



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 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](#)

석사 학위 논문

스토리텔링을 이용한 프로그래밍 교육용
Alice 라이브러리 구축

울산대학교 산업대학원
스마트IT융합학과
이세영

스토리텔링을 이용한 프로그래밍 교육용
Alice 라이브러리 구축


지도교수 배재학


이 논문을 석사 학위 논문으로 제출함


2021년 7월

울산대학교 산업대학원
스마트IT융합학과
이세영

이세영의 석사 학위 논문을 인준함

심사위원 전희성 (인) 

심사위원 권영근 (인) 

심사위원 배재학 (인) 

울산대학교 산업대학원

2021년 7월

[국 문 요 약]

스토리텔링을 이용한 프로그래밍 교육용 Alice 라이브리리 구축

울산대학교 산업대학원
스마트IT융합학과
이 세 영

4차 산업혁명 시대에 필요로 하는 코딩능력은 컴퓨팅 사고력을 바탕으로 한 창의력을 요구한다. 코딩이란 복잡한 문제해결의 과정이며 이는 ‘글쓰기’와 닮아 있다. 자신의 생각을 전달함으로써 복잡한 사고활동의 과정 속에서 문제를 해결해 나가는 형태로 진행되는 ‘글쓰기’ 능력이 알고리즘과 유사하기 때문이다. 이는 스토리텔링을 활용한 프로그래밍 교육이 필요한 이유이다. 이 논문에서는 수많은 교육용 프로그래밍 언어 중에서 스토리텔링에 최적화된 프로그램인 Alice를 소개한다. 학습자가 Alice 프로그램을 활용하여 애니메이션 및 게임을 제작함으로써 자연스럽게 컴퓨팅 사고력과 창의력을 기를 수 있을 것이다. 체계적으로 스토리텔링을 구현하기 위하여 플롯유닛(Plot Units)을 기반으로 스토리를 분석하는 방법을 제시하였다. Plot Units 중 Primitive Plot Units 예제를 바탕으로 등장인물 사이에 일어나는 상호작용을 Affect States와 Casual Links로 분해하였다. 이를 바탕으로 엘리스에서 등장인물의 마음상태를 애니메이션으로 표현할 수 있도록 시나리오와 함께 제시하였다. 다만 Alice에서 미세한 움직임 표현하기 위해 소요되는 시간과 시행착오를 줄이기 위해 중복되는 동작이나 자주 사용하는 패턴을 프로시저로 작성하였다. 더

불어 Primitive Plot Units의 예제를 기반으로 작성된 프로시저 모음들을 엘리스 클래스 파일로 저장하여 라이브러리화 하였다. 학습자는 이 라이브러리를 이용하여 프로시저를 재사용함으로써 등장인물간의 상호작용과 마음상태를 쉽게 표현할 수 있을 뿐 아니라 의도한 결과를 빠르게 얻을 수 있을 것이다.

주요어 : 엘리스3, Plot Units, 스토리 텔링, 프로그래밍 교육, 블록형 언어, 교육용 프로그래밍 언어, 메타버스, 코딩 교육, 엘리스 라이브러리

[목 차]

[국문요약]	i
[목 차]	iii
[표 목 차]	iv
[그림목차]	v
I. 서론	1
1. 연구의 필요성 및 배경	1
2. 연구의 목적	2
3. 논문의 구성	4
II. 관련연구	5
1. 소프트웨어 교육	5
2. 소프트웨어 교육도구	6
가. 메타버스	6
나. 교육용 프로그래밍 언어	12
3. 이야기 구성	18
III. 스토리텔링 적용	19
1. 스토리텔링의 도구	19
가. Alice로 구현하기	21
나. Primitive Plot Units	33
2. 앨리스 라이브러리 구축 및 활용	46
가. 프로시저 작성하기	46
나. 클래스 저장/불러오기	56
IV. 결론 및 향후 연구	61
[참고문헌]	63
[Abstract]	65

[표 목차]

표 1 스토리텔링 도구 : 엘리스와 스크래치 비교	20
표 2 엘리스 갤러리에 정의된 클래스	24

[그림 목차]

그림 1 https://www.roblox.com/create	6
그림 2 로블록스 스튜디오	7
그림 3 https://www.roblox.com/games#/	7
그림 4 https://education.roblox.com	8
그림 5 https://store.unity.com/kr/lego-microgame	9
그림 6 유니티 스토어 - 레고 마이크로게임	9
그림 7 튜토리얼 : 유니티 스토어 - 레고 마이크로게임	10
그림 8 파이썬 코딩을 위한 마인크래프트 서버 구동	11
그림 9 컴퓨터에 설치된 마인크래프트 서버	11
그림 10 마인크래프트 자바 버전 메인 페이지	12
그림 11 스크래치	13
그림 12 엔트리	14
그림 13 엔트리 파이썬	14
그림 14 엘리스 IDE	15
그림 15 코두(Kodu)	16
그림 16 코두 - 아이콘(타일)형식 블록을 사용하여 코딩	16
그림 17 앱인벤터 - 화면 디자인(Designer) 페이지	17
그림 18 앱인벤터 - 블록코딩(Blocks) 페이지	17
그림 19 엘리스 시작 화면 - 배경 선택하기	22
그림 20 엘리스 초기 화면	22
그림 21 Scene Setup 초기 화면	23
그림 22 엘리스 갤러리	24
그림 23 Setup Scene 화면 - Biped Classes	25
그림 24 Biped Adult Person 속성 - outfit 탭 : 의복 속성	26

그림 25 Biped Adult Person 속성 - top/bottom 탭 : 의복 속성	26
그림 26 Biped Adult Person 속성 - hair/hat 탭 : 머리스타일 속성	27
그림 27 Biped Adult Person 속성 - face 탭 : 얼굴형태 속성	27
그림 28 Biped Adult Person 이름 및 속성 저장	28
그림 29 Scene Setup - New Adult 객체 생성	28
그림 30 Biped - Adult 객체의 구성 (Joints)	29
그림 31 getLeftShoulder의 Turn 프로시저를 이용한 왼팔의 움직임 설정	30
그림 32 왼팔 움직임 동작	30
그림 33 프로시저 설정이 잘못되어 자연스럽게 못한 경우	30
그림 34 getLeftClavicle의 Roll 프로시저를 이용한 팔 올리기	31
그림 35 Fnger/FingerKnucle을 이용한 손가락 제어	32
그림 36 myFirstMethod 탭	32
그림 37 스토리보드 스케치	34
그림 38 Success Plot Unit	35
그림 39 Problem Plot Unit	35
그림 40 Primitive Plot Units	37
그림 41 엘리스 기본 프로시저 : say	47
그림 42 Edit Code 버튼	48
그림 43 Editing Area	48
그림 44 Biped 프로시저 만들기	49
그림 45 프로시저 이름 정하기	49
그림 46 Biped 프로시저 탭 - Code Editor	50
그림 47 프로시저 'success_ask_for' 코드	51
그림 48 myFirstMethod에서 사용 가능한 사용자 지정 프로시저	51
그림 49 success_ask_for 실행 화면	52
그림 50 프로시저 'Jump_up' 코드	53

그림 51 Jump_up 프로시저 실행화면	54
그림 52 problem_shakingHead 프로시저 코드	54
그림 53 problem_shakingHead 프로시저 실행화면	55
그림 54 Complex Positive Event : walk_hug 프로시저	55
그림 55 walk_hug 프로시저 코드	56
그림 56 프로시저를 클래스 파일로 저장하기	57
그림 57 프로시저 클래스 불러오기	58
그림 58 프로시저 선택창	58
그림 59 클래스 파일로 불러온 프로시저 목록	59
그림 60 Biped 객체에 적용된 프로시저	59
그림 61 success_ask_for과 Jump_up 프로시저 실행동작	60
그림 62 walk_hug 프로시저 실행동작	60

I. 서론

1. 연구의 필요성 및 배경

4차 산업혁명이 화두로 떠오르면서 교육에 대한 논의가 그 어느 때보다 활발하게 이루어지고 있다. 2차 산업혁명 이전에는 도제식 교육이 있었고, 이후에는 표준화되고 획일적된 근대식 교육이 나타났다.

산업은 지속적으로 발전되었으나 근대식 교육은 큰 변화가 없이 이어져오고 있다. 이에 4차혁명 시대에 필요한 인재를 기르기 위한 교육에 대해 논의되고 있는 것이다. 21세기에 반드시 필요한 역량을 기르기 위해 코딩을 배우고 컴퓨팅 사고력¹⁾과 창의성을 길러야 한다는 주장이 이어지고 있다.

세계 여러 나라에서 정규과목으로 소프트웨어 교육을 실시하고 있고 우리나라에서도 2015년 개정 교육과정에서 소프트웨어 교육이 의무화 되었다. 또한 2022년 개정 교육과정에 소프트웨어 교육 및 인공지능 교육을 확대하기 위한 논의가 이루어지고 있다.

현대 사회에서는 지식을 습득하는 것도 중요하지만 논리적 사고력과 창의력이 바탕이 된 문제해결 능력이 필요하다. 다양한 문제를 창의적이고 효율적으로 해결하는 컴퓨팅 사고력을 함양할 수 있는 소프트웨어 교육이 필요한 이유이다. 그렇다면 4차산업 시대에 필수 교육이라는 코딩교육, 즉 프로그래밍 교육은 기존에 받아온 교육과 완전히 다른 차원의 교육인지에 대해 생각해 볼 필요가 있다.

그동안 교육의 패러다임이 시대에 따라 변화되어 왔지만 지속적으로 이어진 교육 방법이 있는데 바로 ‘이야기(Narrative)’, 즉 ‘스토리텔링’을 이용한 교육이다. 인간의 가장 근본적인 생활수단인 ‘이야기’를 교육의 수단으로 사용하는 것은 모든 영역에서 가능하다. 도제식 교육에서는 경험을 이야기 함으로써 기술을 전수하였고 근대식 교육에서는 의사소통 뿐만 아니라 읽기, 쓰기, 협동학습, 비판적 사고 등을 가르치기 위해 사용되어 왔다.²⁾

이러한 스토리텔링을 이용한 교육 중에서 소프트웨어 교육과 가장 가까운 것을 고른다면 ‘쓰기’라고 할 수 있다. ‘쓰기’는 무엇보다도 창의력이 요구되며 자기주도적인 과목이다. 소프트웨어 교육 또한 컴퓨팅 사고력을 키우기 위해 자기주도적이어야 하며 창의력이 요구되는 쓰기 교육이어야 한다.

그러나 대부분의 소프트웨어 교육이 소프트웨어 사용법이나 기능을 익히는데 치중하거나 프로그래밍 언어의 문법을 배우는 교육으로 나뉘어져 있다. 이로 인해 문제를 파악하고 문제해결 단계를 통해 해결방안을 구상하기보다는 특정 소프트웨어의 기능만 익히게 된다.

컴퓨터에서 문제를 해결하기 위해서는 알고리즘을 구상하고 작성할 수 있어야 한다. 알고리즘이란 문제를 해결하기 위한 절차를 의미하는데 다양한 방안을 구상하고 각각에 따른 효율성을 예견하여 최적의 방법을 찾아내는 효과적인 방법이다. ‘쓰기’는 복잡한 문제해결의 과정이며 쓰기를 통해 자신의 생각을 전달함으로써 복잡한 사고활동의 과정 속에서 문제를 해결해 나가는 형태로 진행되는 데 알고리즘과 닮아 있다.³⁾ 소프트웨어 교육에 스토리텔링을 적용한다면 복잡한 알고리즘을 쉽게 풀어낼 수 있을 것이다.

2. 연구의 목적

소프트웨어 교육이란 문제해결을 알고리즘으로 설계하고 구현하여 프로그래밍 언어라는 도구로 표현할 수 있는 복합적인 능력을 배양하는 교육을 말한다.⁴⁾ 프로그래밍 언어는 보편적으로 C나 Java와 같은 텍스트 기반 언어를 말하는데 학습해야 할 내용이 많고 어려워서 시작하는 단계에서는 흥미를 잃기 쉽다.

누구나 쉽게 배울 수 있는 언어를 이용한다면 좀 더 쉽게 프로그래밍 언어를 익힐 수 있을 것이다. 그동안 교육용 프로그래밍 도구와 학습방법에 대한 많은 연구가 이루어졌고, 그 결과로 다양한 블록형 프로그래밍 언어가 개발되었다. 대부분의 블록형 코딩 프로그램은 블록을 드래그 앤 드롭방식으로 조립

하여 누구나 쉽게 소프트웨어가 작동하는 원리와 동작 알고리즘을 이해할 수 있도록 하는 데 목적을 두고 있다.

소프트웨어 교육에서는 흥미를 느끼게 하고 호기심을 채워주는 것이 중요한데 스토리텔링에 최적화된 프로그래밍 도구인 앨리스(Alice)를 활용한다면 충분히 가능하다고 생각하게 되었다. 블록형 프로그래밍 언어 중에서 스토리텔링을 구현하기 적합하며 객체지향언어를 지원하고 추후 텍스트 코딩으로 연동하여 학습을 지속할 수 있으며 가상현실(VR)을 지원하는 언어는 앨리스가 유일하기 때문이다.

앨리스를 이용하여 이야기를 애니메이션으로 표현하거나 게임으로 만들기 위해 글에 나타난 의도를 파악하여 객체의 움직임과 상호작용을 표현하여야 한다. 이를 위해 등장인물들의 심리적인 상태와 사건을 연결하여 표현하기 위해 Plot Units⁵⁾를 활용할 것이다. 이야기 속 등장인물 사이의 상호작용을 표현하고 상황이나 사건을 묘사하기 위해 문장을 Primitive Plot Units 형태로 작게 분해하고 이를 앨리스로 구현하려고 한다.

스토리텔링과 프로그래밍은 의사소통의 도구이다. 스토리텔링은 사람과 사람간의 의사소통이며 프로그래밍은 사람과 컴퓨터의 의사소통인 것이다. 스토리텔링의 작은 조각인 단어와 문장을 많이 만들어낼 수 있으면 글쓰기가 조금 더 쉬워지는 것처럼 프로그래밍도 작은 단위의 명령어가 많으면 작성하기 수월하다. 이를 위해 여러 가지 작은 단위의 움직임을 Primitive Plot Units에 따라 미리 구현해 놓고 이를 재활용 할 수 있도록 할 것이다.

앨리스에서는 다양한 종류의 기본 메소드를 제공하고 있으나 객체의 움직임을 제어하는 매개변수의 수치에 대한 정확한 가이드라인이 부족하다. 그렇기 때문에 객체의 미세한 움직임을 표현하려면 많은 시간과 노력이 따른다. 이는 프로그래밍 교육에서 필요한 문제 해결력이나 컴퓨팅사고력보다는 교육용 프로그램의 기능을 익히는 데 치중하게 되는 결과를 낳을 수 있다.

이를 방지하고자 Primitive Plot Units의 예제를 통해 등장인물간의 상호작용과 마음상태를 표현할 수 있는 프로시저를 작성하려고 한다. 여러 가지 상황에 맞는 프로시저를 미리 만들어 두고 이를 파일로 저장하면 라이브러리를

구축할 수 있을 것이다. 이를 통해 엘리스 작업 시간을 크게 단축시킬 수 있을 것이고 학습자가 본질적인 프로그래밍 학습에 시간을 할애할 수 있을 것으로 여겨진다.

스토리텔링으로 세상의 모든 것을 표현할 수 있는 것처럼 프로그래밍으로도 표현이 가능하다. 이 논문에서는 스토리텔링에 최적화된 도구인 엘리스에서 프로그래밍을 쉽게 표현하기 위해 작은 단위의 조합인 라이브러리를 구축하는 방법을 제시할 것이다.

3. 논문의 구성

본 논문은 다음과 같이 5개의 장으로 구성된다.

제 1장은 본 연구에 대한 배경 및 목적을 제시하며 제 2장은 컴퓨팅 사고력과 관련된 연구 및 최근 대두되고 있는 메타버스에 대해 알아봄으로써 가상세계에서 이루어질 수 있는 소프트웨어 교육의 가능성에 대해 소개한다. 또한 소프트웨어 교육도구 중 몇 가지를 선택하여 절차지향적, 객체지향적으로 나누어 소개하고 스토리텔링의 기본요소인 Plot Units에 대해 알아본다.

제 3장에서는 스토리텔링에 특화된 소프트웨어 교육도구인 스크래치와 엘리스를 비교해보고 Primitive Plot Units 구현 및 라이브러리 구축 가능성에 대해 연구한다. 제 4장에서는 엘리스 라이브러리 구축에 대한 결론 및 향후 과제에 대해 논의한다.

II. 관련연구

1. 소프트웨어 교육

가. 컴퓨팅 사고력과 소프트웨어 교육의 목적

Seymour papert은 ‘컴퓨팅 사고력’이라는 용어를 1980년에 처음으로 소개하였다.¹⁾ 이후 1996년에 그가 기하학적 아이디어 생성을 위한 접근 방법으로 새로이 사용하면서 다시 한번 소개되었다. 컴퓨터 과학의 중심 사고인 컴퓨팅 사고력이란 ‘문제 해결방법에 대한 사고 과정이며, 문제 해결방법을 형식화하고 그 과정이 효과적으로 실행될 수 있도록 표현하는 것을 포함하는 과정’이라고 정의하였다.

또한, 컴퓨팅 사고력은 수학적 사고, 공학적 사고 및 과학적 사고와 결합하여 복잡한 문제를 분해할 수 있는 분석적 도구라고 하였다. 컴퓨팅 사고력은 컴퓨터 과학자만 알아야 하는 프로그래밍 능력이 아니라 수학·공학·과학이 융화된 고등사고력이고 개념화된 결정체이다.

ISTE & CSTA(International Society for Technology in Education & Computer Science Teachers Association, 2011)에서는 컴퓨팅 사고력을 ‘문제를 해결하는 과정’으로 정의하며 이를 문제 해결을 위한 문제 정립, 자료의 논리적 구성 및 분석, 추상화, 알고리즘적 사고, 분석 및 구현, 일반화 및 전환 등으로 들고 있다.

소프트웨어 교육의 목표는 컴퓨팅 사고력이다. Wing은 컴퓨팅 사고력은 컴퓨터 과학자 뿐만 아니라 누구나 가져야 할 역량임을 강조한다¹⁾. 프로그래밍은 인간이 상상하는 모든것을 표현하고 구체화할 수 있는 도구이다. 프로그래밍을 단순히 컴퓨터 언어 코드를 화면에 나타내는 것이 아닌, 문제를 분석하고 절차적인 해결 과정을 구상해 내고 이를 컴퓨터 언어로 표현하여 문제를 해결하는 창의적이고 효율적인 해결방식이다.

2. 소프트웨어 교육도구

가. 메타버스

‘메타버스’라는 용어가 처음 등장한 것은 1992년 닐 스티븐슨의 ‘스노우 크래쉬’라는 소설에서였다. 메타버스는 ‘Meta’라는 접두어와 ‘Universe’의 뒷부분인 ‘verse’를 결합한 용어인데 가상, 초월을 의미하는 Meta와 현실, 우주를 의미하는 Universe가 만나 현실세계와 같은 가상의 초현실 공간을 의미한다⁶⁾.

