GUID</Name>
            <Value>725313694</Value>
          </OptionParam>
        </OptionParams>
      </EntityComponentData>
      <EntityComponentData>
        <Name>UcrCommunicator</Name>
        <OptionParams>
          <OptionParam>
            <Name />
            <Value />
          </OptionParam>

```

```
</OptionParams>  
</EntityComponentData>  
</Compnents>  
</EntityData>  
</ArrayOfEntityData>
```

<그림 3-6> 사용자 로봇의 XML 데이터 예시

직렬화 모듈을 통해 DIY 저작도구에서 제작한 콘텐츠의 작업 내용을 파일 시스템에 저장하고 이를 언제든지 불러들여서 콘텐츠를 재생할 수 있는 시스템을 만들고, 이를 HTTP 프로토콜을 사용해 클라우드 스토리지에 저장하도록 확장하면, 콘텐츠를 다른 사용자와 공유할 수 있게 된다.



3.3 에디터 인터페이스의 설계

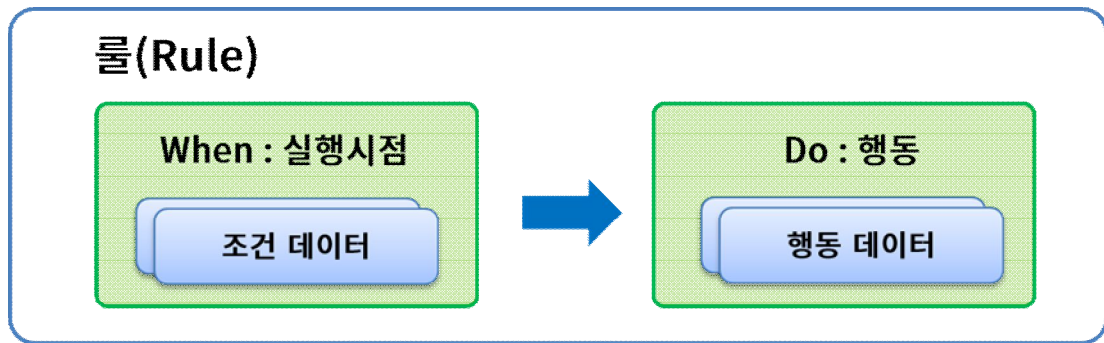
본 연구에서는 초등학생과 일반인들이 손쉽게 사용할 수 있는 인터페이스를 고안해 스마트 프로덕트 DIY 도구에 반영하였다. 이를 위해 가장 먼저 고안한 부분은 전체 콘텐츠의 구성 흐름이다. 비전문가들이 흥미를 가지는데 효과적으로 확인된 스토리텔링 콘텐츠를 사용자들이 쉽게 만들 수 있는데 착안하여 설계 하였다. 이를 위해 스테이지(Stage)라는 개념을 도입하였다. 콘텐츠는 프로젝트(Project)라는 이름으로 사용자들에게 보여지며, 각 프로젝트는 여러 스테이지(Stage)들이 모인 형태로 구성하였다. 프로젝트 단위로 관리되는 콘텐츠가 관리하는 전체 구성 요소들은 <표 3-2>와 같다.

<표 3-2> 프로젝트 구성 요소

구성요소	설명
프로젝트 기본정보	프로젝트의 이름, 장르, 버전, 설명, 생성 시간등의 정보를 저장
사용자 비헤이비어 정보	사용자가 제작한 비헤이비어 편집 정보
사용자 이미지 정보	사용자가 직접 그린 이미지 정보
사용자 사운드 정보	사용자가 콘텐츠 제작을 위해 추가한 사운드 정보
사용자 하드웨어 정보	사용자가 콘텐츠와 연동하기 위해 사용하는 하드웨어 정보
스테이지 정보	스테이지에서 사용하는 배경 등을 기록한 정보
프로젝트 이미지 파일	다른 사용자에게 보여줄 프로젝트 대표 이미지 파일

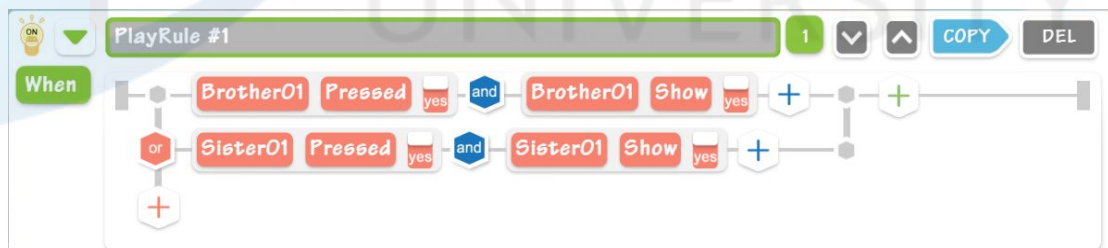
사용자는 각 스테이지에서 사용할 액터를 캐스팅하게 되고, 디렉팅(Directing)을 통해 스테이지에서 액터가 어떻게 움직여야 할지를 지정하게 된다. 하드웨어의 제어나 게임과 같은 멀티미디어에서 동작하는 방법은 대부분 특정 사건(Event)가 발생하면, 사용자가 제작한 로직을 넣어 구현하는 패턴으로 이루어져 있다. 이러한 패턴을 비전문가들도 사용할 수 있도록 언제(When) 어떤 행동(Action)을 한다는 단순한 포맷을 고안하였고 이를 룰

(Rule) 이라고 정했다.



<그림 3-7> 룰의 구성

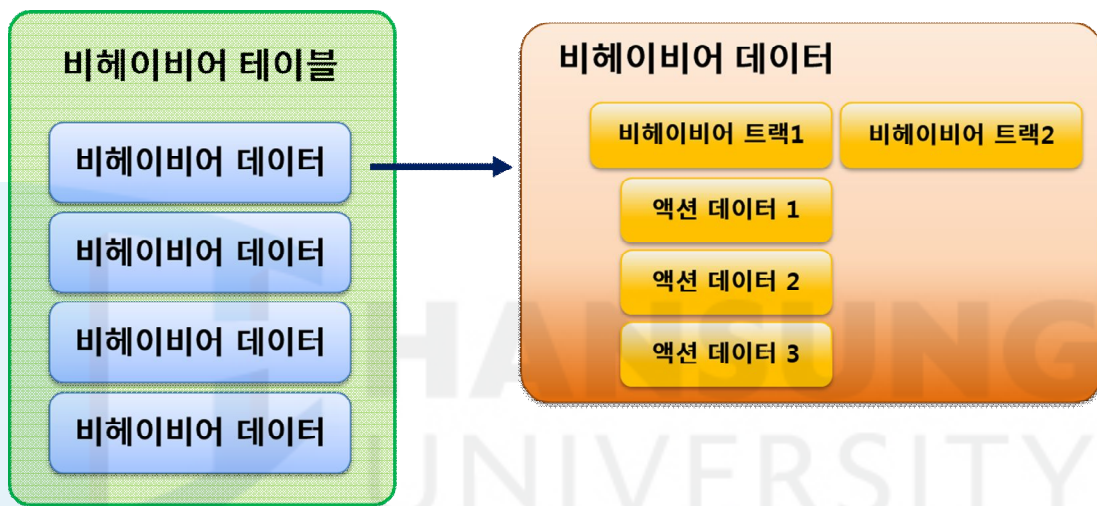
룰은 크게 조건과 행동으로 나뉘는데 조건은 대부분의 사용자의 입력 혹은 액터의 현재 상태, 특정 하드웨어 센서의 값을 비교하는 것으로 구성된다. 또한 여러 가지 조건을 고려할 수 있도록 프로그래밍 언어에서 사용하는 AND와 OR 연산자를 추가로 고려해 복잡한 상황도 고려할 수 있도록 설계하였다. 그리고 AND와 OR의 개념에 대해 생소해할 수 있기 때문에, 가로와 세로의 진행으로 이를 유추할 수 있도록 인터페이스를 고안하였다.