최근 코로나(COVID-19)로 인해 언택트 문화가 활성화되고 메타버스의 경제적 가치에 사람들의 관심이 쏟아지면서 기업이나 국가차원의 관심과 참여도 증가하고 있다. 특히 1997년부터 2012년 사이 출생한 Z세대⁷⁾에게 메타버스는 익숙한 공간이며 이를 활용한 소프트웨어 교육이 늘어나고 있는 중이다.

1) 로블록스 (Roblox)

2003년 설립된 온라인 게임 플랫폼 및 제작 시스템으로 누구나 쉽게 게임을 제작하고 공유 및 플레이 할 수 있는 플랫폼이다⁸⁾.

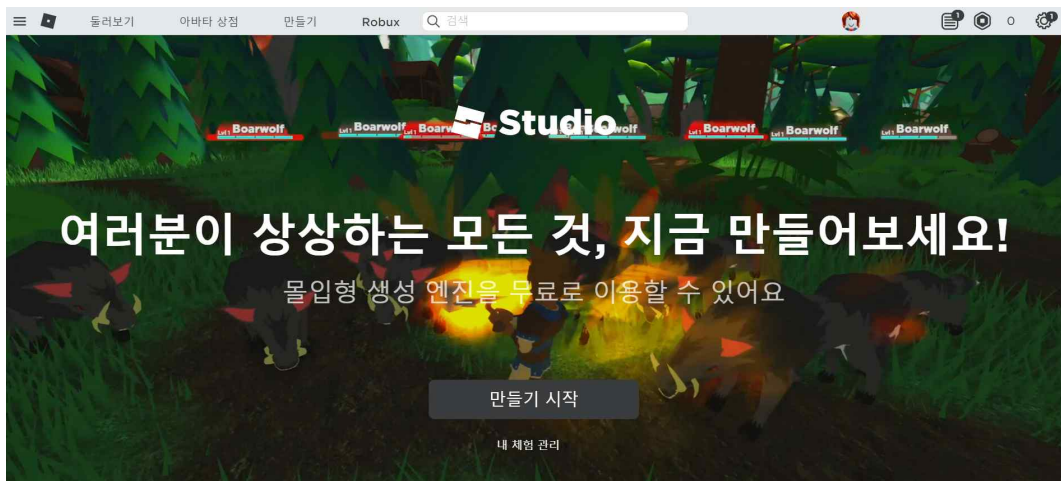


그림 1 <https://www.roblox.com/create>

사용자는 블록스에서 제공하는 ‘로블록스 스튜디오’라는 게임엔진을 사용하여 게임을 직접 개발할 수 있고 제작된 게임은 다른 사용자가 플레이할 수 있도록 발행(publish)하여 공유할 수 있다.

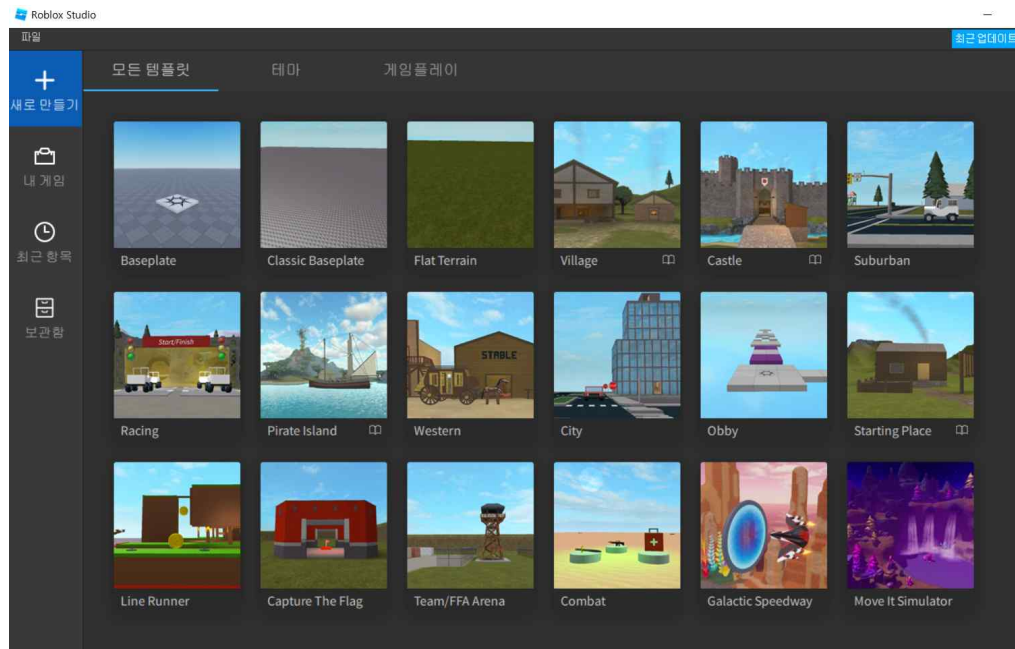


그림 2 로블록스 스튜디오

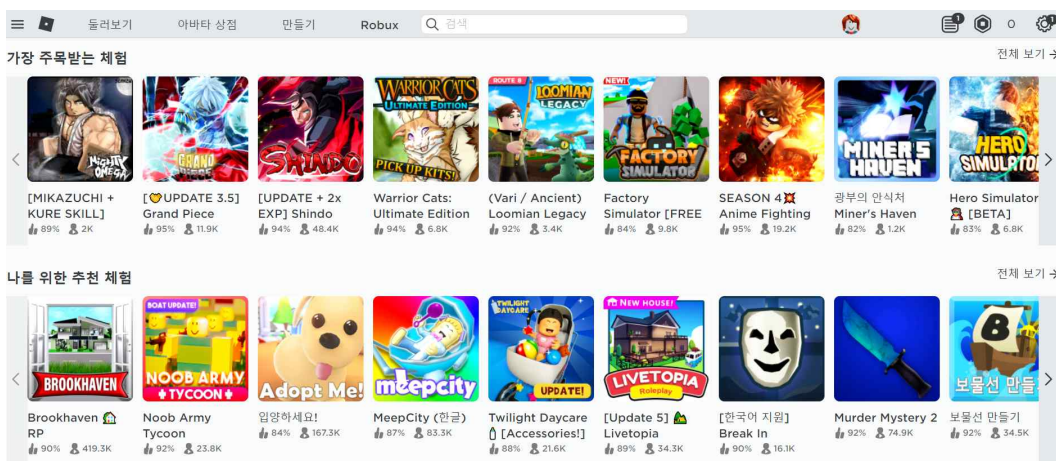


그림 3 <https://www.roblox.com/games#/>

게임 개발을 위해서는 객체지향 프로그래밍인 루아 언어를 사용하는데 콘텐츠를 만들어 수익을 올릴 수도 있다. 또한 각종 템플릿과 튜토리얼을 제공하고 있어 초등학생도 쉽게 게임을 만들 수 있다. 특히 로블록스 에듀케이션 사이트(<https://education.roblox.com>)를 통해 커리큘럼 및 수업 콘텐츠를 무료로 제공하고 있다. 이 콘텐츠를 교육도로 활용하면 모든 연령의 학생에게 소프트웨어 교육 및 게임 디자인, 기업가 정신 등에 대해 쉽게 가르칠 수 있다.

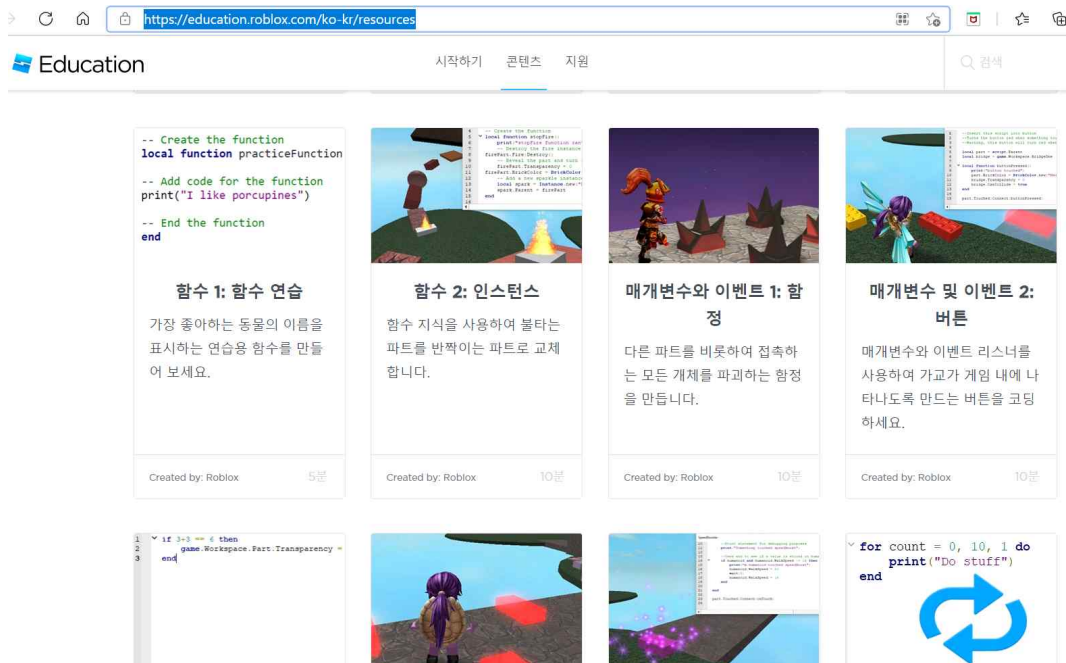


그림 4 <https://education.roblox.com>

2) 유니티 (Unity)

2004년 설립된 유니티는 2D 및 3D 비디오 게임을 제작하는 도구인 게임 엔진이다. 현재는 VR과 애니메이션 등 콘텐츠 제작을 위한 도구로도 널리 사용되어 영역을 넓혀가고 있다. 물리 엔진이 탑재되어 있어 손쉽게 다양한 효과를 주거나 시뮬레이션이 가능하며 완성도 있는 작품을 만들 수 있는 장점이

있다. 하지만 초등학생이 배우기에는 다소 난이도가 높고 C#을 활용하여 게임을 제작하는데 어려움을 느낄 수 있다. 다만 최근 추가된 LEGO 마이크로 게임 제작툴을 활용하면 가상 LEGO 브릭으로 쉽게 게임을 제작하고 공유할 수 있도록 하였다⁹⁾. 유니티에서 제공하는 튜토리얼을 단계별로 따라하면 30분 내에 나만의 게임을 완성할 수 있다.



그림 5 <https://store.unity.com/kr/lego-microgame>

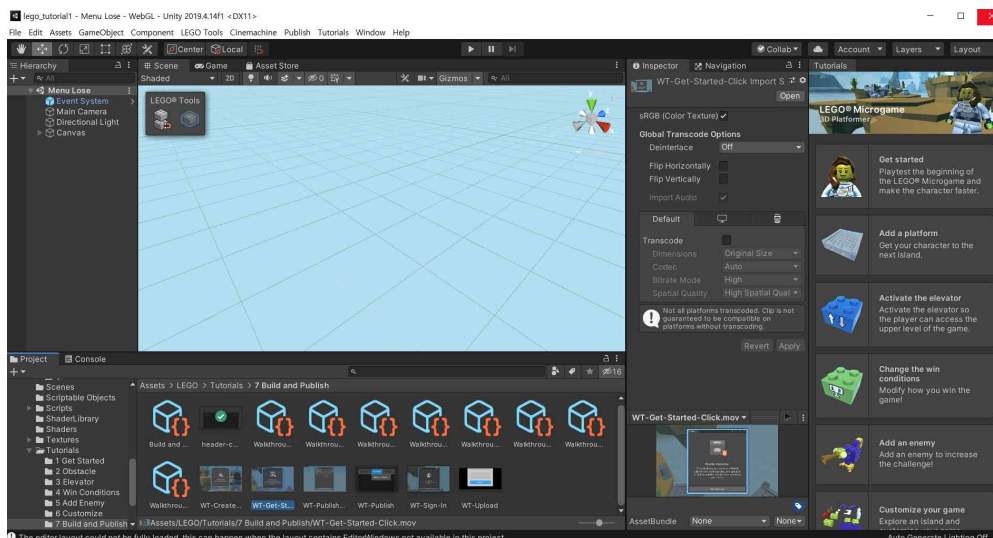


그림 6 유니티 스토어 - 레고 마이크로게임

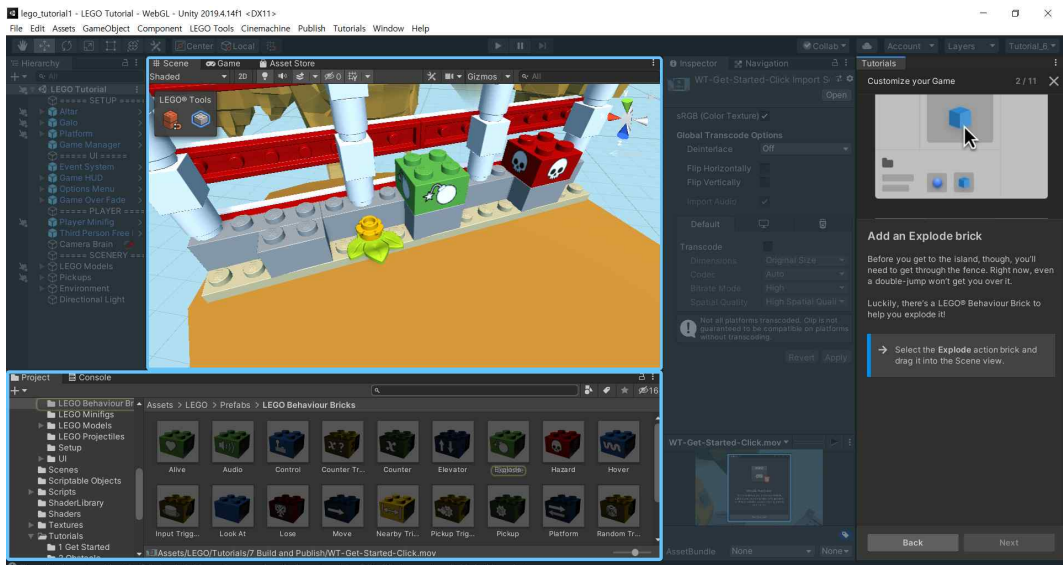


그림 7 튜토리얼 : 유니티 스토어 - 레고 마이크로게임

3) 마인크래프트 (Minecraft)

Mojang Studios에서 제작한 3D 건설 게임으로 3차원 세상에서 블록을 배치하고 부수며 여러 구조물과 작품을 만들며 농사를 짓거나 사냥을 하면서 생존해가는 게임이다¹⁰⁾. 2014년 마이크로소프트가 Mojang을 25억달러에 인수하였고 Mojang은 마이크로소프트의 자회사가 되었다.

코로나로 대면 행사가 어려웠던 작년 어린이날에 청와대가 마인크래프트를 활용해 청와대 가상 공간을 체험할 수 있도록 청와대 맵을 제작하였다. 마인크래프트 자바에디션을 구입하면 마인크래프트를 이용한 코딩이 가능하여 아이들의 흥미를 이끌어줄 수 있다. 더 나아가 API 설치 후 마인크래프트 서버를 실행하면 파이썬에서 API를 통해 컨트롤이 가능하기 때문에 게임을 통한 파이썬 교육이 가능하다. 조건문과 반복문을 적절히 이용하여 건축시간을 단축시키거나 캐릭터의 위치를 순식간에 이동시키는 등 재미의 요소를 가미한다면 파이썬을 통해 객체지향 프로그래밍 개념을 자연스럽게 익힐 수 있다.

```

C:\WINDOWS\system32\cmd.exe - start.bat
Java HotSpot(TM) 64-Bit Server VM warning: Ignoring option MaxPermSize; support was removed in 8.0
*** Error, this build is outdated ***
*** Please download a new build as per instructions from https://www.spigotmc.org/ ***
*** Server will start in 15 seconds ***
Loading libraries, please wait...
[12:26:50 INFO]: Starting minecraft server version 1.11.2
[12:26:50 INFO]: Loading properties
[12:26:50 INFO]: Default game type: CREATIVE
[12:26:50 INFO]: This server is running CraftBukkit version git-Spigot-6de2fbc-claa659 (MC: 1.11.2) (Implementing API version 1.11.2-R0.1-SNAPSHOT)
[12:26:50 INFO]: Debug logging is disabled
[12:26:50 INFO]: Server Ping Player Sample Count: 12

[12:26:50 INFO]: Using 4 threads for Netty based IO
[12:26:51 INFO]: Generating keypair
[12:26:51 INFO]: Starting Minecraft server on *:25565
[12:26:51 INFO]: Using default channel type
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by io.netty.util.internal.PlatformDependent0 (file:/C:/dlab_dc3_win/%ec%94%9c%eb%b2%84%ed%8c%ec%9d%bc/server/spigot.jar) to field java.nio.Buffer.address
WARNING: Please consider reporting this to the maintainers of io.netty.util.internal.PlatformDependent0
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[12:26:51 INFO]: Your platform does not provide complete low-level API for accessing direct buffers reliably. Unless explicitly requested, heap buffer will always be preferred to avoid potential system instability.
[12:26:51 INFO]: Set PluginClassLoader as parallel capable
[12:26:51 INFO]: [SetSpawn] Loading SetSpawn v2.1
[12:26:51 INFO]: [RaspberryJuice] Loading RaspberryJuice v1.7
[12:26:51 WARN]: **** SERVER IS RUNNING IN OFFLINE/INSECURE MODE!
[12:26:51 WARN]: The server will make no attempt to authenticate usernames. Beware.
[12:26:51 WARN]: While this makes the game possible to play without Internet access, it also opens up the ability for hackers to connect with any username they choose.
[12:26:51 WARN]: To change this, set "online-mode" to "true" in the server.properties file.
[12:26:51 INFO]: **** Beginning UUID conversion, this may take A LONG time ****
[12:26:51 INFO]: Preparing level "world"
[12:26:51 INFO]: ----- World Settings For [world] -----
[12:26:51 INFO]: Item Merge Radius: 2.5
[12:26:51 INFO]: Item Despawn Rate: 6000
[12:26:51 INFO]: Arrow Despawn Rate: 1200
[12:26:51 INFO]: View Distance: 10
[12:26:51 INFO]: Allow Zombie Pigmen to spawn from portal blocks: true
[12:26:51 INFO]: Nerfing mobs spawned from spawners: false

```

그림 8 파이썬 코딩을 위한 마인크래프트 서버 구동



그림 9 컴퓨터에 설치된 마인크래프트 서버

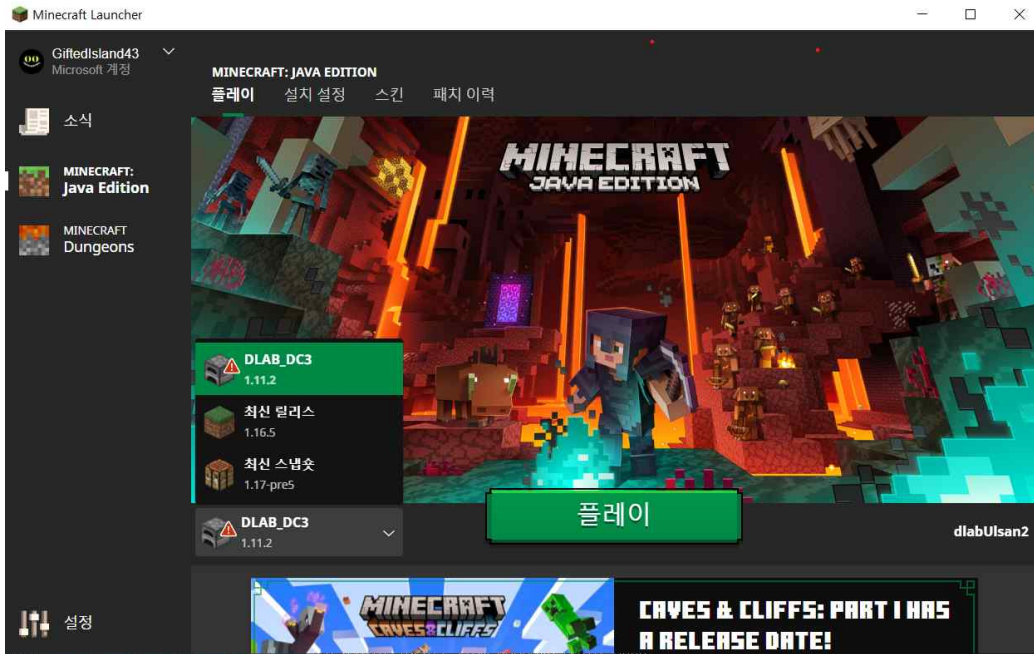


그림 10 마인크래프트 자바 버전 메인 페이지

나. 교육용 프로그래밍 언어

소프트웨어 개발이 목적인 일반 프로그래밍 언어와 달리 교육용 프로그래밍 언어는 프로그래밍 학습용으로 개발된 언어이다. 문법이 복잡하고 문법오류(Syntax error)에 민감한 일반 프로그래밍 언어와 달리 초보자가 직관적으로 사용할 수 있도록 시각적인 요소를 가지고 있는 경우가 많다. 초보자에게 어려울 수 있는 오류문제를 최소화 하기 위해 대부분 블록형으로 조합하는 방법을 사용한다. 텍스트를 사용하지 않고 그래픽적으로 프로그래밍이 가능하다고 하여 비주얼 프로그래밍 언어(Visual Programming Language, VPL)라고 분류되기도 한다¹¹⁾.

1) 절차지향형 프로그래밍 교육도구

가) 스크래치 (Scratch)¹²⁾

스크래치(Scratch)는 미국 MIT Media Lab에서 만든 블록형 프로그래밍 언어이다. 사용자가 감독이 되어 한편의 작품을 만드는 설정을 바탕으로 배우에 해당하는 스프라이트와 무대가 되는 배경에 블록을 조합하여 프로그래밍 할 수 있도록 한다. 팔레트 영역에 모여있는 블록들을 조립하여 스프라이트의 움직임을 시각적으로 확인하면서 제어할 수 있어 초보자에게 보다 쉽게 프로그래밍 기초개념을 학습시킬 수 있는 도구이다.

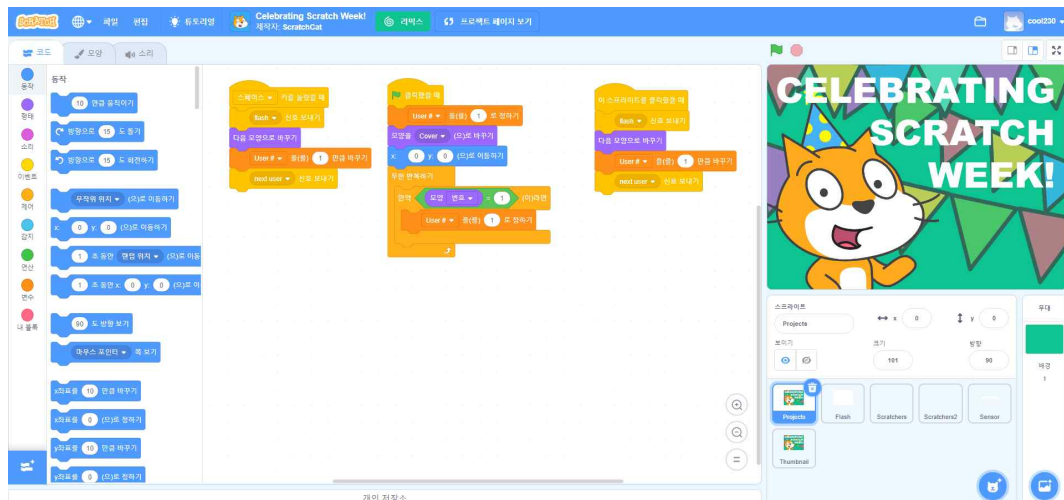


그림 11 스크래치

나) 엔트리 (Entry)¹³⁾

MIT 블록 코딩을 참고하여 소프트웨어를 배울 수 있는 공간을 조성하고 우리나라 교육 현장에 더 적합한 환경을 만들기 위해 엔트리가 출시되었다. 비영리 교육 플랫폼으로 현재는 인공지능과 데이터분석 기능을 추가해 AI 머신러닝 교육에 활용할 수 있도록 하였다. 드래그 앤 드롭을 이용한 블록형 언어지만 엔트리파이썬을 제공하여 텍스트 언어로 어떻게 표현될 수 있는지 알아볼 수 있도록 하였다.

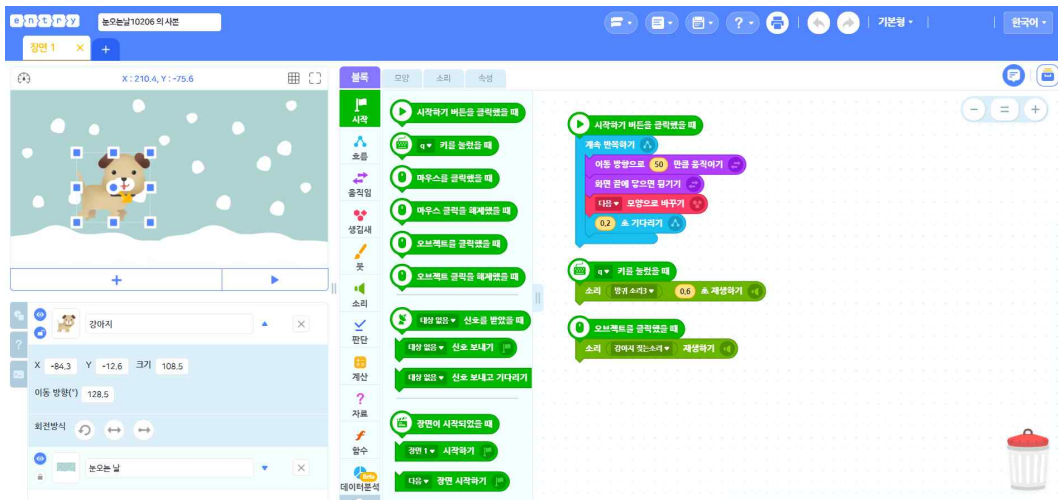


그림 12 엔트리

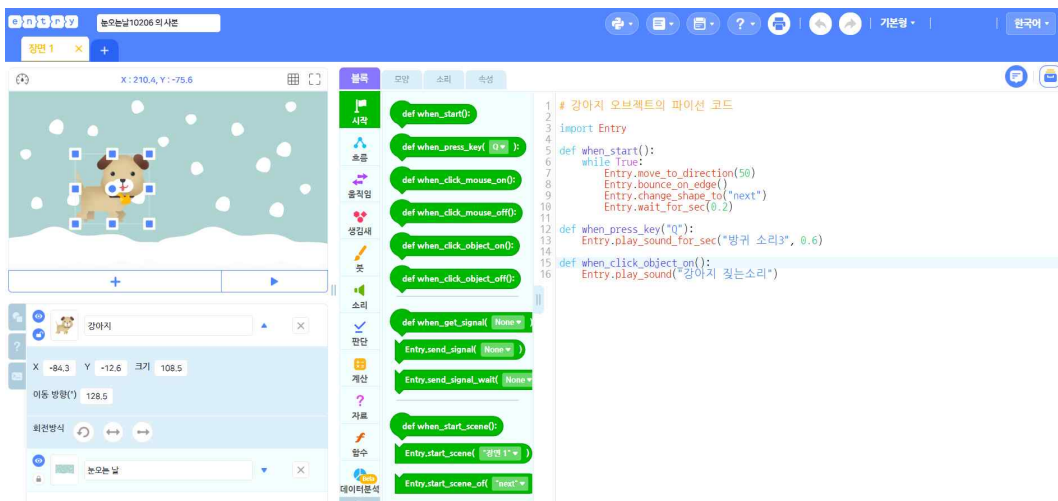


그림 13 엔트리 파이썬

2) 객체지향형 프로그래밍 교육도구

가) 앨리스 (Alice)¹⁴⁾

앨리스는 Carnegie Mellon 대학이 무료로 제공하는 교육용 소프트웨어로 3D 애니메이션과 게임을 쉽게 만들 수 있는 도구이다. 앨리스는 1995년 VR 프로토타이핑 도구로 개발되어 현재는 스토리텔링을 이용한 컴퓨터 교육 및

애니메이션, 게임 제작도구로 널리 사용되고 있다.

Java 프로그램을 효과적으로 학습할 수 있도록 고안된 블록형 언어인 앨리스는 객체지향 언어를 애니메이션을 통해 쉽게 익힐 수 있도록 커리큘럼을 제공하고 있으며 넷빈즈(NetBeans) 개발 환경을 플러그인으로 지원함으로써 블록코딩을 자바 텍스트 언어로 변환할 수 있는 환경을 제공해준다. 이를 통해 블록코딩에서 텍스트코딩으로 쉽게 나아갈 수 있는 교두보 역할을 하고 있다. 또한 Alice3 Plyer라는 어플리케이션을 통해 앨리스를 통해 구현한 애니메이션을 앨리스 IDE 설치 없이 볼 수 있도록 하고 있으며 VR mode 설정을 통해 VR과 연동이 가능하다.

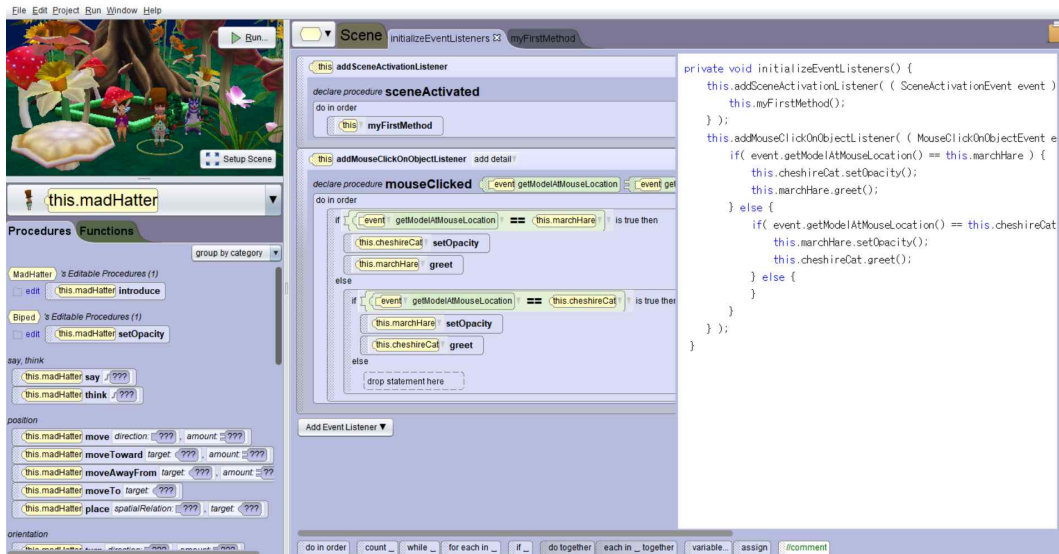


그림 14 앨리스 IDE

나) 코두 (Kodu)¹⁵⁾

코두는 마이크로소프트 코두랩에서 개발하였으며 게임제작용 3D 비주얼 프로그래밍 언어이다. 초보자도 쉽게 활용할 수 있도록 타일 형식의 아이콘 명령어(Tile-based)를 통해 스토리텔링 기반의 게임 프로그램을 학습하는데 적합하다. 다른 언어들과 달리 게임제작 전용 언어로 컴퓨터 언어가 제공하는

명령을 모두 제공하지는 않지만 시각적으로 쉽게 게임을 제작할 수 있어 초중 등 학생들에게 적합한 언어이다. 현재는 오픈소스로 전환되어 깃허브(GitHub)를 통해 사이트를 호스팅하고 있다.



그림 15 코두(Kodu)



그림 16 코두 - 아이콘(타일)형식 블록을 사용하여 코딩

다) 앱 인벤터 (App Inventor)¹⁶⁾

앱 인벤터는 안드로이드 기반 휴대폰 앱을 개발할 수 있도록 만들어진 스마트폰용 앱 저작도구이다. 자바를 기반으로 제작되었으며 명령어를 입력하는 형식이 아닌 블록을 맞추어 조립하는 형태로 되어 있어 손쉽게 앱 제작이 가능하다. 앱 디자인 페이지를 코딩부분과 분리하여 화면을 디자인 하는데 편리하게 구성되어 있다. 초중등 학생들에게 객체지향 언어의 개념을 학습시키기 적합한 도구이다.¹⁷⁾

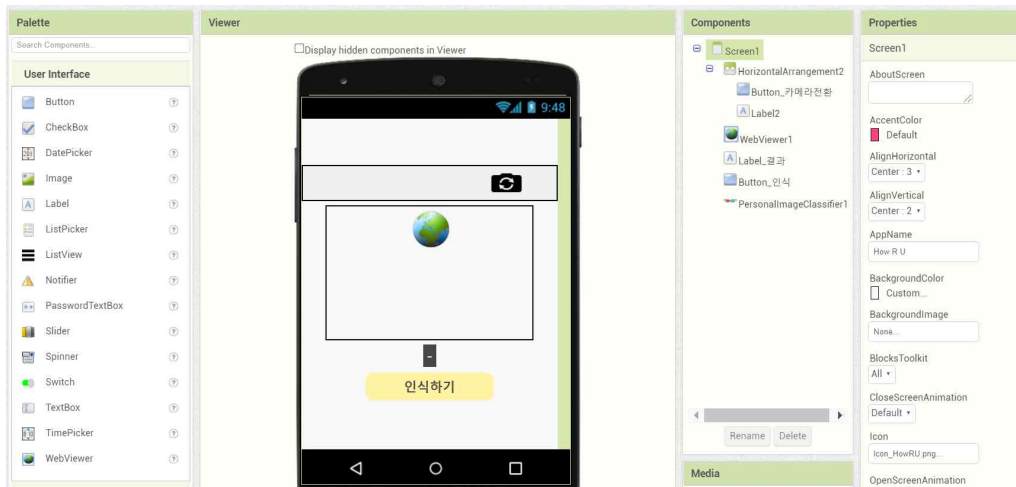


그림 17 앱인벤터 - 화면 디자인(Designer) 페이지

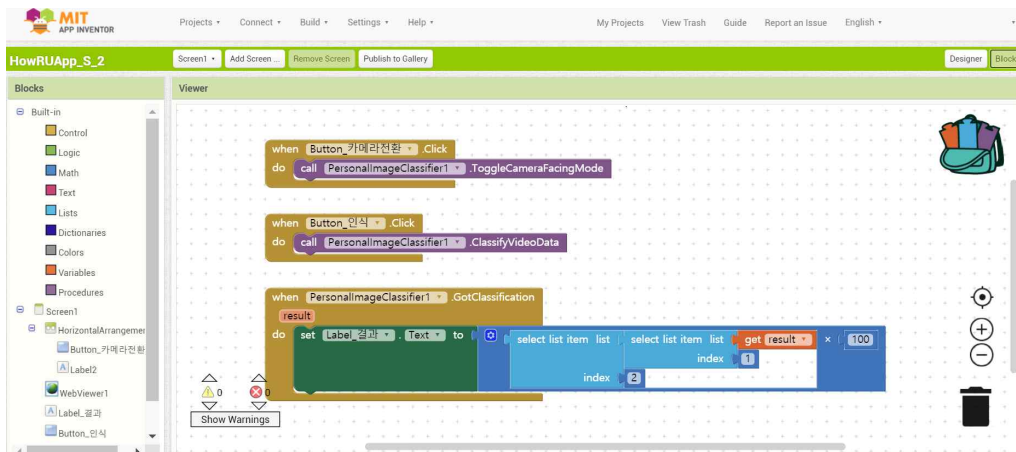


그림 18 앱인벤터 - 블록코딩(Blocks) 페이지

3. 이야기 구성

자연언어처리 분야에서 문서요약이나 컴퓨터 텍스트 이해에 관한 연구 및 질의응답, 대화형 Agent 등과 같이 컴퓨터를 이용하여 인간의 언어현상을 묘사할 수 있는 연구는 오래도록 활발하게 진행되어 왔다. 더 나아가 인공지능을 이용한 창작활동에 대한 연구도 이루어지고 있다.

사람의 개입이 필요하다고는 하지만 BRUTUS라는 이야기를 들려주는 인공지능 프로그램에 대한 연구도 있었고¹⁸⁾ 글을 써주는 인공지능도 있다.¹⁹⁾ 언어에 따른 차이는 있겠으나 인공지능의 자연어 처리 능력은 최근 급속하게 향상되었으며 챗봇이나 인간과의 협업을 통한 창작활동 등을 통해 일상생활에서 사용되고 있다.

Event Relations in Plan-Based Plot Composition(Angelo E. M. Ciarlini, ACM Computers in Entertainment, Vol. 7, No. 4, Article 55, Publication date: December 2009)에서는 게임 제작을 위해 상호작용이 가능한 스토리텔링 및 플롯을 계획에 따라 체계적으로 구현하는 과정을 제시하고 있다. 사용자의 선택에 따라 스토리가 달라지는 시스템을 구성하는 것이다.

이야기를 구성하는 데 있어 가장 기본이 되는 것이 Plot Units이라고 할 수 있다.⁵⁾ 이야기의 흐름을 분석하거나 요약하기, 또는 플롯 구성을 위한 기본 단위로 Plot Units를 사용하여 이야기에 등장하는 주인공들간의 상호작용을 표현하는 것이다.

Plot Units를 등장인물간의 내부상황과 외부상황을 표현하는 기본 단위로 사용하여 이야기를 체계적으로 구성할 수 있으며 Plot Units의 종류와 개수가 많으면 많을수록 이야기는 풍성해질 것이다.

III. 스토리텔링 적용

1. 스토리텔링의 도구

소프트웨어 교육은 일상생활에서의 문제를 컴퓨팅 관점에서 규정하고 문제 해결 방법을 탐색해 효율적인 해결절차를 마련하는 것인데 이를 컴퓨터 사고력이라 한다.¹⁾ Jeannette Wing(2006)은 컴퓨팅 사고는 읽기, 쓰기, 연산과 같이 현대사회를 살아가는 데 기본적이고 필수적인 기술이라고 하였다. 컴퓨팅 사고란 문제해결에 컴퓨팅 기술을 활용하는 것으로 분해(Decomposition), 패턴인식(Pattern Recognition), 추상화(Abstract), 알고리즘(Algorithm)의 단계로 이루어져 있다.