<그림 3-8> 조건 설정을 위해 고안한 인터페이스

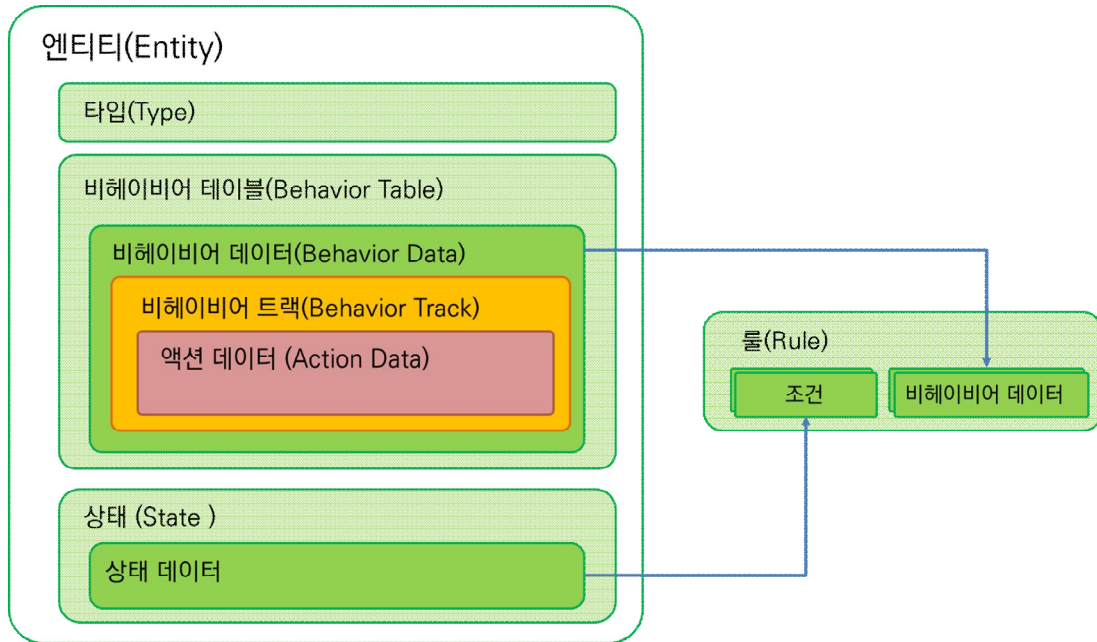
룰의 조건이 완성되면, 각 액터들이 어떻게 동작할지 행동을 지정해야 하는데, 모든 엔티티들이 공통으로 가지고 있는 행동이 있는 반면, 엔티티별로 가지는 다른 행동이 있게 된다. 이렇게 엔티티별로 행동을 구분하기 위해 엔티티 별로 수행 가능한 행동 목록을 담은 비헤이비어 테이블이라는 모델을 추가로 구성하였다. 비헤이비어 설계에서 고안한 부분은 멀티미디어의 애니메이션과 하드웨어의 액추에이터의 움직임이다. 멀티미디어의 경우 스키닝 애니메

이션(Skinning Animation)이라 불리는 특정 타임라인에 위치, 회전, 스케일과 같은 값을 등록하는 키프레임(KeyFrame)을 설정한 후, 키프레임간 데이터를 보간하는 방식으로 자연스러운 움직임을 구성하며, 하드웨어의 액츄에이터의 동작도 특정 타임라인에 액츄에이터의 각도와 기타 상태를 지정하는 형태로 하드웨어의 모션을 구성한다. 이 둘을 포괄하는 모델을 제시하기 위해 비헤이비어 데이터는 타임 라인에 기반을 둔 액션 데이터들로 구성되도록 설계하였다. <그림 3-9>는 이러한 요소를 모두 반영한 확장된 비헤이비어 모델을 정리한 예이다.



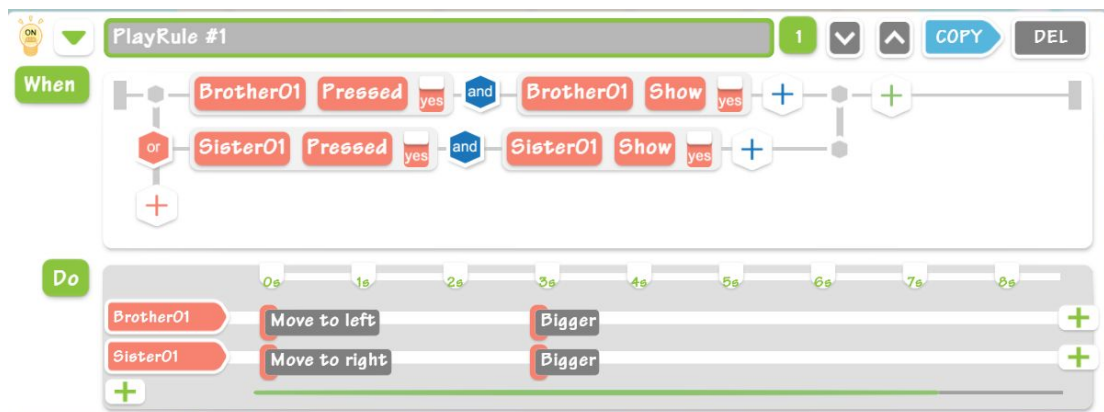
<그림 3-9> 비헤이비어 모델의 구성 요소

조건 모델과 비헤이비어 모델이 완성되면 룰 시스템을 구성할 수 있게 되며, 엔티티의 상태 값을 참조로 하는 조건과 엔티티가 수행할 수 있는 비헤이비어 모델이 결합되어 하나의 룰 시스템이 완성된다. <그림 3-10>는 룰 모델까지 확장한 엔티티 모델을 보여주고 있다.



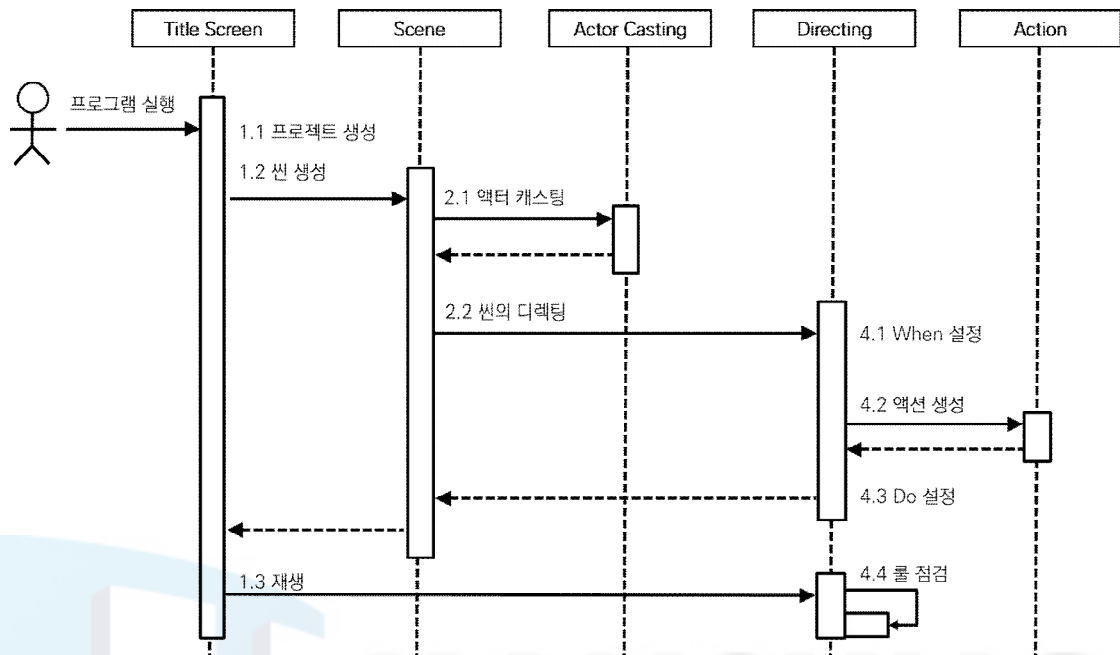
<그림 3-10> 확장된 최종 엔티티 모델

이렇게 설계된 룰의 개념은 사용자가 인식하기 편하게, 조건은 When으로, 비헤이비어는 Do로 표시하도록 인터페이스를 제작하였다. <그림 3-11>은 썸에서 캐스팅된 남자아이와 여자아이중 하나를 클릭하는 경우, 남자아이는 왼쪽으로 이동했다가 3초 후에 커지고, 여자아이는 오른쪽으로 이동했다가 3초 후에 커지는 룰은 아래와 같이 제작하도록 설계하였다.