알고리즘적 사고를 통한 문제 해결과정에 공통적으로 갖추어야 할 역량이 창의성이다. 스토리텔링은 무엇보다 창의적인 활동이며 이를 통해 컴퓨터사고력을 키울 수 있다. 그러나 보편적으로 프로그래밍 언어를 학습하는 방식은 텍스트 코딩을 통해 언어 문법을 외우고 실행하여 오류를 수정해가는 방법이다. 프로그래밍에 입문하는 학생들에게 문법오류는 심리적 부담으로 다가오고 소프트웨어 교육의 본래 목적을 잃어버리게 되는 경우가 있다.

앨리스는 이러한 요인을 해결하기 위해 개발되었고 다양한 그래픽 객체를 통해 3차원 가상공간을 활용하는 멀티미디어 환경을 제공한다. 앨리스는 드래그 앤 드롭 인터페이스를 지원하는 블록코딩 형식으로 구문오류를 방지할 뿐 아니라 객체지향 언어를 시각적으로 학습하는데 특히 유용하다.

학습자는 스스로 애니메이션을 제작하면서 자연스럽게 컴퓨터 사고력 및 창의력을 기를 수 있게 된다. 또한 프로그래밍 입문 교육에 스토리텔링을 활용하여 앨리스로 교육한 사례²⁰⁾를 통해 학생의 흥미를 유발시키고 텍스트 코딩에 대한 부담을 덜어주는데 효과적이라는 것을 알 수 있다.

스토리텔링 구현을 위한 대표적인 교육용 프로그래밍 도구로 알려진 스크래치를 앨리스와 비교해 보았다.

	앨리스	스크래치
다양한 배경	O	O
등장인물	O	O
3D	O	X
객체지향	O	X
코드 표현	블록(타일)형 - 한 개 화면에 표현	블록형 - 각 스프라이트마다 코드 화면이 따로 있음
VR 연계	O	X
텍스트 변환	O	X
문법오류(Syntax Error)	X	X
캐릭터의 Joints(관절) 유 무	O	X
캐릭터 움직임	용이함	캐릭터를 여러개 추가하면 가능
배경과 캐릭터의 다양성	△	O

표 1 스토리텔링 도구 : 앨리스와 스크래치 비교

[표 1]에서 알 수 있듯이 앨리스는 3D 환경을 제공하며 객체지향방식을 사용한다. 또한 코드를 표현함에 있어 스크래치는 각각의 스프라이트(등장인물) 별로 별도의 페이지에 코딩하는 방식을 취하므로 스프라이트가 많아질수록 복잡하고 코드의 가독성이 떨어지게 된다. 반면 앨리스는 MyFirstMethod 탭에 모든 코드를 작성하고 객체에 따른 프로시저를 한눈에 볼 수 있기 때문에 가독성이 높다.

특히 앨리스는 유니티 플러그인을 지원하여 VR 환경에서 작업이 가능하며 NetBeans를 통한 자바 코드로의 변환도 용이하다. 또한 캐릭터에 Joints를 적용함으로써 자유롭고 자연스러운 행동을 표현할 수 있다는 장점이 있다.

다만 스크래치에서는 새로운 캐릭터를 그리거나 이미지 파일을 스프라이트로 바꾸는 것이 용이하지만 3D 환경인 앨리스에서는 제한적으로만 가능하다는 아쉬움이 있다.

그럼에도 스토리텔링 교육을 위한 도구로 앨리스가 스크래치에 비해 우위에 있으며 스토리텔링 애니메이션을 구현하는데 효과적이고 최적화된 프로그램이라 할 수 있겠다.

또한 다른 블록형 프로그램과 달리 객체지향의 특성을 완벽하게 지원하여 학습자가 자연스럽게 객체지향 개념을 파악할 수 있게 한다. 앨리스를 통해 3D 가상환경을 경험해 봄으로써 3D 모델링 환경에 익숙해지는 것도 의미를 지닌다고 생각된다.

가. Alice로 구현하기

앨리스는 추상적인 개념을 현실적으로 표현할 수 있는 형태를 가지고 있다. 앨리스는 크게 '월드'라는 배경과 월드가 포함하고 있는 객체로 구성되어 있으며 월드 자체도 객체이다. 스토리를 애니메이션으로 구현하기 위해 Alice를 사용하기로 하였다. 사용하기 위한 기본 작업은 월드와 주인공을 설정하는 것인데

앨리스를 실행하면 월드의 배경을 선택할 수 있는 템플릿 화면이 [그림 19]와 같이 나타난다.

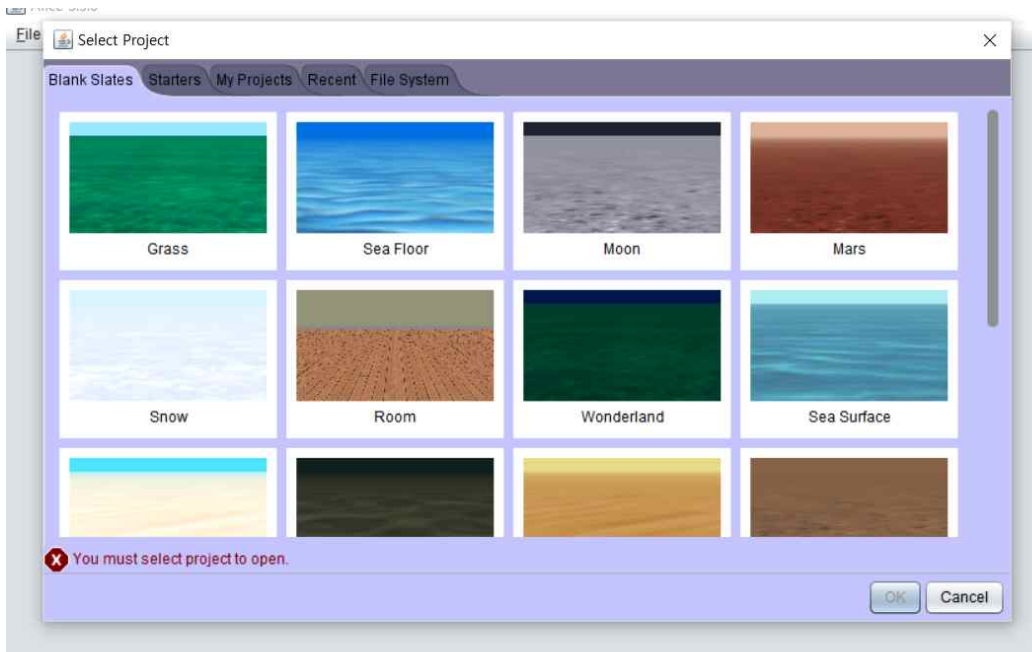


그림 19 엘리스 시작 화면 - 배경 선택하기

원하는 배경을 선택 후 <OK> 버튼을 누르면 엘리스 초기 화면이 [그림 23]과 같이 나타난다.

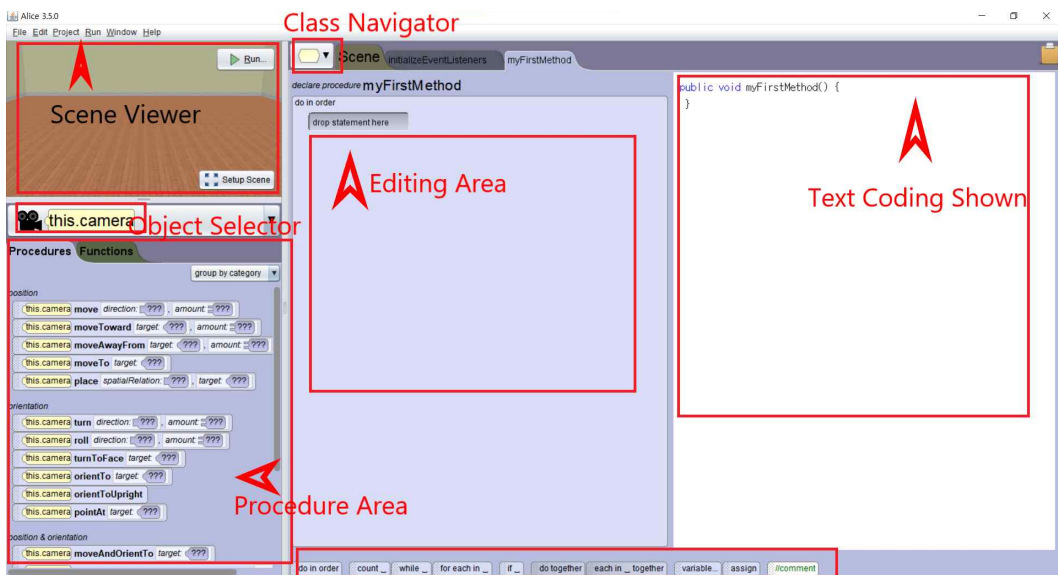


그림 20 엘리스 초기 화면

Screen Viewer에는 작성하고자 하는 3D 월드가 나타나게 되며 Object

Selector를 통해 객체를 선택할 수 있다. 앨리스에는 기본적으로 제공하는 프로시저가 있는데 Procedure Area 부분에서 찾아볼 수 있다. 필요한 프로시저를 드래그 앤 드롭으로 Editing Area로 끌어와 조립하면 코드를 완성할 수 있다. Scene Viewer의 <Run> 버튼은 작성된 코드가 의도한 대로 작동되는지 확인할 수 있는 실행 버튼이다.

<Scene Setup> 버튼은 스토리텔링에 사용할 객체를 추가할 수 있는 기능이다. <Scene Setup> 버튼을 누르면 [그림 21]과 같이 씬을 설정할 수 있는 화면이 나타난다.

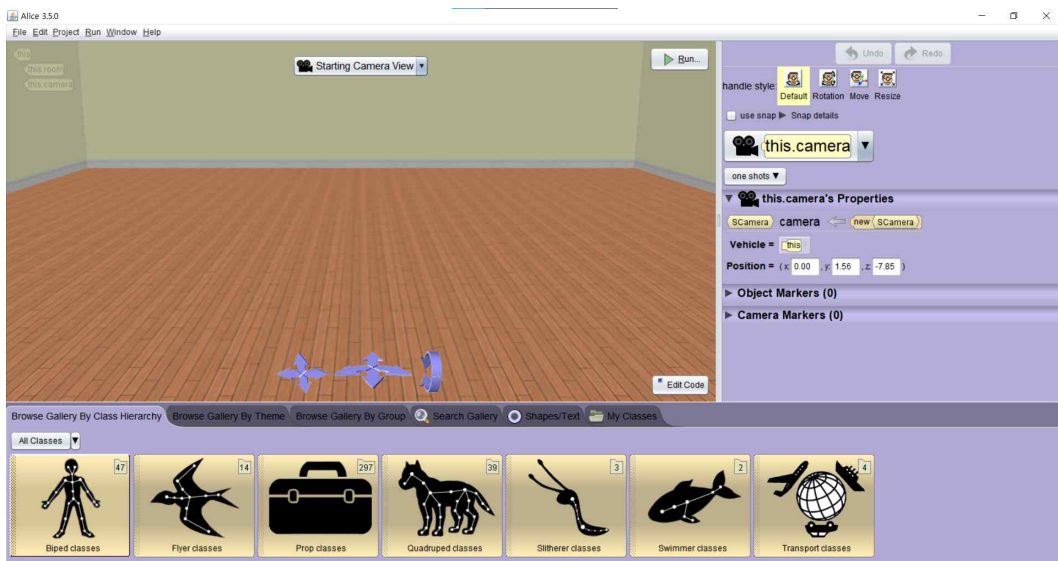


그림 21 Scene Setup 초기 화면

월드를 구성하고 있는 객체는 클래스로부터 만들어지는데 앨리스에서는 ‘갤러리’라는 컬렉션에서 객체를 선택할 수 있다. 컬렉션은 다양한 형태를 일곱 가지의 클래스로 분류해 놓은 집합이며 각각의 클래스의 구성은 [그림 22]와 [표 2]에 정리하였다.

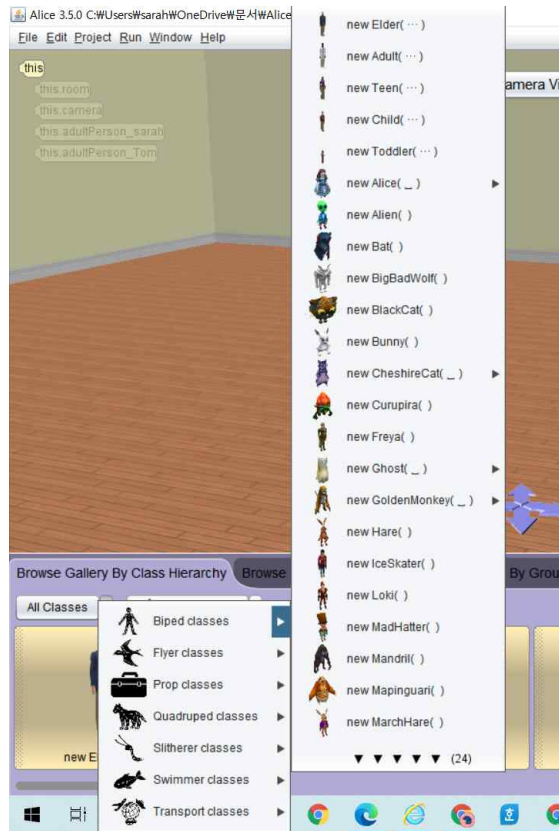


그림 22 엘리스 갤러리

클래스	분류 기준	대표 객체들
Biped	두 발로 서있는 생물	Adult Person, Teen Person
Flyer	날개가 있는 생물	Blue Bird, Chicken, Penguin
Prop	각종 소품	Trees, Castle, Furniture
Qadruped	네 발로 서있는 생물	Cat, Bear, Dog, Hippo
Slithered	기어다니는 생물	HermitSnail, LongNeckSnail
Swimmer	헤엄치는 생물	Fish Classes, MarineMammal
Transport	교통수단	AirCraft, Automobile, Train, WaterCraft

표 2 엘리스 갤러리에 정의된 클래스

객체를 선택하기 위해 갤러리의 ‘Biped Classes’-‘New Adult(...)’를 선택하면 다양한 속성을 지정할 수 있다.

‘New Adult’는 성인 캐릭터인데 얼굴 생김새, 머리모양, 피부색, 의류를 지정할 수 있고 원한다면 다른 성별 및 나이대의 캐릭터로 쉽게 바꿀 수 있다. 또는 <generate random person> 버튼을 눌러 무작위로 객체를 지정할 수도 있다.[그림 24]~[그림 27]

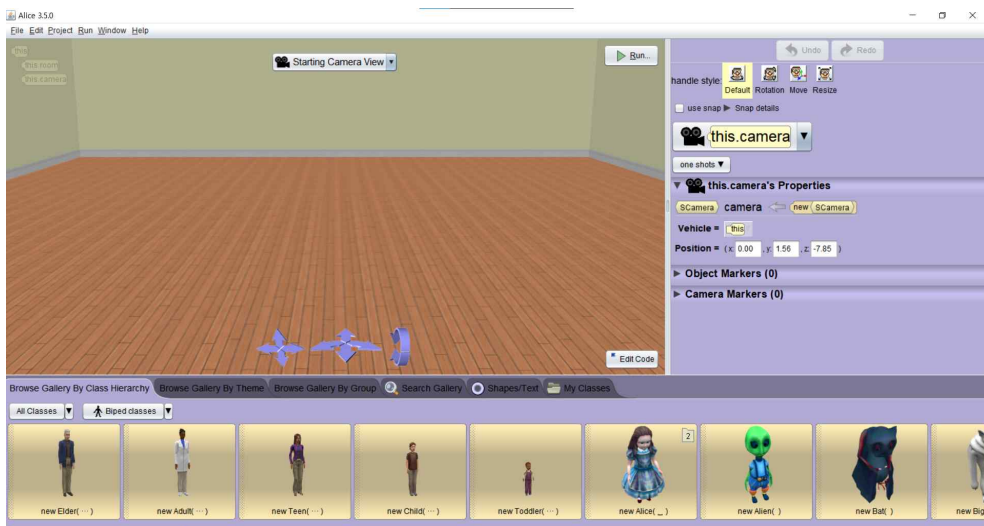


그림 23 Setup Scene 화면 - Biped Classes

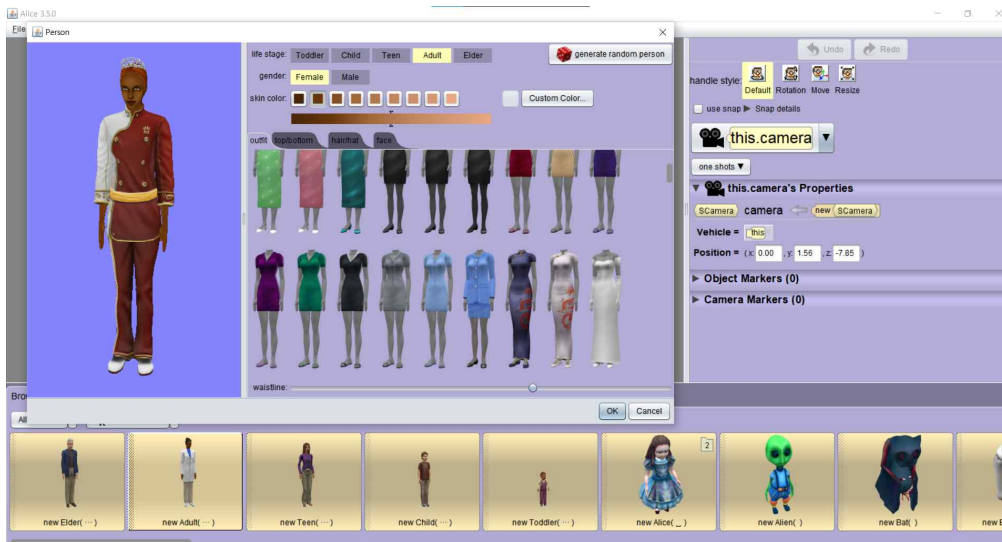


그림 24 Biped Adult Person 속성 - outfit 탭 : 의복 속성



그림 25 Biped Adult Person 속성 - top/bottom 탭 : 의복 속성

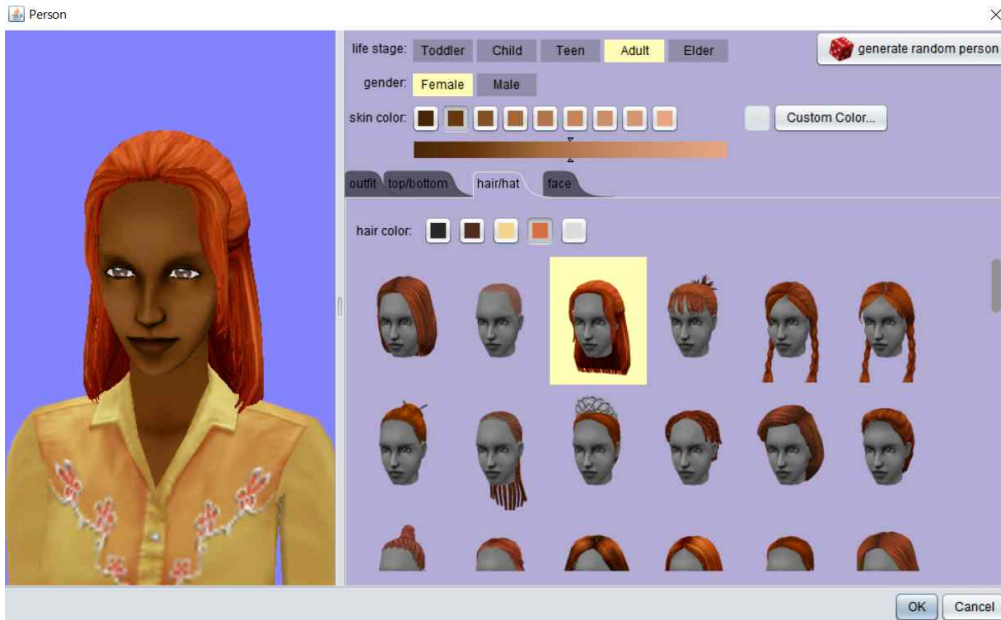


그림 26 Biped Adult Person 속성 - hair/hat 탭 : 머리스타일 속성

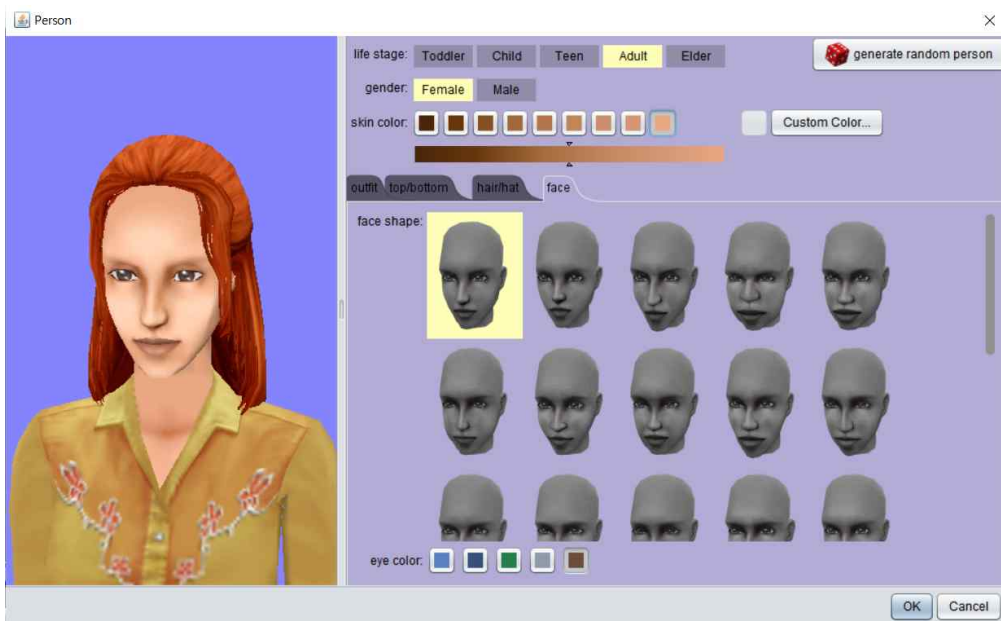


그림 27 Biped Adult Person 속성 - face 탭 : 얼굴형태 속성

속성을 모두 지정한 후 <OK>버튼을 누르면 [그림 28]과 같은 객체 저장

화면이 나타나고 'initializer' 부분에 지정된 속성이 추가된 것을 볼 수 있다.

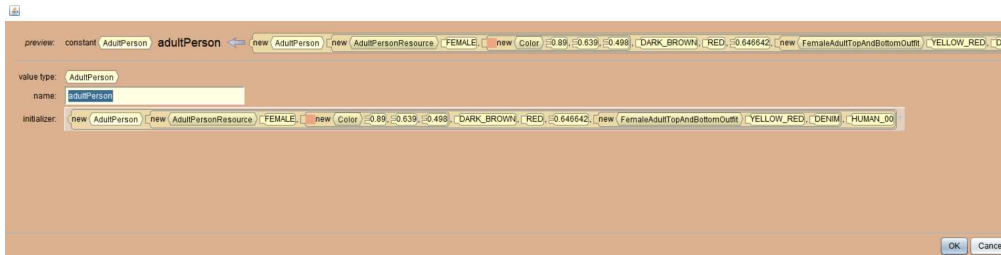


그림 28 Biped Adult Person 이름 및 속성 저장

객체를 생성하면 [그림 29]와 같이 객체가 생성된 것을 확인할 수 있다.

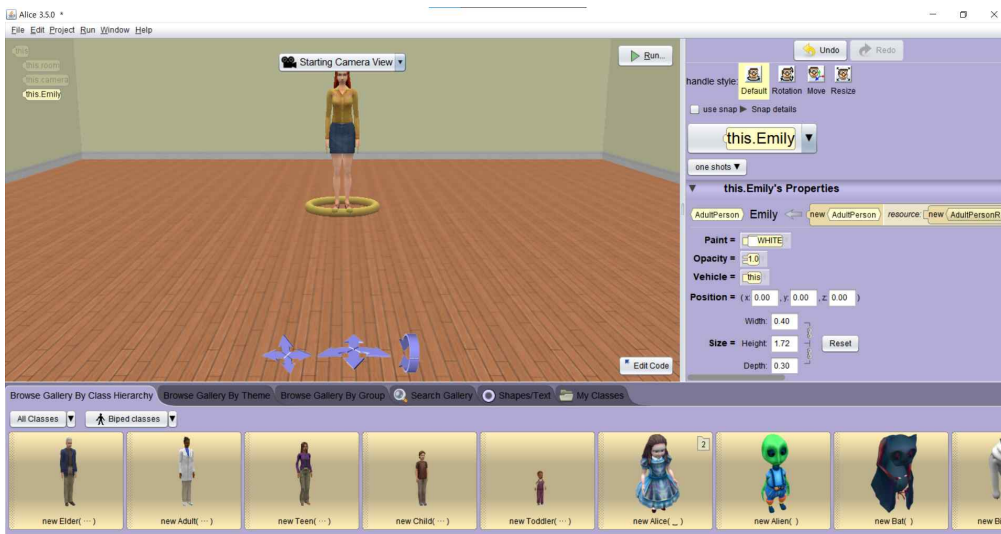


그림 29 Scene Setup - New Adult 객체 생성

앨리스에서 객체를 이동시키거나 투명도, 크기 등의 속성을 변경하고 싶을 때 <SetupScene> 화면의 우측 패널에서 변경이 가능하다.

앨리스의 Biped 객체는 40여 개의 Joints로 구성되어 있고 각각의 Joint는 프로시저를 사용한 제어가 가능하다. [그림 30]

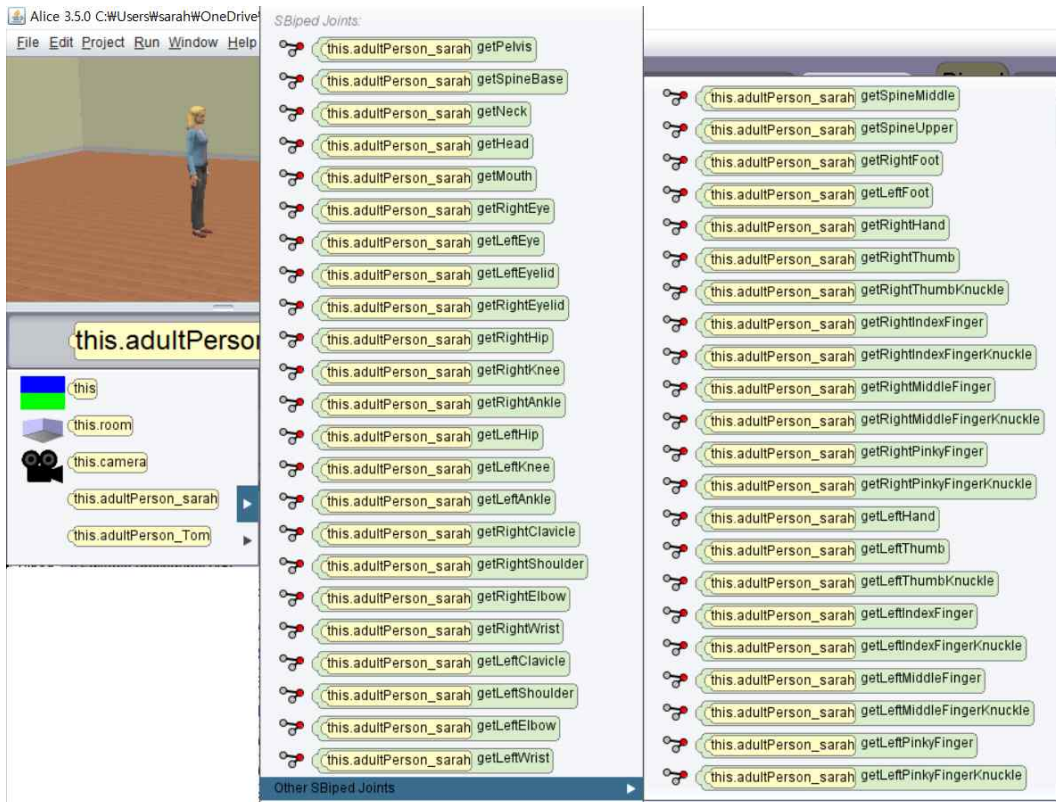


그림 30 Biped - Adult 객체의 구성 (Joints)

앨리스에서 객체의 움직임을 확인해 보기 위해 왼손을 올리고 손가락을 구부리는 간단한 움직임을 구현해 보려 한다. 팔을 앞으로 올리기 위해서는 두 가지 방법을 사용할 수 있다. 첫 번째는 getLeftShoulder(왼쪽어깨)의 turn 프로시저를 사용하는 방법이고 나머지는 getLeftClavicle(왼쪽 쇄골)의 roll 프로시저를 사용하는 것이다.

앨리스에서 객체의 getLeftShoulder를 선택한 후 <Oneshot> 버튼을 누르고 Procedure를 선택하면 getLeftShoulder에서 사용가능한 프로시저가 [그림 31]과 같이 나타난다. 이 중 Direction을 Right으로 정하고 Amount를 0.25로 선택하면 [그림 32]와 같이 왼쪽 팔을 올린 모습을 볼 수 있다. Amount는 1.0이 대체적으로 360도 회전한다고 볼 수 있고 0.25의 경우에는 팔을 4분의 1정도 올린 모습이 된다.

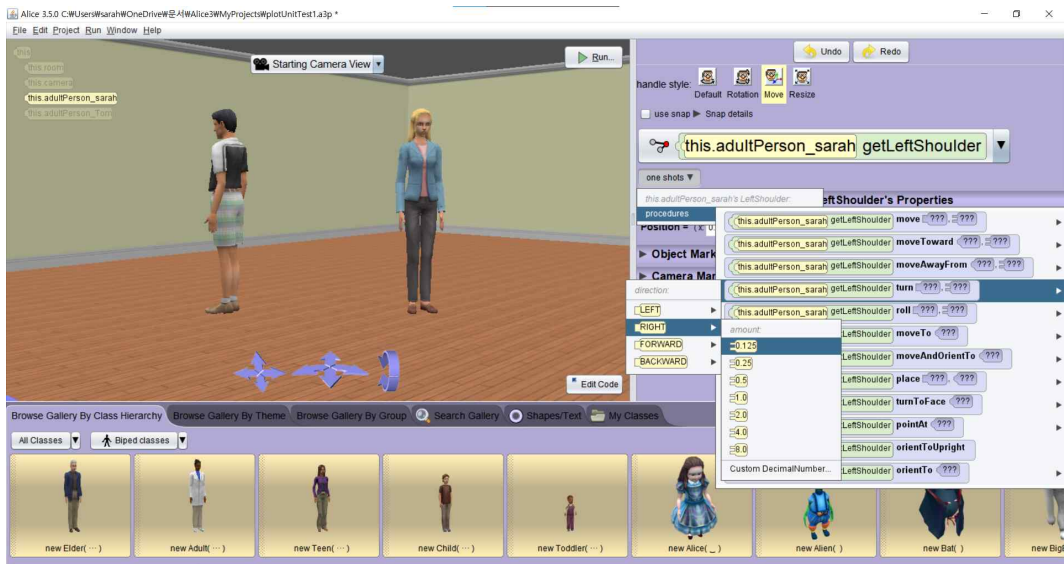


그림 31 getLeftShoulder의 Turn 프로시저를 이용한 왼팔의 움직임 설정



그림 32 왼팔 움직임 동작

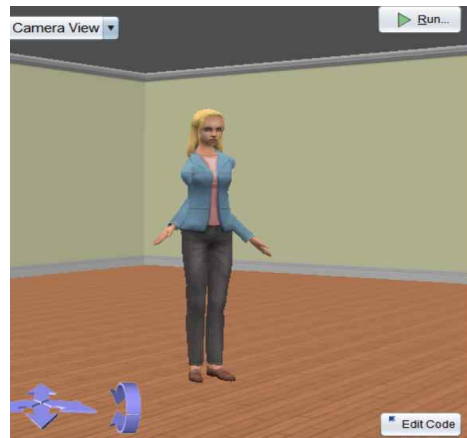


그림 33 프로시저 설정이 잘못되어 자연스럽지 못한 경우

적절하지 않은 프로시저를 사용하거나 Amount에 대입된 숫자가 적절하지 않은 경우 캐릭터의 움직임이 [그림 36]과 같이 무척 어색해지기 때문에 주의 하여야 한다.

팔을 올리기 위한 다른 방법으로 getLeftClavicle의 Roll 프로시저를 사용할 수 있다. Left를 0.125로 지정하면 [그림 34]와 같이 왼쪽 팔이 올라간 모습으로 캐릭터를 변형할 수 있다.

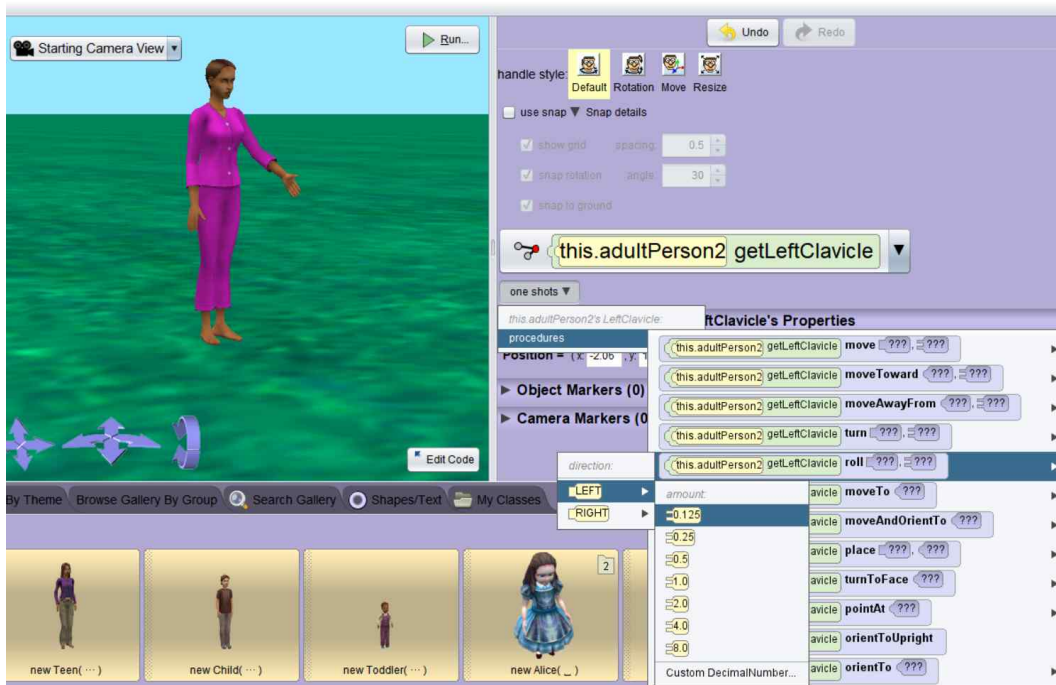


그림 34 getLeftClavicle의 Roll 프로시저를 이용한 팔 올리기

앨리스의 Joint를 이용하면 큰 관절의 움직임 뿐 아니라 디테일한 움직임도 표현이 가능하다. 예를 들면 getRightEye(getLeftEye)나 getRightEyelid-(getLeftEyelid)를 이용하여 눈의 크기를 조절할 수 있고 입모양도 변형이 가능하다. 손가락의 모양도 [그림 38]과 같이 getThumb, getThumbKnuckle, getIndexFinger, getIndexFingerKnuckle 등을 이용해 미세하게 제어할 수 있다.

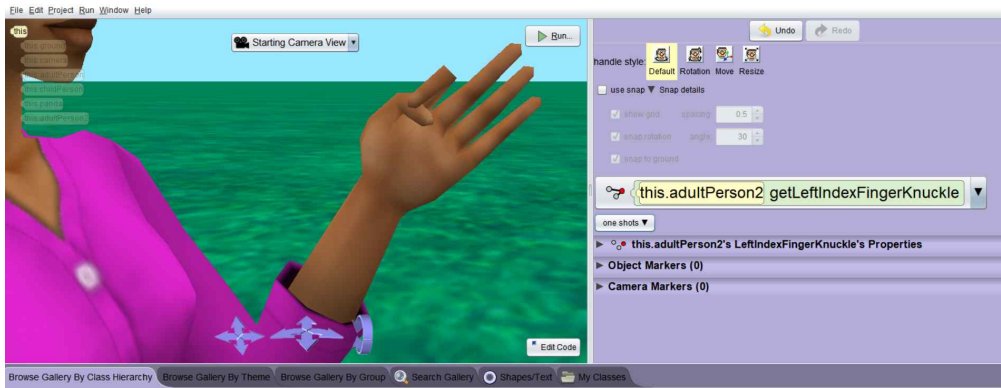


그림 35 Finger/FingerKnucle을 이용한 손가락 제어

앨리스에서 코드를 작성하기 위해서는 <Edit Code> 버튼을 누르고 'myFirstMethod' 탭을 선택하여 이 곳에 블록을 조립하여야 한다. [그림 36]

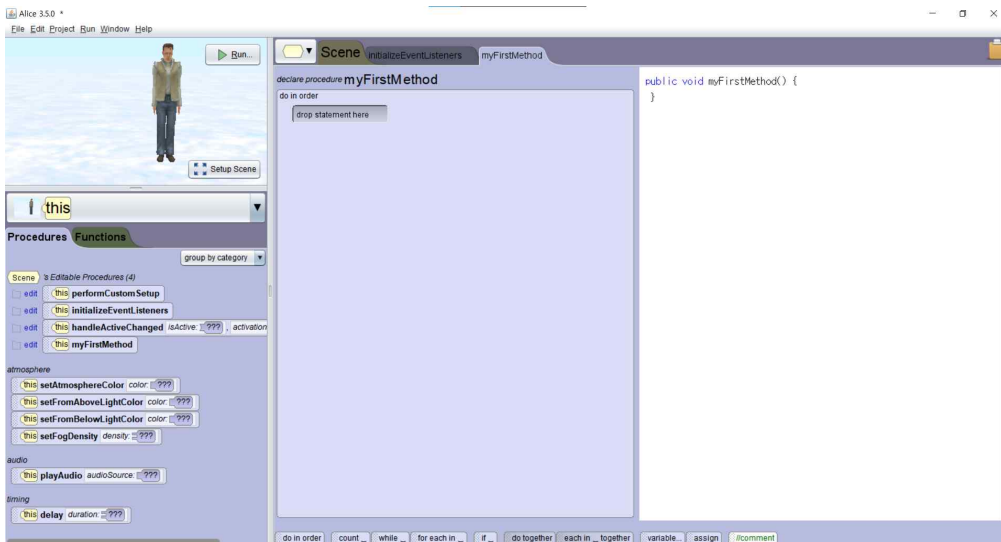


그림 36 myFirstMethod 탭

앨리스에서는 프로시저가 호출될 때마다 객체의 행동이 화면에 나타나게 된다. 이를 통해 시각적으로 관찰이 가능하며 객체지향 프로그래밍의 추상적인 원리를 이해하는데 도움을 준다.