<그림 3-11> 룰 제작을 위해 고안한 사용자 인터페이스

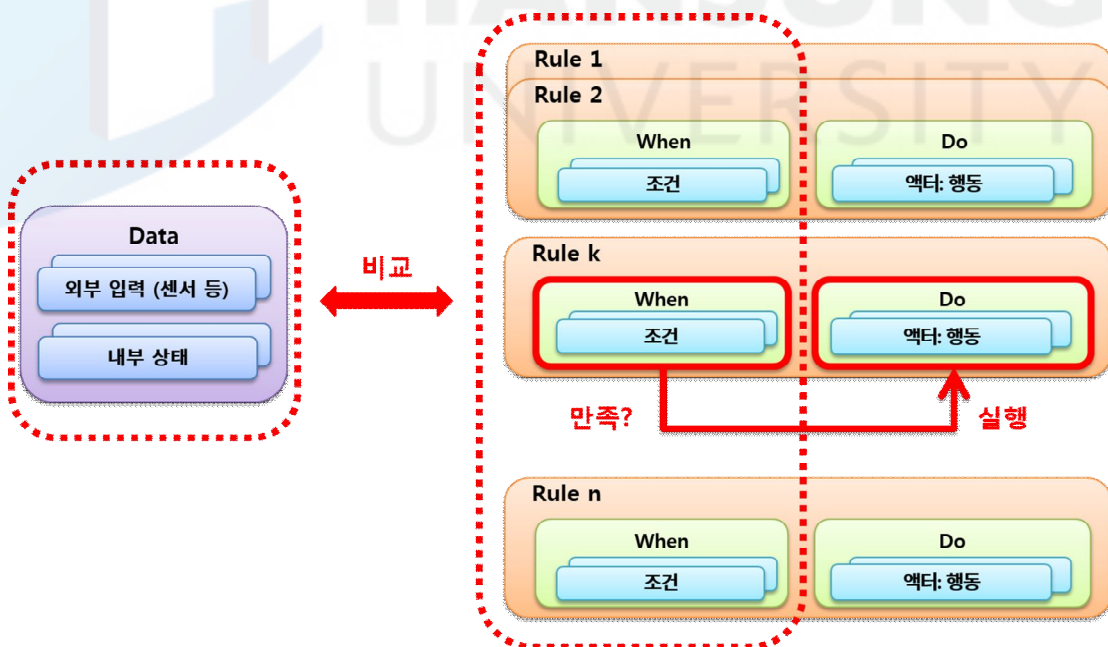
정리하면 사용자는 <그림 3-12>와 같은 흐름으로 스마트 프로젝트 DIY 도구를 이용하여 하나의 콘텐츠를 완성하게 된다.



<그림 3-12> 스마트 프로젝트 DIY 제작 흐름

3.4 런타임 엔진의 설계

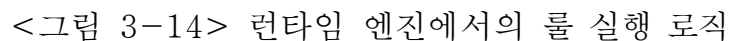
사용자가 콘텐츠를 제작을 완료하면 실행 버튼을 눌러 콘텐츠를 재생하게 되는데, 이를 수행하는 것이 런타임 엔진이다. 런타임 엔진은 지정된 간격에서 사용자가 설정한 룰을 통합 관리하고, 조건에 맞을 경우 이를 수행하는 로직으로 구성되어 있다. 따라서 런타임 엔진은 다수의 룰을 관리하도록 설계되어 있는데, 런타임 엔진 설계시 고려할 점은 다수의 룰에서 조건 맞아 행동을 수행하는 경우, 만일 같은 기능을 사용하는 행동이 중복되는 경우 이를 처리하는 시스템의 고안이다. 예를 들어 A라는 룰의 행동은 캐릭터가 왼쪽으로 움직이는 것이고, B라는 룰의 행동은 캐릭터가 오른쪽으로 움직이는 상황을 가정할 때, A와 B룰이 동시에 만족한다면 최종적으로 캐릭터의 움직임을 어떻게 정할 것인가에 대한 조정이 필요하게 된다. 왼쪽으로 움직인 후에, 오른쪽으로 움직일 것인지, 오른쪽으로 움직인 후에 왼쪽으로 움직일 것인지, 아니면 가만히 있을 것인지로 나누어지게 된다.



<그림 3-13> 다수의 룰과 외부 환경의 비교

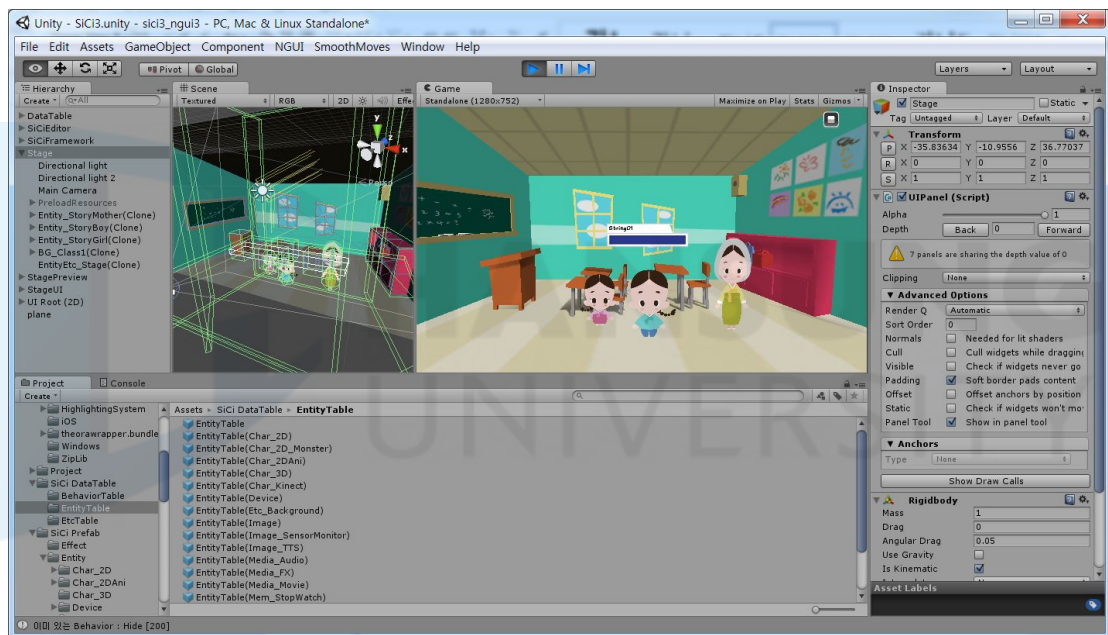
이러한 다양한 경우를 모두 설정하게 만드는 것은 비전문가들에게 어려운

```
graph TD
    Start([Start]) --> D1{현재 조건이 만족된  
다른 룰이 없는가?}
    D1 -- Y --> End([End])
    D1 -- N --> D2{다른 룰과  
액션이 충돌하는가?}
    D2 -- Y --> D3{다른 룰보다  
우선 순위가 높은가?}
    D2 -- N --> D4{충돌처리방식이  
무시인가?}
    D3 -- Y --> End
    D3 -- N --> D4
    D4 -- Y --> End
    D4 -- N --> D2
```



- 39 -

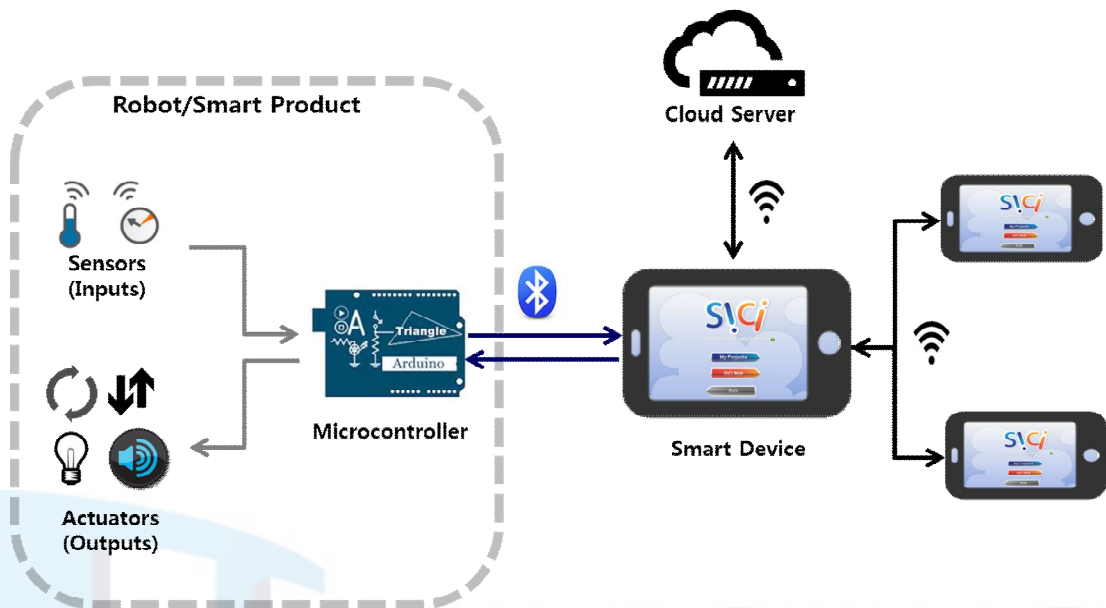