또한 엘리스는 사용자가 블록을 편집기에 드래그 앤 드롭하여 개발하기 때문에 문법오류(Syntax Errors)를 발생시키지 않는다. 프로시저를 실행하는데 변수 할당이나 매개변수 값이 필요한 경우 자동으로 팝업 메뉴가 나타나기 때문에 오류를 범할 수 없다.

다만, 프로시저의 매개변수 값에 따라 객체의 움직임이 달라지는데 값에 대한 가이드 라인이 없기 때문에 직접 입력하여 객체의 움직임을 일일이 확인하여야 한다는 단점이 있다. 입력한 값에 따라 객체의 행동이 부자연스러워질 수 있고 의도한 결과를 얻기까지 작업시간이 오래 걸린다.

엘리스 객체의 모양이나 행동을 자연스럽게 제어하려면 수많은 시행착오와 무수한 시간을 필요로 한다. 이러한 제약이 자칫 스토리를 구성하는 데 방해가 될 수 있고 프로그래밍 학습보다는 캐릭터의 움직임에 중점을 둘 수 있다.

이러한 문제를 제거하고 학습자가 좀 더 스토리텔링과 프로그래밍 학습에 집중하여 보다 쉽게 애니메이션을 제작할 수 있도록 다양한 프로시저를 클래스로 만들어 제공하려고 한다.

나. Primitive Plot Units

소프트웨어는 일상생활과 밀접한 관련이 있다. 일상 생활 속의 문제점을 파악하여 소프트웨어를 만들어 해결하거나 필요한 것을 논리화 하여 코딩으로 작성할 수 있다. 필요한 것을 생각해 내고 코드로 작성하는 것 자체가 스토리텔링이라 할 수 있다.

스토리텔링은 콘텐츠인 Narratives(이야기)가 있고 Narratives에 등장하는 주인공(객체)들 사이에서 일어나는 상호작용을 묘사한다. 엘리스로 애니메이션을 구현하기 위해 Narratives(이야기)가 구성되면 시나리오를 만들고 도식화 하는 작업이 필요하다²¹).[그림 37]

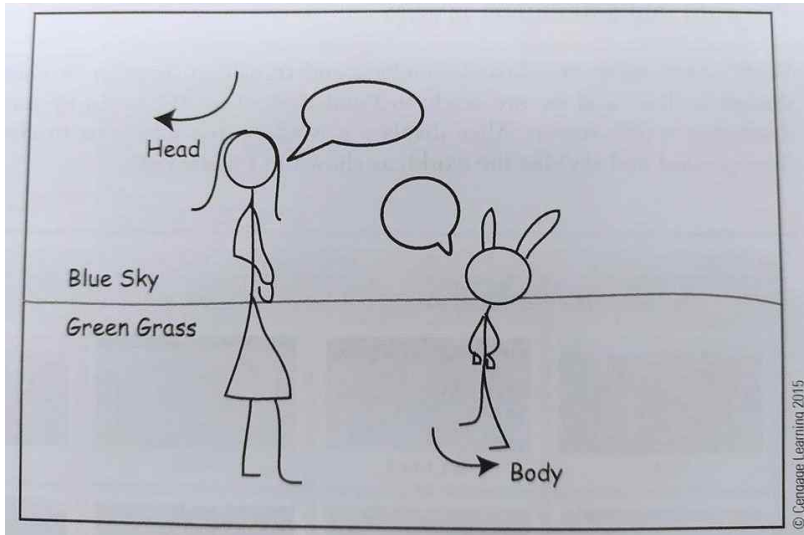


그림 37 스토리보드 스케치²¹⁾

스토리에 따라 애니메이션을 구성하다 보면 특정한 움직임이나 행동 패턴이 반복되는 경우가 있다. 등장인물의 행동이나 움직임을 통해 심상을 표현해야 하는 경우, 몇 가지 패턴을 저장해 놓고 필요할 때마다 사용하면 시간을 절약할 수 있고 좀 더 스토리에 집중할 수 있을 것이다.

이야기를 구현하는 방법에는 여러 가지가 있으나 그 중 Plot Units⁵⁾를 엘리스에 적용해 보고자 한다. Plot Units 중심으로 몇 가지 패턴을 엘리스에서 프로시저로 구성해 놓으면 애니메이션을 쉽게 완성할 수 있을 것이다.

Plot Units(줄거리 단위)는 Affect State(심상)란 실체로 구성되는데 객체의 마음상태를 나타낸다. 줄거리 단위는 [그림 40]과 같이 나타낼 수 있으며 등장인물의 관념적인 계획, 목적, 외부 사건에 대한 반응들을 나타낸다.²²⁾ 줄거리 단위는 2개의 심상을 연결한 Causal Links(인과관계 사슬)로 표현된다. 이 사슬은 두 심상의 관계를 자극(동기부여), 실현, 종료, 대등을 뜻하는 motivation(m), actualization(a), termination(t), equivalence(e) 등으로 구분한다.

[그림 40]에서 'M'이란 객체의 마음상태를 표시하는 심상이다. '+'는 기쁜 일이나 좋은 사건을 나타내는 심상이다. 반면 '-'는 나쁜 일이나 좋지 않은 사건을 나타내는 심상이다. 화살표는 인과관계 사슬을 표현한다.

[그림 38]에 Success라는 Plot Unit이 있다. 그림에서 ‘M’은 어떤 마음의 결정을 내린 상태와 행동을 나타내고, 그로 인해 좋은 일(‘+’)이 있어났음이 표현되어 있다. 이 둘의 관계는 마음가짐이 행동으로 표출되어 실현된 것이기 때문에 actualization(‘a’)이라는 인과관계 사슬로 표시한 것이다.

SUCCESS



그림 38 Success Plot Unit

Success Plot Unit의 예문인 “You ask for a raise and you get it.”을 살펴본다.

You ask for a raise and you get it.

M : you ask for a raise (월급인상을 원하다)

→ : and (그래서)

+ : you get it (바람이 이루어지다)

월급 인상을 요구한 ‘you ask for a raise’를 마음상태인 ‘M’으로 나타낼 수 있다. 그 바람이 이루어진 ‘you get it’이 좋은 사건을 나타내는 ‘+’로 표현되었다. 이 예문에서는 의도와 사건을 연결하는 인과관계 사슬 ‘a’는 ‘and’에 대응한다.

[그림 39]에 또 다른 Plot Unit인 Problem이 있다. 이는 좋지 않은 사건(‘-’)의 발생으로 말미암아(‘m’) 마음상태(‘M’)가 되었다는 것을 표현한다.

PROBLEM



그림 39 Problem Plot Unit

Problem Plot Unit의 예문인 ‘You get fired and need a job’을 살펴본다.

You get fired and need a job.

- : you get fired (해고를 당하다)

→ : and (그래서)

M : need a job (일이 필요하다)

‘you get fired’는 좋지 않은 사건이다. 나쁜 사건 발생으로 말미암아(‘m’) 직업을 구해야 한다는, 즉 ‘need a job’이라는 마음 상태(‘M’)이 되었다는 것을 표현한다.

Plot Units는 표현할 이야기의 복잡도에 따라 기본 유닛인 Primitive Plot Units와 복합적인 Complex Plot Units로 나눌 수 있다. 이 논문에서는 한 사람에게 일어나는 일을 나타내는 Primitive Plot Units의 경우만 다루기로 한다.

Primitive Plot Units의 예문을 앨리스로 구현하기 위해 각 문장을 Affect States 요소와 Casual Links 요소로 분해하였다. 또한 분해한 조각들이 앨리스를 통해 어떤 방식으로 표현될 수 있을지 간단한 시나리오를 함께 제시하였다. Primitive Plot Units는 대부분 한 사람에게 일어난 일을 도식화한 것이나, 다양하게 표현하기 위해 몇 가지 시나리오 예시에는 등장인물이 한 명 이상 있을 수 있다.

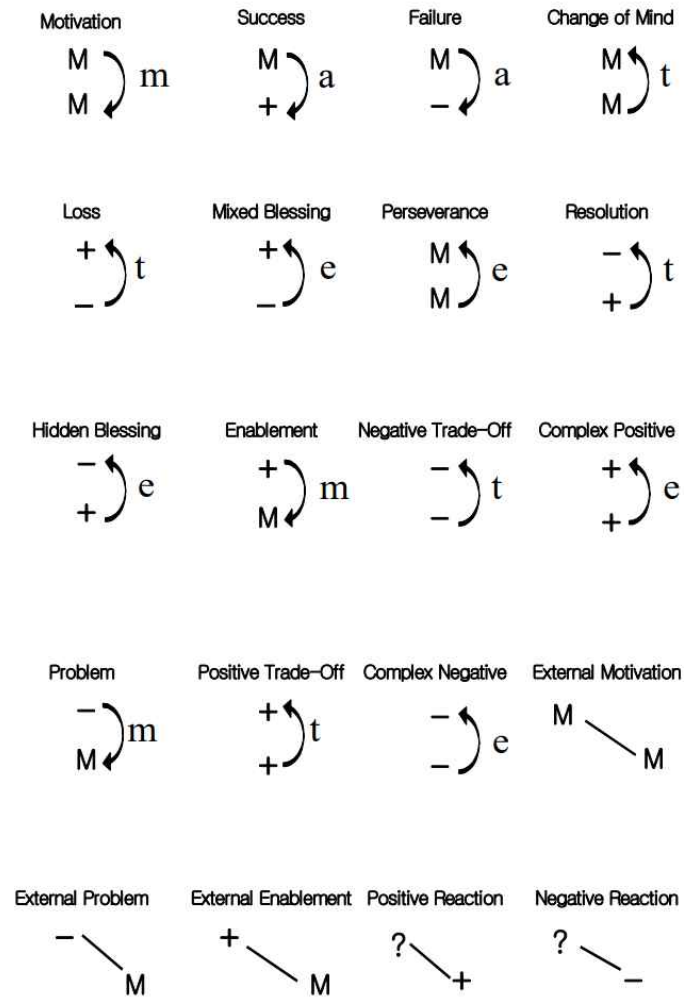
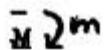


그림 40 Primitive Plot Units

1) Plot Units 예제 및 시나리오 구성 예시

1. Problem

PROBLEM



- ▶ You get fired and need a job.

- : you get fired

→ : and

M : need a job

▶ 시나리오 1 :

사람1 - "I am fired"라고 말한다. (-)

사람1 - 머리를 숙이고 손으로 머리를 감싼다. (-m->)

사람1 - 고개를 들고 "I need a job" 말한 후 걸어나간다. (M)

▶ 시나리오 2 :

사람1 - "중간고사 망쳤다"라고 말한다. (-)

사람1 - 머리를 숙이고 손으로 머리를 감싼다. (-m->)

사람1 - "기말고사를 잘봐야지"라고 말한다. 주먹을 불끈 쥐어보인다. (M)

2. Success

SUCCESS



▶ You ask for a raise and you get it.

M : you ask for a raise

→ : and

+ : you get it

▶ 시나리오 :

사람1 - "I need a raise" 라고 말한다. (M)

(한쪽 무릎을 꿇고 손을 모음.)

사람1 - I get a notice (-a->) (편지를 받음 / 눈을 크게 뜸)

사람1 - 뒤돌아서서 한 손을 올리고 뛰어간다. (+)

(또는 점프하면서 양팔을 올린다.)

3. Failure

FAILURE



- ▶ Your proposal of marriage is declined.

M : your proposal of marriage

→ : (proposal)

- : is declined

- ▶ 시나리오 :

남자1 - "Will you marry me?"라고 말한다. (M) (무릎꿇고 손을 모은다)
: Success의 첫 번째 심상과 동일한 행동표현

남자1 - "No"라고 들었다. (-a->)

남자1 - 머리를 숙이고 손으로 머리를 감싼다. (-)
: Problem unit의 (-)와 동일한 행동 표현

4. Resolution

RESOLUTION



- ▶ You fix a flat tire after a blow out.

- : a blow out

→ : after

+ : you fix a flat tire

- ▶ 시나리오 :

자동차 - 화면 가운데로 오다가 멈춘다. (-)

사람1 - 바퀴 옆에 앉았다가 일어선다. (-t->)

자동차 - 다시 움직인다. (+)

사람1 - OK/Yes 행동을 취한다. (손가락으로 OK 사인을 함/주먹을 쥐고 팔을 올렸다가 살짝 끌어내리는 동작)

5. Loss

LOSS

+ 5 t
- 5 t

- ▶ The woman you love leaves you.
+ : you love the woman
→ : (leaves)
- : the woman you love leaves you

- ▶ 시나리오 1 :
 - 남자 - 사람1을 보고 꽃(소품)을 전달하면서 "I love you"라고 말한다. (+)
 - 여자 - 남자를 보며 "Sorry" 라고 말하고 뒤돌아서 사라진다. (-)
 - 남자 - 여자가 뒤돌아가면 고개를 숙이고 좌우로 흔든다. (-t->)

- ▶ 시나리오 2 :
 - 남자 - 기분 좋은 상태 (배부른/행복한)로 있다. (+)
 - 남자 - 전화를 받았다 (유통기간을 확인하였다.) (-t->)
 - 남자 - 사랑하는 사람이 떠났다. (먹은 음식이 유통기간이 지난것이였다.) (-)

6. Pos. Trade-off

POS. TRADE-OFF

+ 5 t
+ 5 t

- ▶ You buy a new Toyota and then inherit a Porsche.
+ : you buy a new Toyota
→ : and then
+ : inherit a Porsche

- ▶ 시나리오 1 :
 - 사람1 - 차 옆에서 차키를 손에 들고 머리와 몸을 좌우로 흔든다.
기분좋은 상태로 있다. (+)
 - 사람1 - 좋은 소식을 듣는다. (편지가 날아옴) (-t->)

자동차 – 사람1 앞에 선다.
사람1 – 양팔을 들고 점프를 몇 번 한다. (+)
(Success와 동일)

▶ 시나리오 2 :

사람1 – 사람2에게 토요타 자동차 키를 건넨다. (+)
사람2 – 키를 받으며 춤추듯이 머리와 몸을 좌우로 흔든다. (+)
사람3 – 포르쉐 자동차 열쇠를 사람2에게 건넨다. (-t->)
사람2 – 양팔을 들고 점프를 몇 번 한다. (+)
(Success와 동일)

7. Neg. Trade-off

NEG. TRADE-OFF

- 5t

▶ You get fired so you don't have to take a lousy job assignment.
- : you get fired
→ so you don't have to
- : take a lousy job assignment

▶ 시나리오 1 :

책상 위에 종이와 책이 올려져있다.
사람1 : 나쁜 소식을 듣는다. (편지가 전해진다) (-)
사람1 : 머리를 숙이고 손으로 머리를 감싼다. (-)
사람1 : 책상위의 종이와 책을 손으로 밀어버린다.
(lousy job assignment를 수행할 필요가 없다.) (-t->)

▶ 시나리오 2 :

책상 위에 종이와 책이 올려져있다.
사람1 : 걸어들어와 "You are fired" 라고 말하고 뒤돌아 나간다. (-)
사람2 : 머리를 숙이고 손으로 머리를 감싼다. (-)
사람2 : 책상위의 종이와 책을 손으로 밀어버리고(-)
고개를 숙이고 걸어나간다. (-t->)

8. Perseverence

PERSEVERENCE

M M 5 e

- ▶ You reapply to Yale after being rejected.

M : you reapply to Yale

→ : after

M : being rejected

- ▶ 시나리오 :

사람1 - 'Rejected'라고 써있는 봉투를 전달받는다.

팔을 늘어뜨리고 고개를 숙인다.(M)

사람1 - 잠시 눈을 감고 있다가 고개를 들고 눈을 뜬다. (-e->)

받은 봉투를 버린다. 눈을 크게 뜬다.

'Reapply'라고 적힌 종이를 어깨높이로 들어올린다. (M)

9. Hidden Blessing

(나쁜 사건 속에 좋은 사건이 숨어 있다는 의미)

HIDDEN BLESSING

+ 5 e

- ▶ You sprain an ankle and win damages.

+ : win damages

→ : and

- : you sprain an ankle

- ▶ 시나리오 :

사람1 - 걸어가다가 넘어져 발목이 꺾인다. 고개를 숙이고 앉아있다. (-)

사람2 - 'insurance - damage'라고 써있는 봉투를 본다. (+)

사람1 - 고개를 들고 한쪽팔을 높이 올린다. (-e->)

10. Mixed Blessing

(좋은 사건에 나쁜 사건이 숨어있었다는 의미 - 세웅지마)

MIXED BLESSING

+ ↘ e
- ↘

▶ You buy a car and it turns out to be a lemon.

- : it turns out to be lemon

→ : and

+ : you buy a car

▶ 시나리오 :

사람1 - 자동차 키를 들고 점프를 두 번 한다. (+)

자동차 - 앞으로 조금 움직이다가 멈춘다. (-)

사람1 - 어깨를 늘어뜨리고 고개를 숙인다. (-e->)

11. Change of Mind

CHANGE OF MIND

M ↘ t
M ↘

▶ You want to buy a car but decide against it.

M : You want to buy a car

→ : but

M : decide against it

▶ 시나리오 :

사람1 - 차가 많이 있는 쪽(딜러샵)으로 걸어간다.

'I want to buy a car'라고 생각한다.(말풍선) (M)

사람1 - 차 앞에서 잠시 멈추고 머리를 한쪽으로 기울인 채 손가락으로 턱을 만진다. (마음을 고쳐먹었다) (M)

사람1 - 손을 내리고 돌아서 왔던 길로 걸어간다. (but) (-t->)

12. Motivation

MOTIVATION



- ▶ You want to reach a client so you call him.

M : you want to reach a client

→ : so

M : you call him

- ▶ 시나리오 :

사람1 - 책상 앞에서 한손으로 턱을 괴고 있다. (Change of Mind와 동일)

'client'라는 말풍선을 나타낸다. (M)

사람1 - 전화기를 들어 올린다. (-m->)

사람1 - 전화기를 귀에 대고 'Hello'라고 말한다. (M)

13. Enablement

ENABLEMENT



- ▶ You receive a book and decide to read it.