가는 일이다. 본 논문에서는 이를 직접 구현하는 대신, 이를 대신 제공해주는 전문적인 솔루션을 연동하는 방식을 택했다. 이를 위해 유니티(Unity)라는 솔루션을 적극 활용하였다. 유니티는 게임 및 반응형 멀티미디어를 구축할 수 있는 도구로 전 세계 400만 명이 사용하는 범용적인 도구이자 개발 플랫폼이다. 유니티의 가장 큰 특징은 멀티 플랫폼을 지원한다는 데 있다. 유니티에서 결과물을 제작하면 자동으로 윈도우, 맥, iOS, 안드로이드 운영체제 전용의 실행파일로 만들어지는 것이 가능하다 (유니티 지원 플랫폼, 2014. 12.). 그리고 <그림 3-15>과 같이 통합 인터페이스 도구를 제공하고 있어서 본 논문에서의 작업의 생산성을 크게 높일 수 있었다.



<그림 3-15> 유니티 통합 인터페이스

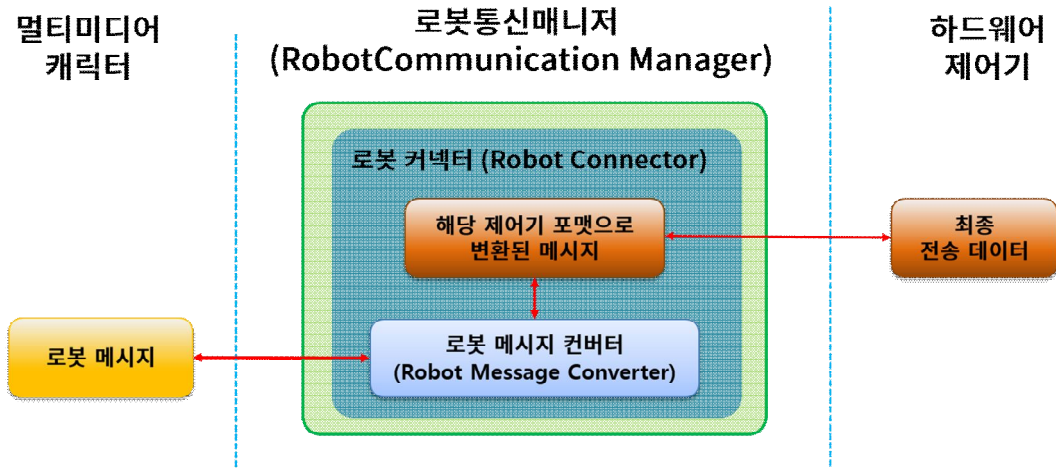
유니티 솔루션의 다른 장점은 확장성이다. 스마트폰에 내장되어 있는 음성 인식, 블루투스 및 자이로 센서를 활용할 수 있으며, 플러그인 시스템을 통해서 키넥트와 같은 부가장치로부터 입력을 받는 것이 가능하다. 그리고 HTTP 통신을 쉽게 쓸 수 있는 라이브러리가 제공되고 있어서 이를 사용해 다른 콘텐츠와 연동하는 것도 가능하다. 본 논문에서는 최근에 나온 대부분의 로봇 제어기들이 블루투스 통신을 지원한다는 점을 활용해 스마트폰과 하드웨어 제

여기 사이의 통신은 블루투스로 구현하였고, 인터넷이 가능한 기기들 간에는 HTTP 통신을 활용해 멀티미디어간 통신을 구현하였다.



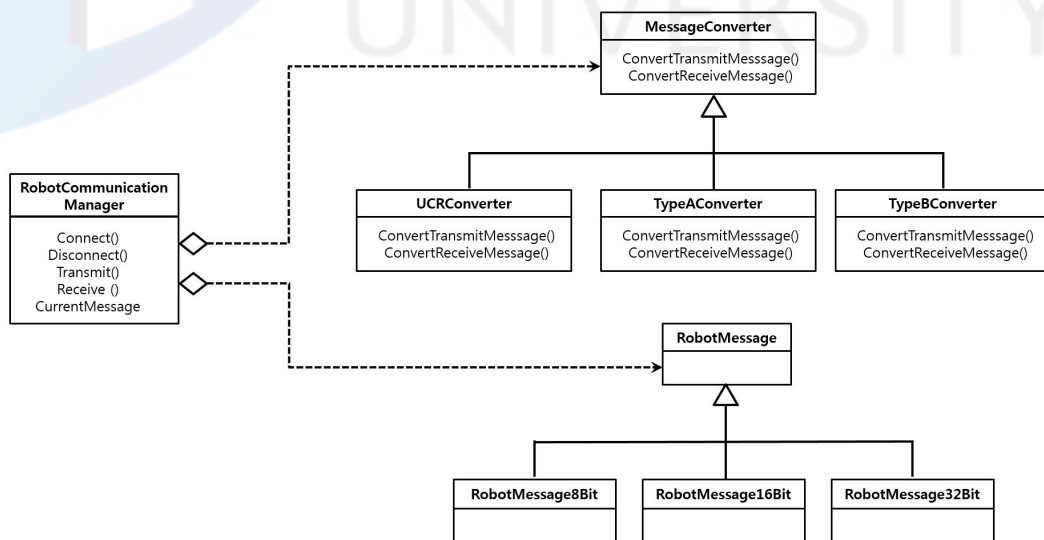
<그림 3-16> 스마트 디바이스와 하드웨어의 통신

이러한 통신 시스템에서 추가적으로 고려할 사항은 하드웨어 제어기와 스마트폰 콘텐츠 사이의 통신 프로토콜의 설계다. 멀티미디어 콘텐츠에서 제어기에 보내는 인터페이스는 일원화하되 실제로 보내는 프로토콜의 데이터는 최종 메시지 컨버터(Message Converter)를 거쳐 제어기가 이해하는 비트스트림(Bit Stream)으로 보내게 된다. 이는 교구용 제어기들의 메모리 용량이 크지 않기 때문에, 제어기에서 이를 구현하게 되면, 부착된 센서와 액추에이터에 내릴 명령의 공간이 부족해지기 때문이다. 따라서 스마트폰에서 하드웨어 제어기와 통신하는 부분을 <그림 3-17>과 같이 설계하였다.



<그림 3-17> 이 기종 하드웨어 통신을 위한 메시지 변환

다수의 로봇과 멀티미디어 사이에서 로봇 메시지를 처리해주는 로봇 통신 매니저는 통신 속도와 로봇의 처리 속도를 감안해 일정 주기 동안 보낼 메시지와 받을 메시지를 큐에 쌓아서 하나씩 처리하는 방식으로 구성되어 있다. 로봇 매니저와 로봇 메시지 그리고 로봇 메시지 컨버터의 구성은 <그림 3-18>와 같이 구성되어 있다.



<그림 3-18> 로봇 통신 매니저 관련 클래스 다이어그램

IV. 구 현 결 과

4.1 선행 사례와의 비교 분석

본 연구의 설계와 사용자 인터페이스로 최종 완성된 스마트 프로덕트 DIY 도구의 평가를 위해 본 연구와 가장 유사한 환경과 목적을 가지고 있는 스크래치와의 비교를 진행하였다.

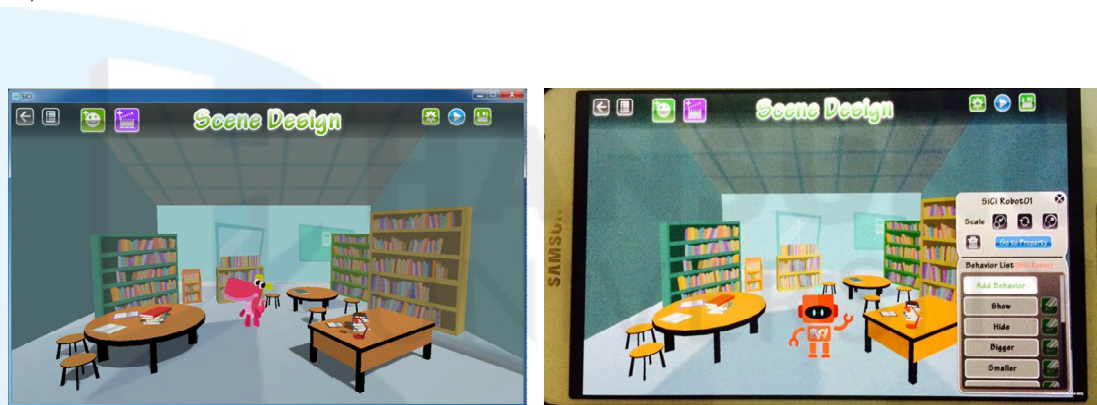
<표 4-1> 스크래치와 본 연구의 비교

항목	스크래치	스마트프로덕트 DIY 도구
목적	소프트웨어(멀티미디어) 창작을 기반으로 한 8세 이상의 프로그래밍 학습 환경	하드웨어(로봇)와 소프트웨어(멀티미디어)를 결합한 창작을 지원하는 비전문가용 콘텐츠 저작 환경
저작환경	데스크톱 환경	데스크톱 및 스마트 디바이스 환경
배경/이미지	2D	2D를 포함한 3D
프로그래밍 추상도 수준	명령어 기반 프로그래밍 프로그래밍 언어의 제어 구조를 그대로 사용	액터에 기반한 표현 규칙과 타임라인을 이용해 프로그래밍과 객체의 개념 학습.
연동 가능한 하드웨어	Pico Board 및 WeDo 지원 비공식적인 별도의 프로젝트로 Arduino도 지원	올로, 바이올로이드, 다윈 Arduino 지원
멀티미디어 애니메이션	여러 장의 스프라이트 이미지를 돌리는 방식으로 구현	캐릭터의 관절 정보를 활용한 스킨닝(Skinning) 애니메이션 구현
고급 기능	영상처리 응용, 증강현실 응용	TTS, 음성인식, 색상인식, 증강현실 응용, 멀티스크린 연동
특장점	프로그래밍 개념을 익힐 수 있는 데 용이하고 다양한 조건, 연산과 순차적 작업 요구에 대한 학습 효과	기술 소양이 낮아도 사용 가능하며, 프로그래밍의 기본 개념을 익힐 수 있음. 다양한 비헤이비어의 제공으로 콘텐츠 제작의 효율성이 높아 사용자의 흥미 유발을 유도

4.2 멀티 플랫폼 기반의 콘텐츠 편집 및 실행 결과

본 연구와 스크래치의 차이점은 스마트 디바이스 환경을 지원하는지에 대한 여부로 볼 수 있다. 이는 스마트폰이 가지고 있는 휴대성과, 스마트폰에 내장된 음성 인식과 같은 고급 기능을 DIY 도구에서 사용할 수 있다는 장점을 가지게 되며, 스마트폰 간의 통신을 통해 멀티 스크린에서 콘텐츠를 연동할 수 있다는 새로운 시도도 가능해진다.

본 연구를 사용해 PC환경과 안드로이드 환경에서 DIY 도구 및 콘텐츠 재생 환경이 동일하게 동작하는 것을 확인할 수 있었다. <그림 4-1>은 각각의 환경에서 재생한 결과 화면이다. 사용자 인터페이스를 포함한 모든 요소들이 해상도가 다른 환경에서 동일한 기능으로 보임을 확인할 수 있다.



PC 버전 구동 화면

안드로이드 버전 구동 화면

<그림 4-1> 멀티 플랫폼에서 구동되는 DIY 저작도구

위의 설계를 통해 실행 가능한 운영 체제는 다음과 같았다. iOS의 경우에는 애플의 블루투스 보안정책으로 인해 허가 없이 사용할 수 없었다.

<표 4-2> 스마트 프로덕트 DIY 도구에서 지원하는 플랫폼 목록

운영체제	실행 장치 (확인된 장비)	콘텐츠 편집 여부	콘텐츠 실행 여부
Windows XP 이상	데스크톱 PC	가능	가능

	노트북		
Android 2.2 이상	삼성 갤럭시 탭 10.1 삼성 갤럭시 노트 10.1 삼성 갤럭시 S 삼성 갤럭시 S2 삼성 갤럭시 S3 삼성 갤럭시 플레이어 LG 옵티머스 G LG 옵티머스 LTE2	가능	가능
iOS 4.0 이상 버전	아이폰 4 아이폰 4S 아이패드 아이패드 2	가능	불가능

또한 본 연구에서는 또한 PC 기반의 마우스 인터페이스와 스마트폰 기반의 터치 인터페이스 간의 일관성 있는 조작법을 제공하기 위해 <표 4-3>와 같이 조작 행위를 정하고, 이를 통일 시켰다.