+ : you receive a book

→ : and

M : decide to read it

- ▶ 시나리오 :

사람1 - 책상 위에 있는 책을 쳐다본다. (+)

사람1 - 책을 들어올린다. (-m->)

사람2 - '읽어보야지'라고 생각하면서(말풍선) 책을 펴본다. (M)

14. Complex Positive Event

COMPLEX POS. EVENT

+ } e
+

- ▶ A gift is indicative of close friendship.
 - + : a gift (giving a gift)
 - : is
 - + : indicative close friendship

- ▶ 시나리오 :
 - 사람1 - 'For you, from your friend'라고 쓰여 있는 선물을 발견한다. (+)
 - 사람1 - 선물을 들고 눈을 크게 뜬다. (-e->)
 - 사람1 - 친구를 생각한다. (+)

- ▶ 시나리오 :
 - 사람1 - 사람2에게 선물을 건넨다. (+)
 - 사람2 - 선물을 받으며 눈을 크게 뜬다. (→)
 - 사람1을 향해 두팔을 벌리고 포옹한다. (+)

15. Complex Negative Event

COMPLEX NEG. EVENT

- } e
-

- ▶ Your house burns down and you aren't covered.
 - : your house burns down
 - : and
 - : you aren't covered

- ▶ 시나리오 :
 - 집 - 불에 타고 있다. (-)
 - 사람1 - 보상이 안된다는 소식을 듣는다. (-e->)
 - 사람1 - 좌우로 왔다 갔다 하다가 머리를 두 손으로 감싼 후 주저앉는다. (-)

2. 엘리스 라이브러리 구축 및 활용

Primitive Plot Units의 예문을 엘리스로 구현하기 위해 각 문장을 분해하고 엘리스를 통해 표현될 수 있도록 시나리오를 제시하였다. 그런데 시나리오를 살펴보면, 공통된 표현이나 자주 사용하는 유닛이 등장한다는 것을 알 수 있다. 엘리스의 객체는 Joints를 통해 세부적으로 나누어져 있기에 섬세한 표현이 가능하지만 이를 자연스럽게 구현하기 위해서는 상당한 시간이 걸린다는 단점이 있다.

이런 반복적인 작업을 효율적으로 수행하기 위해 Plot Units의 예제를 분석하여 자주 사용되는 동작을 프로시저로 작성해 재활용 할 수 있도록 라이브러리화 하는 방법을 고안하였다.

엘리스는 기본 클래스(캐릭터)가 다양하고 그에 따른 프로시저도 풍부하게 제공되고 있다. 프로시저는 크게 두 가지로 나눌 수 있는데 엘리스에서 제공하는 프로시저와 사용자가 지정하는 프로시저이다. 엘리스에서 기본으로 제공하는 프로시저는 모든 클래스에서 공통으로 사용할 수 있는 프로시저와 특정 클래스에서만 사용할 수 있는 프로시저가 있다. 사용자 정의 프로시저의 경우에는 특정 클래스에서만 사용할 수 있다.

가. 프로시저 작성하기

앞에서 Plot Units의 예문을 Affect States, Casual Links, 심상의 관계에 따라 분해하였다. 분해한 조각들은 엘리스에서 애니메이션을 구현하기 위한 행동을 표현할 수 있는 ‘작은 조각’이라 할 수 있다. 이 작은 조각을 하나의 ‘프로시저’¹⁾로 정의하여 애니메이션에 사용하려고 한다.

예문 중 ‘Success’에 해당하는 장면을 만들어 보려고 한다. 해당 시나리오는 다음과 같다.

1) 이전 버전인 Alice2에서는 ‘메소드’라 지칭하였다.

- ㉠ 사람1 - "I need a raise" 라고 말한다. (M)
(한쪽 무릎을 꿇고 손을 모음.)
- ㉡ 사람1 - I get a notice (-a->) (편지를 받음 / 눈을 크게 뜸)
- ㉢ 사람1 - 뒤돌아서서 한 손을 올리고 뛰어간다. (+)
(또는 점프하면서 양팔을 올린다.)

앨리스에서 배경이나 등장인물을 설정하는 부분은 [그림 19]부터 [그림 29]를 참고하여 작성하였다. 또한 시나리오 중 대사가 있는 경우 앨리스의 기본 프로시저인 'say' 프로시저를 이용하면 쉽게 구축할 수 있기 때문에 생략하도록 하겠다. [그림 41]

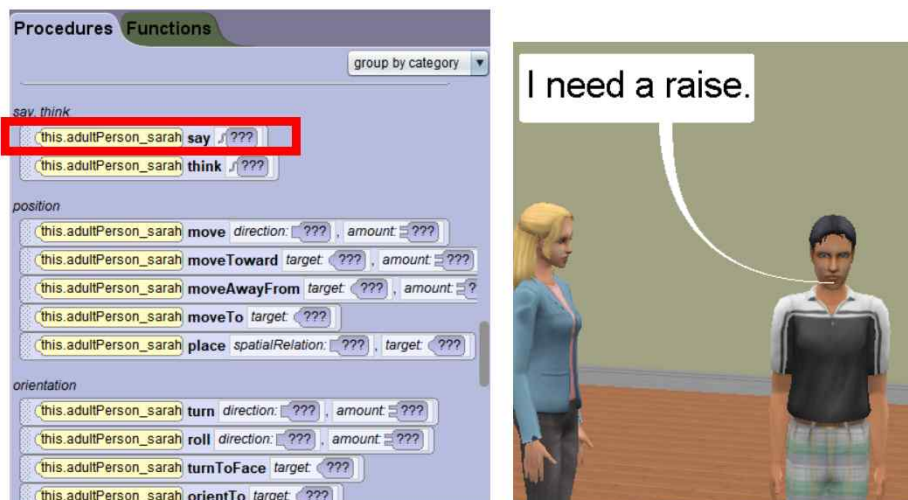


그림 41 앨리스 기본 프로시저 : say

시나리오 ㉠의 첫 번째 행동인 '한쪽 무릎을 꿇고 손을 모음'을 프로시저로 만들어 보려고 한다. 먼저 이 프로시저를 어떤 클래스 수준에서 사용할지 결정해야 한다. 가장 상위 클래스인 Biped에 프로시저를 만들어 놓으면 Biped 클래스 및 하위 수준의 모든 클래스로 만들어진 객체에서 사용이 가능하다. 만약 Biped의 바로 아래 수준의 Person 클래스에 프로시저를 작성하면 Person 클래스와 하위 하위 수준의 클래스로 만들어진 객체에 적용이 가능한 프로시저가 만들어진다.

Biped에서 적용이 가능한 프로시저를 만들기 위해 [그림 42]의 <Edit Code> 버튼을 누르면 코드를 작성할 수 있는 화면이 나타난다. [그림 43]



그림 42 Edit Code 버튼

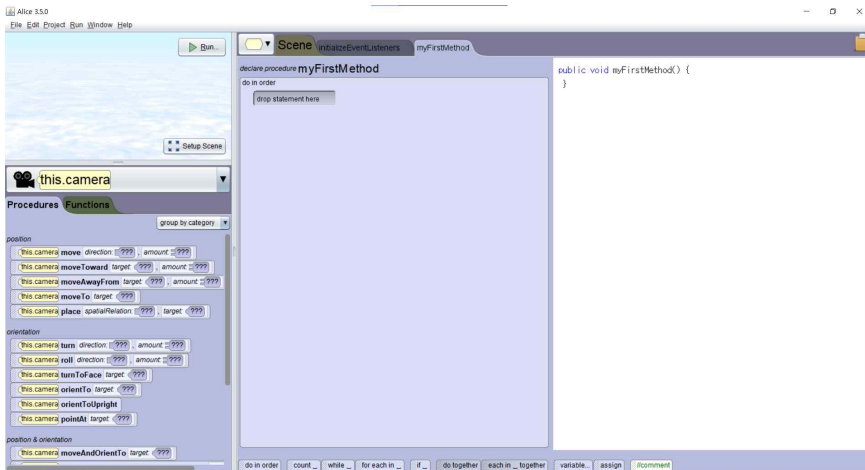


그림 43 Editing Area

[그림 44]와 같이 클래스 네비게이터 버튼을 누른 후 'Biped-Add Biped Procedure...'을 선택한다. 작성할 프로시저의 이름을 입력하고 <OK> 버튼을 누르면 프로시저를 작성할 준비가 된 것이다. 여기에서는 'success_ask_for'라는 이름을 부여하였다. [그림 45]

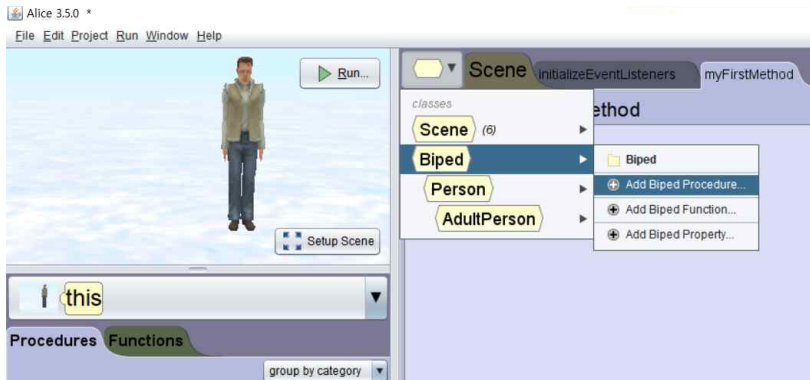


그림 44 Biped 프로시저 만들기

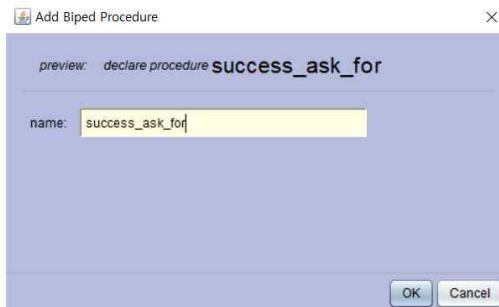


그림 45 프로시저 이름 정하기

<OK> 버튼을 누르면 [그림 46]과 같이 프로시저 탭으로 이동하게 되는데 이곳에 시나리오에 따라 코드를 작성한다.

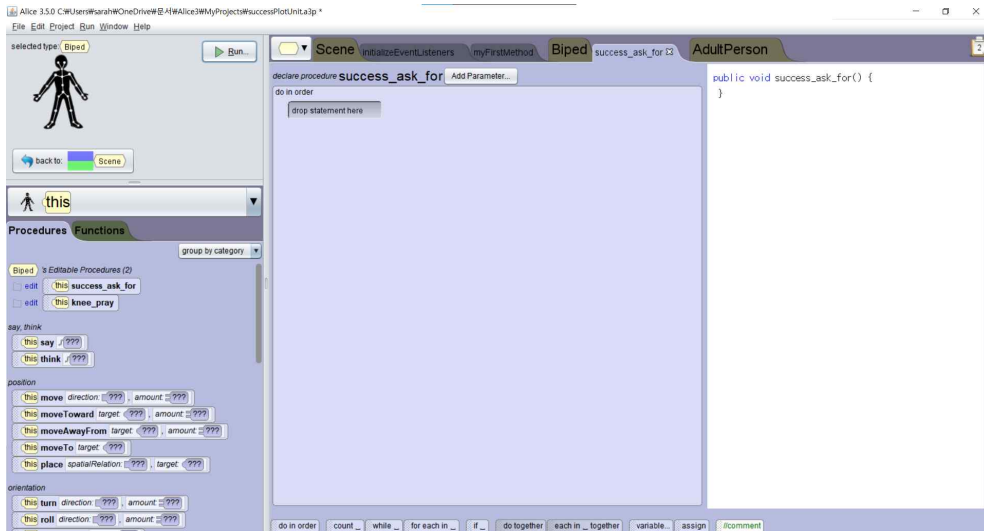


그림 46 Biped 프로시저 탭 - Code Editor

‘한쪽 무릎을 꿇고 손을 모음’의 동작을 나타내기 위해 다음과 같이 동작을 나누어 볼 수 있다.

- 한쪽 무릎을 구부린다.
- 다른쪽 무릎을 꿇는다.
- 객체를 아래 방향으로 이동시킨다.
- 양팔을 올린다.
- 손을 모아 손가락을 구부린다.

나누어진 동작들을 앨리스에 적용해 다음과 같은 코드를 만들 수 있다. [그림 47]

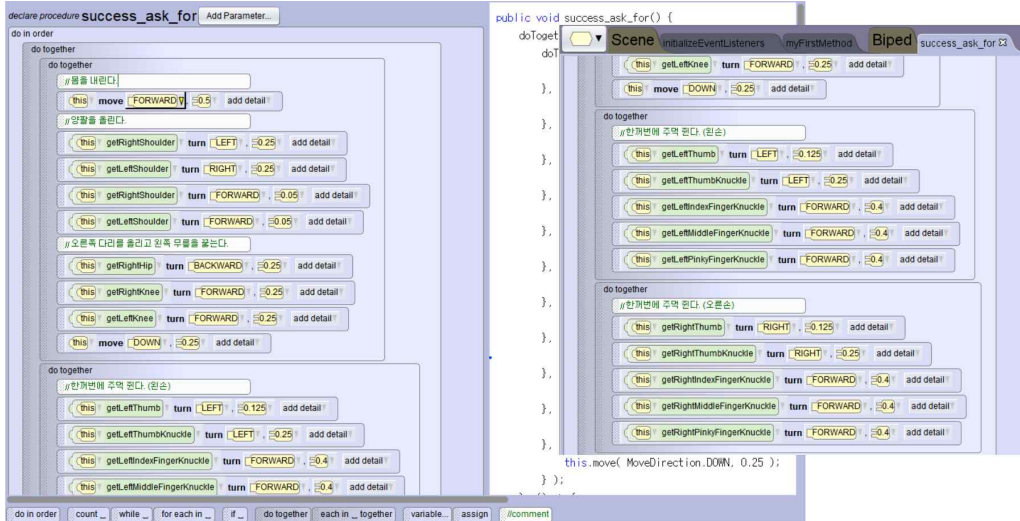


그림 47 프로시저 'success_ask_for' 코드

코드가 완성되면 이 프로시저를 사용하기 위해 'myFirstMethod' 탭으로 이동한다. 프로시저를 적용할 객체를 선택하면 [그림 48]과 같이 'success_ask_for' 프로시저가 나타난 것을 볼 수 있다. 이 블록을 오른쪽 에디터 쪽으로 드래그 앤 드롭한 후 <Run> 버튼을 누르면 방금 만들었던 프로시저가 실행되는 것을 볼 수 있다. [그림 49]

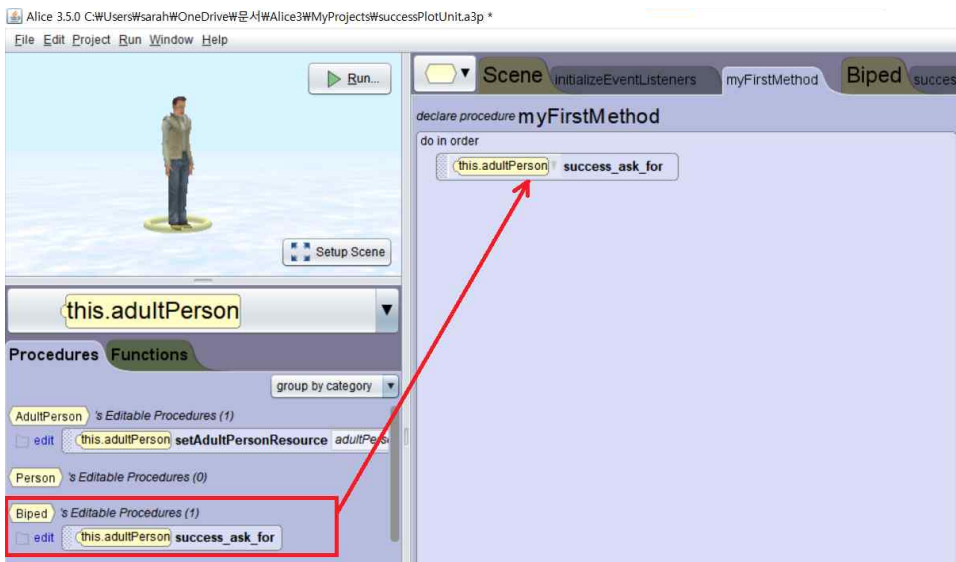


그림 48 myFirstMethod에서 사용 가능한 사용자 지정 프로시저

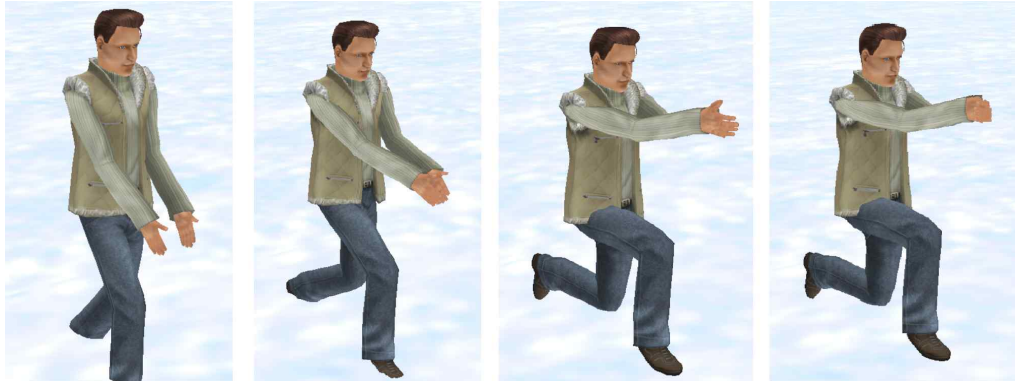


그림 49 success_ask_for 실행 화면

시나리오 중 ㉔에 해당하는 ‘점프하면서 양팔을 올린다’에 해당하는 동작을 만들기 위해 마찬가지로 다음과 같이 단계를 나누어 보았다.

- 몸을 위로 움직인다.
- 양팔을 위로 올린다.
- 무릎을 구부린다.
- 원래 위치로 되돌린다.

나누어진 동작들을 ‘Jump_up’이라는 프로시저로 [그림 50]과 같이 코드화 하였다.