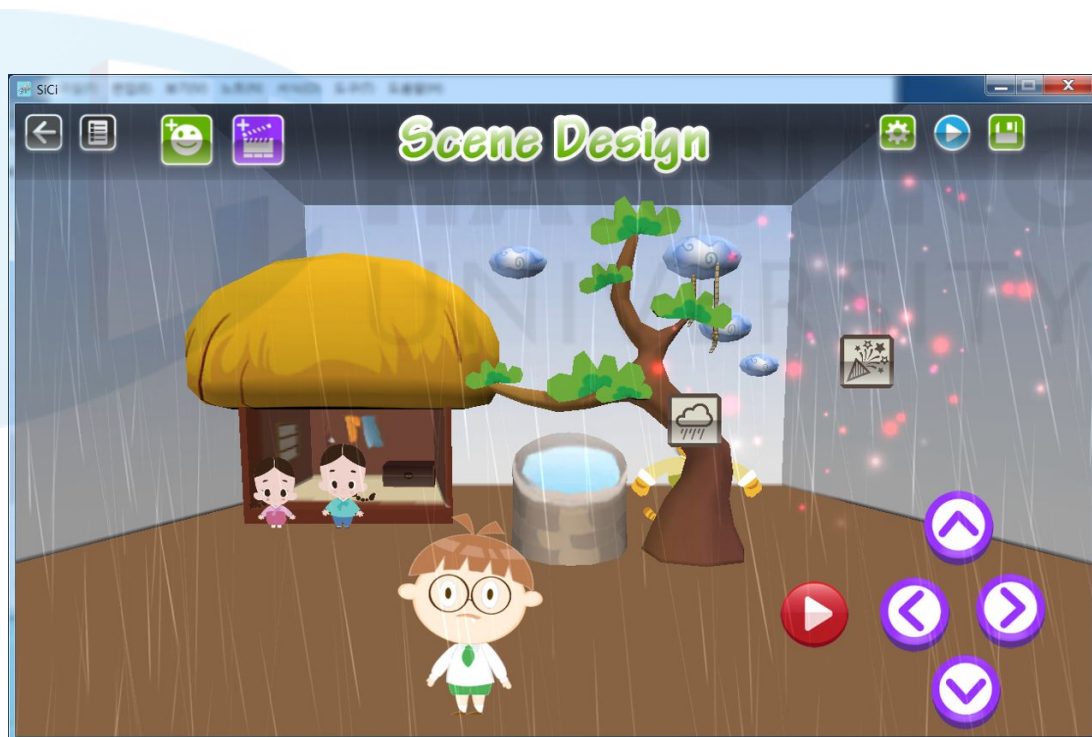
<표 4-3> 멀티 플랫폼 사용자 인터페이스의 규격

조작 행위	마우스 인터페이스	터치 인터페이스
선택	대상 1회 클릭	대상 1회 탭
추가	특정 영역을 1회 클릭	특정 영역을 1회 탭
삭제	특정 영역으로 대상을 드래그 & 드롭	특정 영역으로 대상을 드래그 & 드롭
이동	대상 드래그 & 드롭	대상 드래그 & 드롭
수정	1.5초 클릭하고 있기	1.5초 누르고 있기
편집	슬라이드 바 드래그	슬라이드 바 밀기

본 연구가 스크래치와 다른 차이점을 들 수 있다면 멀티미디어 표현력의 강화다. 스크래치가 예전에 구현하던 2차원 방식의 단순한 이미지 변경의 형태로 멀티미디어를 재생하였다면, 본 연구에서는 게임 엔진이 제공하는 다양한 기능들을 활용해 다음 항목과 같은 다양한 멀티미디어 효과를 사용자가 쉽게 사용할 수 있는 것이 특징이다.

- 가) 3D 공간의 표현
- 나) 3D 캐릭터의 표현
- 다) 물리 시뮬레이션 효과
- 라) 입자 시뮬레이션에 기반을 둔 파티클 이펙트
- 마) 관절 정보를 기반으로 동작하는 스키닝 애니메이션 시스템
- 바) 공간감 있는 3차원 사운드

<그림 4-2>는 3차원 공간에서의 캐릭터 배치, 캐릭터 애니메이션 그리고 주변 환경들과의 충돌 처리, 이펙트 및 사운드 효과와 같은 멀티미디어 요소를 배치한 화면이다. 그리고 UI 버튼을 배치함으로써 캐릭터를 이동시키고 씬을 전환하는 기능이 추가하여 사용자가 DIY도구를 통해 스토리텔링 콘텐츠 제작이 가능하도록 설계되었다.

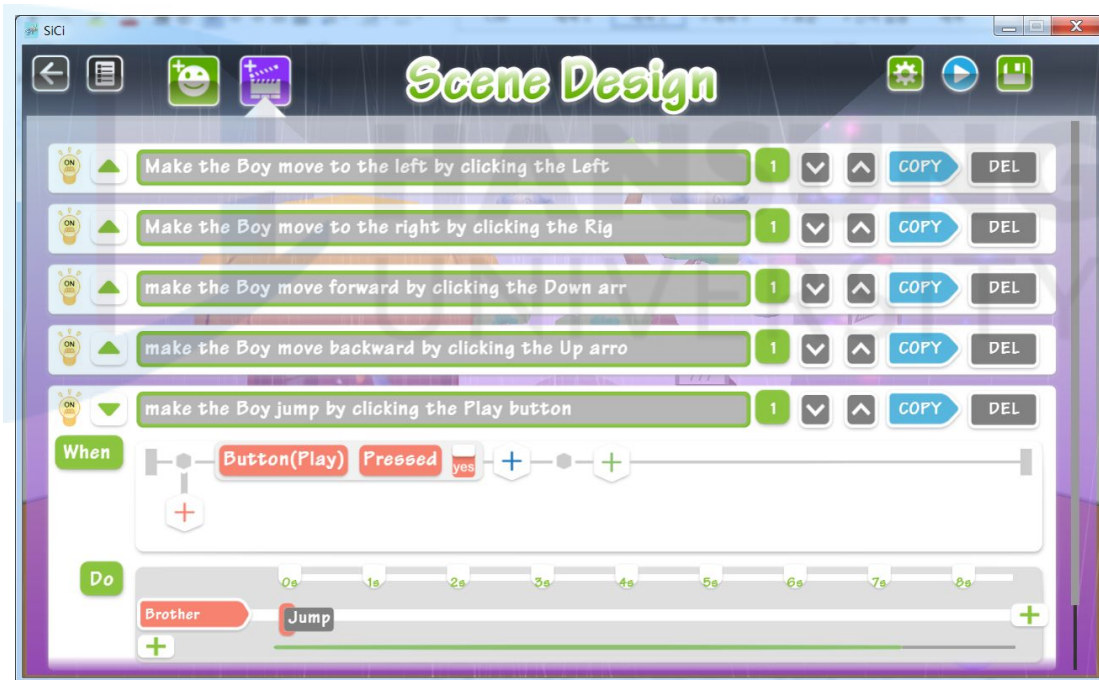


<그림 4-2> DIY 도구로 표현된 멀티미디어 콘텐츠 예시

4.3 비전문가를 위한 에디터 인터페이스 구현

본 논문에서는 프로그래밍 학습의 방법적인 부분에서는 스크래치와는 다른 방향을 취했는데, 스크래치가 일반 프로그래밍 언어가 가지는 제어 구조를 그대로 사용하는 방식을 택하였지만, 보다 편리하도록 언제 무엇을 한다는 컨셉을 기반으로 실행 시기를 결정하는 조건과 원하는 비헤이비어만 선택하면 콘텐츠가 동작할 수 있도록 구성하였다. 따라서 기술 소양이 낮아도 콘텐츠 제작이 가능하며, 프로그래밍 방식을 익히지는 않지만 프로그래밍이 동작하는 개념을 학습하는데 주안점을 두었다.

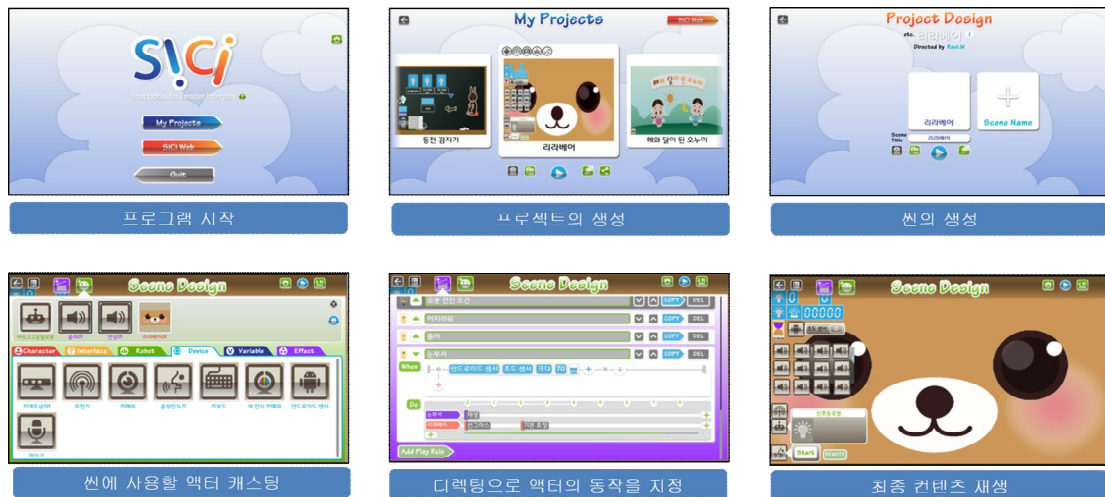
<그림 4-3>은 다수의 룰 설정을 사용해 버튼으로 주인공 캐릭터를 움직이고, 주인공을 점프 시키는 로직을 구성한 예시이다.



<그림 4-3> 액터 조종을 위해 사용자가 제작한 디렉팅 화면

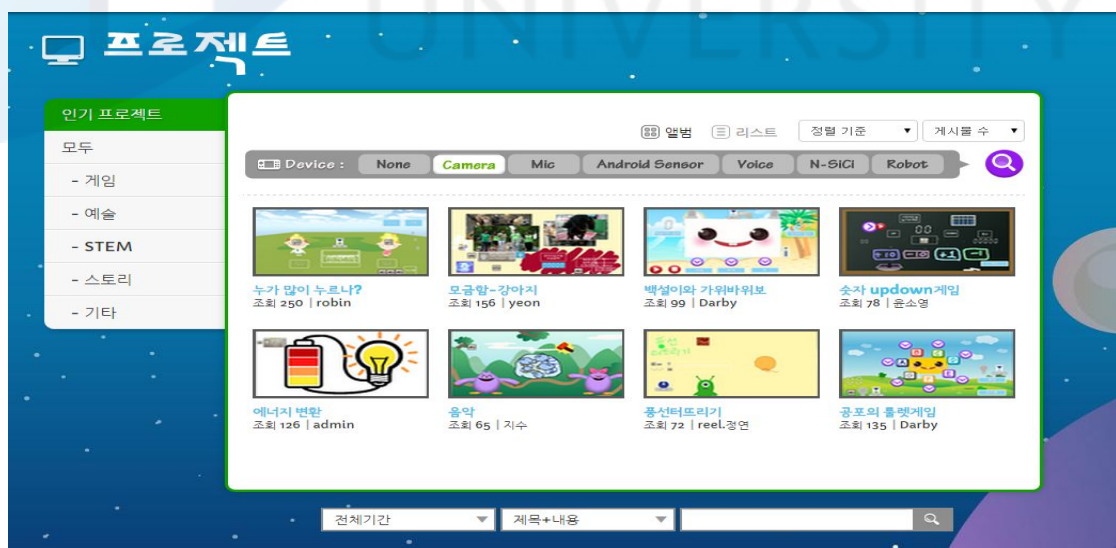
시스템에서는 엔티티, 비헤이비어, 룰의 용어로 설계하였지만, 비전문가들이 이해하기 쉽도록 사용자 인터페이스에서는 씬, 액터, 디렉팅이라는 용어로 변환하고, 인터페이스 흐름을 사용자가 스토리텔링 콘텐츠를 제작하는 프로듀서가 된 느낌을 받도록 최대한 유도하였다. <그림 4-4>는 완성

된 스마트 프로젝트 DIY 의 전체 진행을 나타낸 결과 화면이다.



<그림 4-4> 콘텐츠 제작을 위한 사용자 인터페이스 흐름

이렇게 콘텐츠 제작이 완성되면 XML로 변환하여 웹 서버에 이를 전송해 사용자와 콘텐츠를 공유할 수 있도록 설계하였는데, <그림 4-5>에서 보이는 웹사이트 구축을 통해 이를 검증하였다.



<그림 4-5> 사용자가 제작한 콘텐츠를 보여주는 웹 서비스

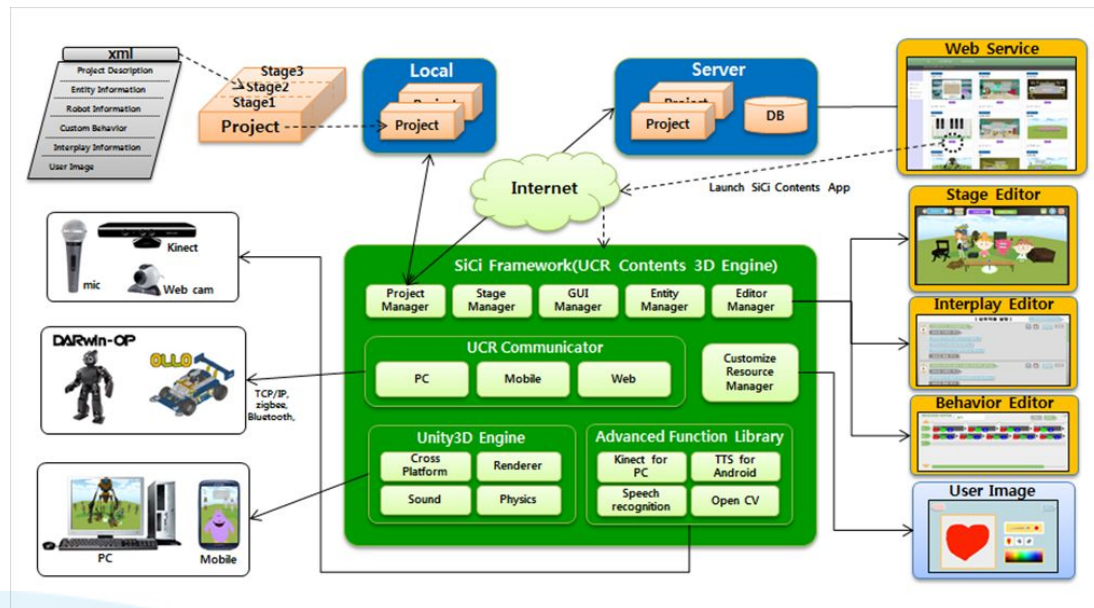
4.4 스마트 프로덕트 콘텐츠의 제작 및 실행

본 연구에서 주안점을 두었던 부분은 센서와 액추에이터를 제공하는 하드웨어와 컴퓨터 또는 스마트 폰을 통합하는 스마트 프로덕트 콘텐츠를 제작할 수 있는 도구의 설계다. 본 연구를 통해 초보자용 교구 로봇부터 전문가용 로봇까지 다양한 로봇 하드웨어와 스마트폰 및 데스크탑 PC에서 재생되는 멀티미디어가 상호 작용을 하는 콘텐츠 제작이 가능함을 검증하게 되었다. <그림 4-6> 은 스마트 프로덕트 DIY 도구를 사용해 제작한 다양한 스마트 프로덕트 콘텐츠의 예시다.



<그림 4-6> 스마트 프로덕트 DIY 콘텐츠 사례

이렇게 제작된 최종 소프트웨어의 구성은 <그림 4-7>과 같다.



<그림 4-7> 최종 소프트웨어 구성

4.5 스마트 프로덕트 DIY 도구의 사용성 평가

스마트 프로덕트의 사용성 검토를 위해 스크래치를 사용해본 경험이 있는 40명의 교사를 대상으로 필드 테스트(Field Test)를 진행해보았고 <표 4-4>와 같은 결과를 받게 되었다.

<표 4-4> 스마트 프로덕트 DIY 도구에 대한 평가 결과

문항	결과
프로그램을 이용하면 학생들의 IT 소양과 흥미를 높일 수 있을 것 같다.	6.67
프로그램을 이용하면 학생들의 집중도를 높일 수 있을 것 같다.	6.51
프로그램에 로봇, 영상, 소리를 융합하면 교과 교육에 유용할 것 같다.	6.37
나는 프로그램 사용에 쉽게 익숙해질 수 있을 것 같다.	6.10
1~2학년 학생들이 프로그램 사용에 쉽게 익숙해질 수 있을 것 같다.	3.67
3~4학년 학생들이 프로그램 사용에 쉽게 익숙해질 수 있을 것 같다.	4.83
5~6학년 학생들이 프로그램 사용에 쉽게 익숙해질 수 있을 것 같다.	6.35
프로그램은 전반적으로 만족스러운 것 같다.	6.49
기회가 되면 학교에서 아이들과 프로그램을 사용해 볼 생각이다.	6.53
이후에 심화된 교육이 있으면 다시 참가하고 싶다.	6.65

설문 결과를 분석해보면 교사들은 프로그램에 대해 전반적으로 만족감을 나타내고 있으며, 소프트웨어와 하드웨어가 융합된 콘텐츠를 만들 수 있다는 점에서 학생들의 집중도를 높이고, 로봇, 영상, 소리를 함께 융합한다는 점에서 IT 소양과 교과 교육을 유용하게 만들 수 있으리라는 의견을 나타내었다. 교사들도 직접 사용할 경우에도 본 연구로 제작된 프로그램이 어렵지 않다는 의견을 주었고, 다만 1학년에서 4학년 초등학생들에게는 어려울 것 같고, 최소한 초등학교 5학년은 되어야 익숙하게 사용할 것이라는 의견을 주었다.

V. 결 론

본 연구는 스마트 프로덕트를 포괄하는 개념에서 소프트웨어 영역에 해당하는 멀티미디어와 하드웨어 영역에 해당하는 제어기 컨트롤러가 하나의 일관된 모델을 기반으로 융합될 수 있을 것인가라는 물음으로부터 출발하였다. 다양한 종류의 스마트 폰 및 데스크톱 컴퓨터 환경에서 동일하게 멀티미디어를 재생하는 데 그치지 않고, 시중에 판매되고 있는 다양한 로봇 하드웨어와 현재 연구 중에 있는 프레임웍까지 포함한 설계를 구상하고, 다양한 소프트웨어 솔루션을 결합하여 스마트 프로덕트의 관점에서 모두를 포괄하는 모델 수립이 가능하다는 것을 실제로 사용가능한 DIY 도구 구현을 통해 증명하였다.

이러한 모델을 기반으로 제작한 DIY 도구를 비전문가들이 사용해 손쉽게 스마트 프로덕트 콘텐츠를 제작할 수 있도록 사용자 인터페이스를 고안했다는 점도 본 연구의 또 다른 주안점인데, 게임 엔진이라 불리는 반응형 멀티미디어 소프트웨어의 기능을 적극 활용하여 비전문가들도 간단한 명령으로 다채로운 멀티미디어 효과를 구현할 수 있도록 제공하고, 로봇 및 멀티미디어 캐릭터의 제어를 위해 일반적인 프로그래밍 언어에서 사용하는 제어 명령을 사용하기보다 When과 Do의 두 단계로 이루어진 상위 개념으로 추상화하여 로직을 제작하는 방식을 고안하였다.

본 연구에서 고안한 사용자 인터페이스는 사용성을 검증하고자 일선 교사들과의 설문을 진행하였다. 설문에 참가한 교사들로부터 좋은 반응을 이끌어 내었고, 1학년부터 4학년까지 저학년들도 쉽게 사용할 수 있도록 툴팁이나 튜토리얼과 같은 사용자 인터페이스를 향후에 추가 보완하면 보다 나은 결과를 낼 수 있을 것으로 기대된다.

참 고 문 헌

1. 국내문헌

- 권이혁. (2010). 『로봇을 이용한 알고리즘 교육이 정보과학영재의 문제해결력에 미치는 영향』. 석사학위논문, 한국교원대학교 교육대학원
- 김미영. (2013). 『로봇을 활용한 융합형(STEAM) PBL 중등 수학 수업에 관한 연구』. 석사학위논문, 서울시립대학교 교육대학원
- 김성수. (2014). 『실무에서의 유니티 3D를 이용한 사용자 인터페이스 개발 연구』. 석사학위논문, 배재대학교 대학원
- 김진성. (2012). 『리액티브 로봇 시스템의 직관적 프로그래밍을 지원하는 스마트 기기 기반의 통합 개발 환경 설계』. 석사학위논문, 한성대학교 대학원
- 박동희. (2009). 『이토이 기반의 교육용 로봇 제어 프로그래밍을 위한 IDE 개발』. 석사학위논문, 고려대학교 대학원
- 박영숙 외, 2013 from 박영숙 제롬 글렌, 테드 고든, 엘리자베스 플로레스큐 (2013). 유엔미래보고서 2040. 서울: 교보문고.
- 박흥재. (2010). 『아동의 창의력 신장을 위한 로봇 교육연극 프로그램 개발 및 적용』. 한국실과교육학회지, Vol.23 No.2 : 159-181
- 신재완. (2010). 『u-GIS DSMS에서 이기종 데이터 처리를 위한 어댑터 설계 및 구현』. 석사학위논문, 인하대학교 대학원
- 이평화. (2012). 『로봇과 안드로이드 장치의 연동을 위한 소프트웨어 프레임 워크』. 석사학위논문, 한양대학교 대학원
- 조이제. (2007). 『초등학생을 위한 로봇 기반의 프로그래밍 학습 시스템 개발』. 석사학위논문, 대구교육대학교 교육대학원
- 홍기천. (2009). 『레고 NXT 로봇을 활용한 예비교사의 프로그래밍 언어 수업 방안』. 한국정보교육학회논문지 Vol.13 No.1 : 71-78

IT동아 뉴스, 2013. from <http://it.donga.com/16957/>

로보티즈 로보플러스, 2014. from <http://support.robotis.com/ko/>
문화일보 뉴스 2013.02 from
<http://www.munhwa.com/news/view.html?no=2013022501070324104002>
블로터 뉴스, 2014. from <http://www.bloter.net/archives/214534>
블로터 뉴스, 2014.06. from <http://www.bloter.net/archives/197687>
엔트리교육, 2014. from <http://play-entry.com/>



2. 국외문헌

Michel Resnick, (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*

Mitchel Resnick, (2009). Scratch: Programming for All. *Communications of the ACM*, Vol. 52 No. 11

Nilsson, Nils J.(1994). Teleo-Reactive Programs for Agent Control. *Journal of Artificial Intelligence Research*, Vol.1 : 139-158

Plauska, 2014. Reflections on Using Robots and Visual Programming Environments for Project-Based Teaching, *Elektronika IR Elektrotechnika*, Vol 20 No. 1

GameSalad Creator, 2014. from <http://gamesalad.com/creator>

GameSalad Education, 2014. from <http://gamesalad.com/education/stories-from-teachers>

GameSalad Featured, 2014. from <http://gamesalad.com/featured-games>

Lego Mindstorms, 2014. from <http://www.lego.com/ko-kr/mindstorms>
S4A, 2014. 12. from <http://s4a.cat/>

Scott Bilas, 2002. A Data-Driven Game Object System, *Game Developer Conference 2002*, from http://scottbilas.com/files/2002/gdc_san_jose/game_objects_slides.pdf

Scratch Explore, 2014. from <http://scratch.mit.edu/explore/>

Scratch Licensing, 2014. from http://wiki.scratch.mit.edu/wiki/Scratch_Source_Code_License

Scratch WeDo, 2014. from http://wiki.scratch.mit.edu/wiki/LEGO%C2%AE_WeDo%E2%84%A6

A2_Construction_Set

Scratch, 2003. from <http://scratch.mit.edu/>

Youtube Statistics, 2014. from

<https://www.youtube.com/yt/press/ko/statistics.html>



ABSTRACT

Designing a Smart Product DIY Tool for Non-Expert Developers

Lee, Deuk Woo

Major in Information & Communication
Engineering

Dept. of Information & Communication
Engineering

The Graduate School

Hansung University

This thesis presents a design and implementation of an authoring tool for non-expert developers which supports realization of their own ideas in the form of unique smart products. The author suggests three key designs of new smart product DIY tool which covers not only multimedia but also things in human life. The first part consists of software design of abstract layer which covers multimedia unit and hardware controller. The second part consists of user interface design to allow non-expert developers can create their own smart product contents easily. The third part consists of communication interface design which allows interaction among multimedia units and hardware controllers. Finally, the author verifies the three key system designs by implementing the actual smart product DIY tool and analyse experimental results.

【keywords】 ICT, DIY, Authoring Tool, Smart Product, Non-expert Developers, Smartphone, Multi-Platform, Multimedia, Robot, Hardware, Controller, Sensor, Actuator, Contents