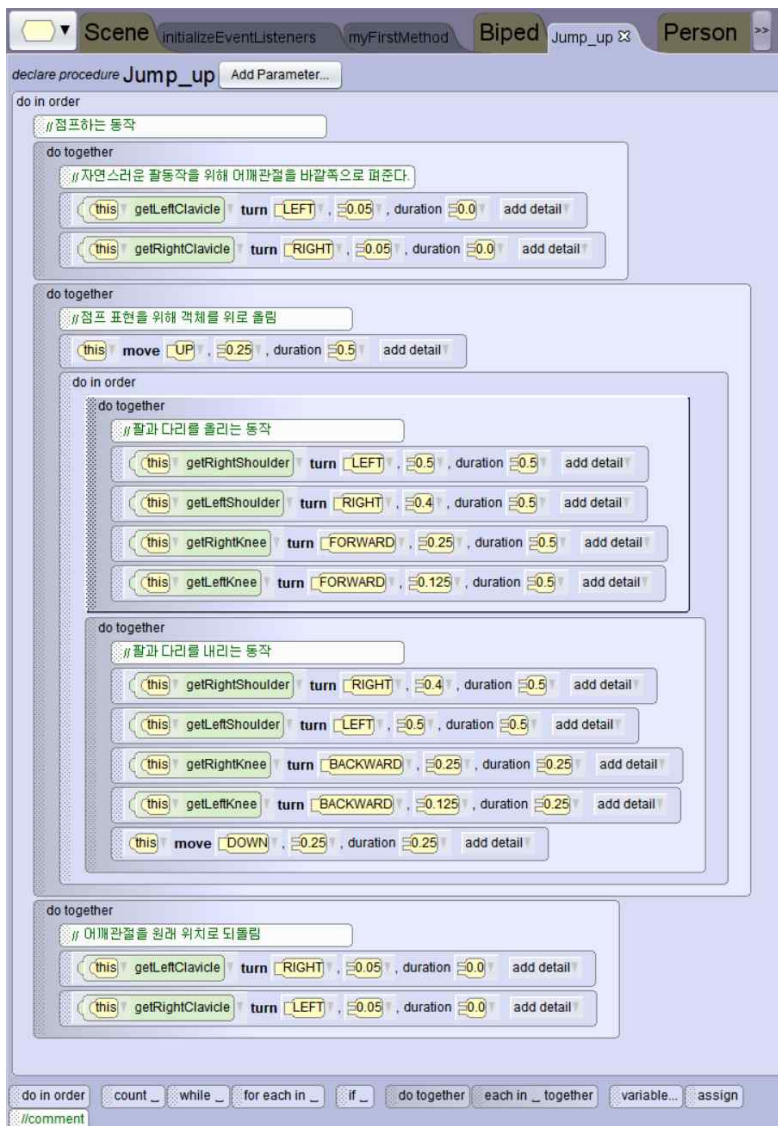


그림 50 프로시저 'Jump_up' 코드

이렇게 만들어진 프로시저를 AdultPerson 클래스로 만들어진 두 개의 객체에 적용하여 앞과 옆에서 관찰할 수 있도록 배치하였다. 이를 실행하면 다음 [그림 51]과 같이 동작하는 것을 볼 수 있다.



그림 51 Jump_up 프로시저 실행화면

마찬가지로 Plot Units의 1번 ‘Problem’의 시나리오 (-)를 나타내는 행동 표현인 ‘한 손으로 이마를 감싸며 고개를 젓는다’의 코드와 실행동작은 [그림 52]와 [그림 53]과 같이 나타내었다.

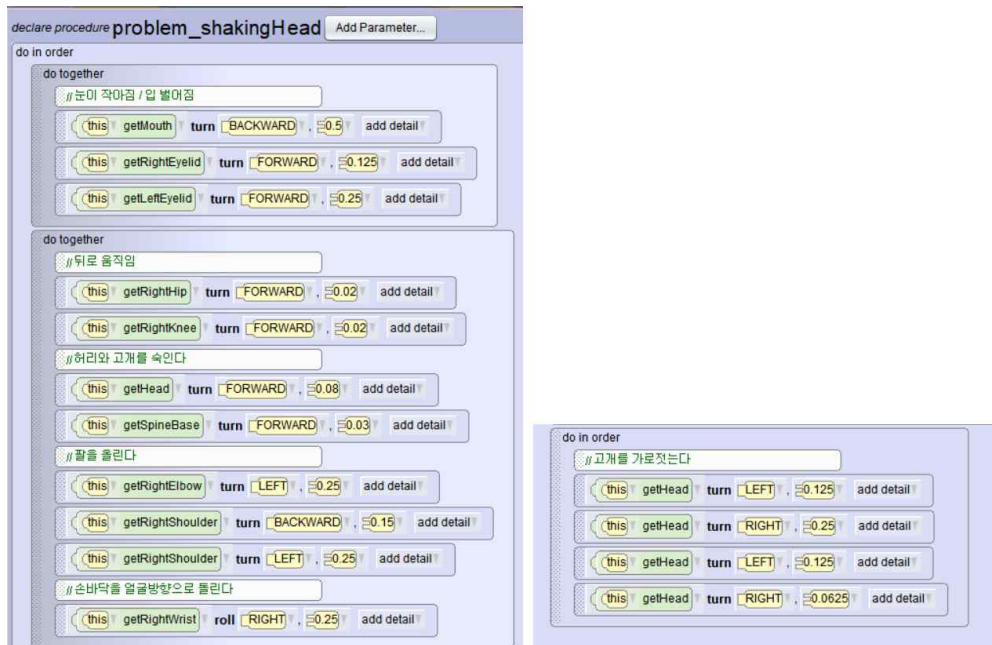


그림 52 problem_shakingHead 프로시저 코드



그림 53 problem_shakingHead 프로시저 실행화면

Primitive Plot Units를 통해 한 사람의 마음상태를 표현한 동작을 주로 구현하였으나 [그림 54]와 같이 두 사람의 행동도 표현이 가능하다. Complex Positive Event의 두 번째 시나리오를 참고하였다.



그림 54 Complex Positive Event : walk_hug 프로시저

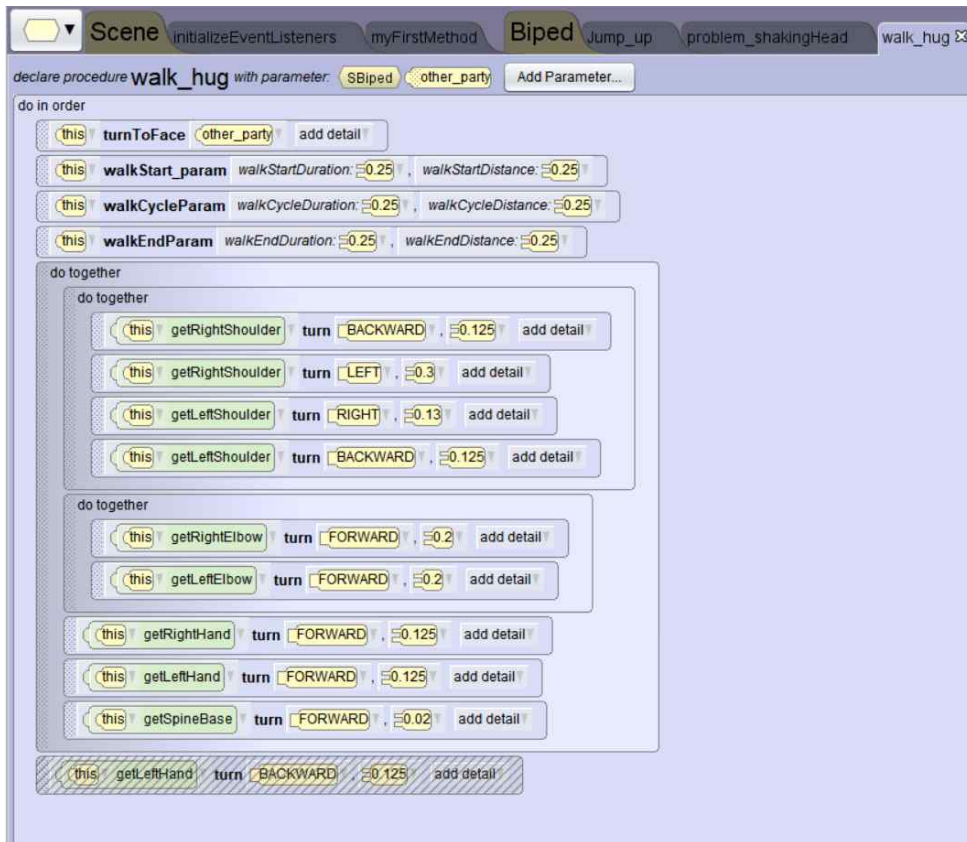


그림 55 walk_hug 프로시저 코드

나. 클래스 저장/불러오기

한 개 이상의 프로시저를 만들었다면 프로시저를 클래스 파일로 저장하여 다른 월드의 객체에서 불러올 수 있다. 확장자는 .a3c 이며 여기에 제시된 프로시저들은 앨리스의 Biped 객체 이하에 적용할 수 있다.

‘Biped’ 탭으로 이동하여 <Save to Class File..> 버튼을 누르면 .a3c 파일로 저장할 수 있다. [그림 56]

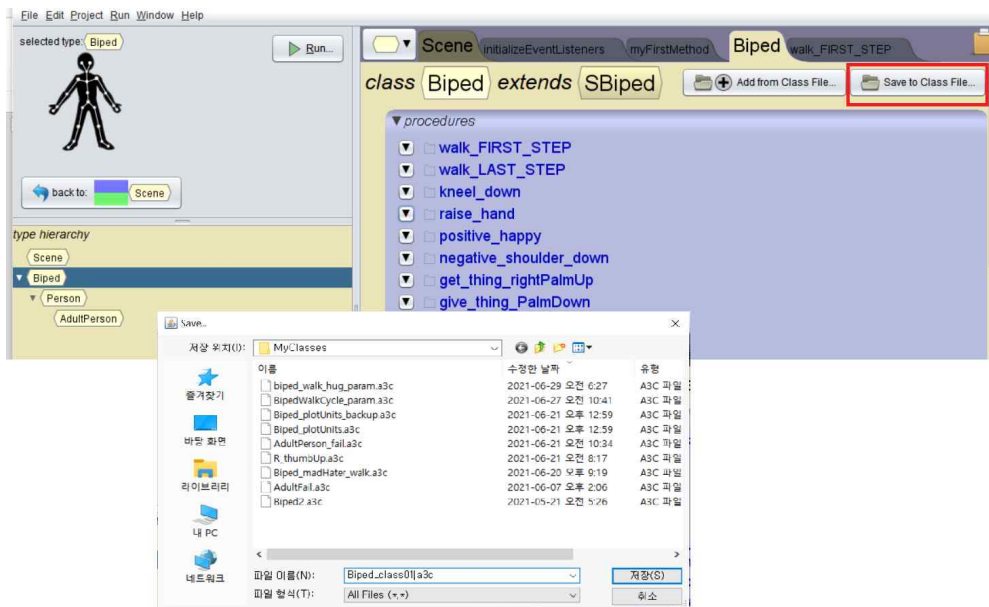


그림 56 프로시저를 클래스 파일로 저장하기

이렇게 저장된 클래스를 불러오기 위해 새로운 월드에 Biped 객체를 생성한 후 <Add from class File...> 버튼을 누른다[그림 57]. 확장자가 .a3c인 클래스 파일을 선택하면 적용 가능한 프로시저를 선택할 수 있는 창이 나온다. [그림 58][그림 59]

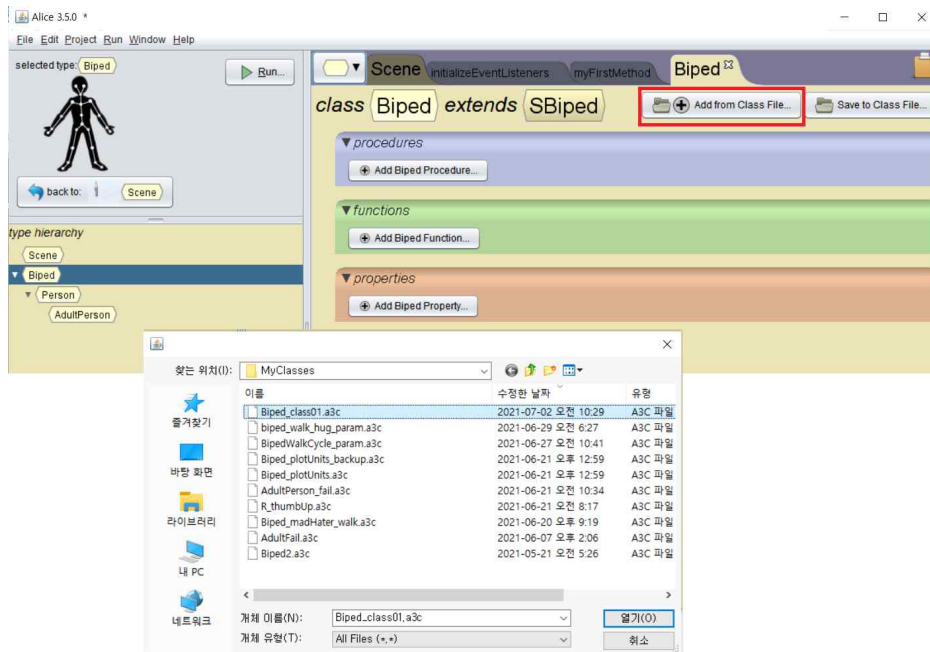


그림 57 프로시저 클래스 불러오기

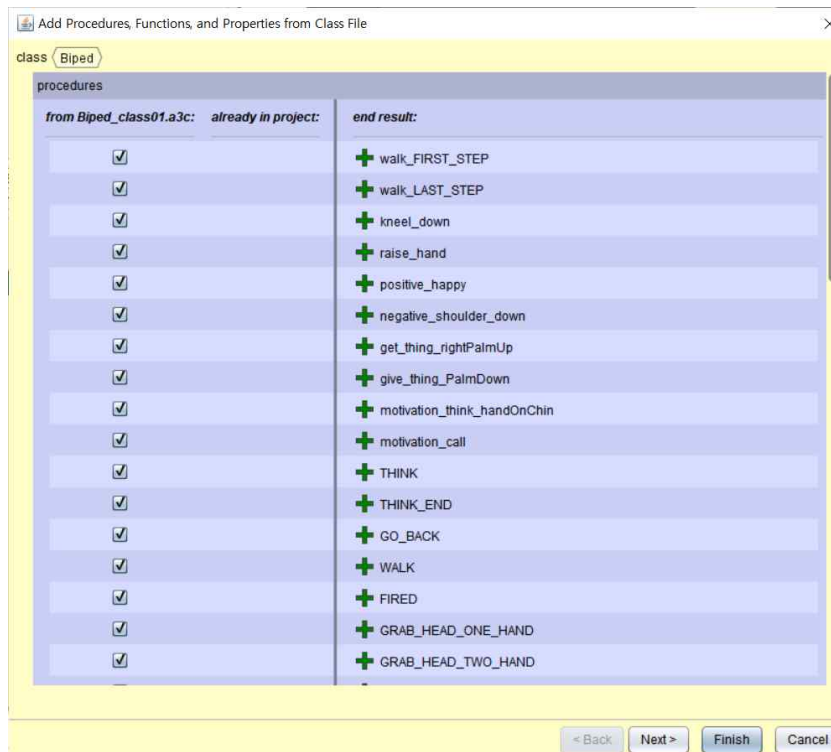


그림 58 프로시저 선택창

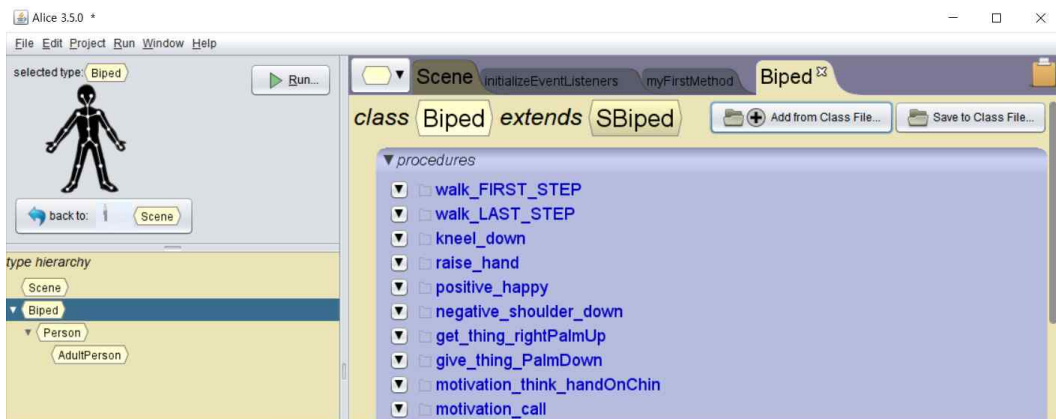


그림 59 클래스 파일로 불러온 프로시저 목록

‘myFirstMethods’ 탭으로 이동하여 객체를 살펴보면 사용자 지정 프로시저가 추가된 것을 알 수 있다.

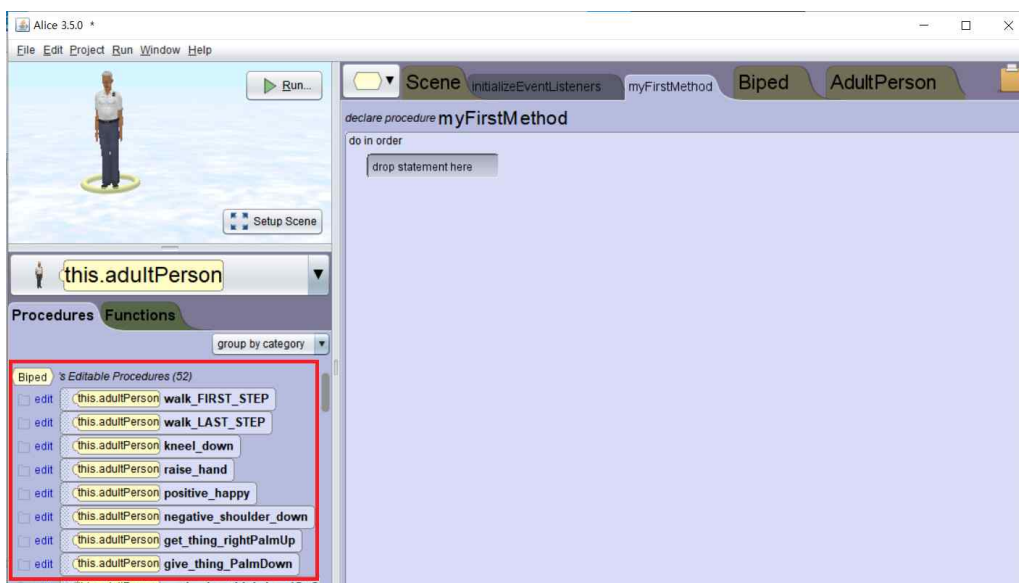


그림 60 Biped 객체에 적용된 프로시저

클래스 파일을 불러왔을 때 프로시저가 제대로 동작되는지 알아보기 위해 새로운 프로젝트에 Biped 객체를 생성한다. 비교를 위해 elderPerson, adultPerson, teenPerson의 세 객체를 생성하여 프로시저를 적용해 [그림 61]

과 [그림 62]와 같이 실행되는 것을 볼 수 있다. 각각의 객체는 다르지만 앞에서 보여진 예시와 동일한 동작이 구현되는 것을 확인할 수 있다.



그림 61 success_ask_for과 Jump_up 프로시저 실행동작



그림 62 walk_hug 프로시저 실행동작

Plot Units의 시나리오에 따라 세부 행동을 프로시저로 구현하면 여러개의 프로시저를 저장할 수 있다. 이를 클래스 파일로 저장하면 소규모의 Primitive Plot Units 라이브러리를 구축하게 되는 것이다.

사용자는 이 라이브러리를 불러올 때 필요한 프로시저만 선택적으로 불러올 수 있고 이를 객체에 적용시키는 방법으로 복잡한 동작들을 쉽게 재사용 할 수 있다.

IV. 결론 및 향후 연구

본 논문에서는 ‘프로그래밍’ 교육은 ‘글쓰기’ 교육이라는 주제로 스토리텔링을 활용한 프로그래밍 교육을 제시하였다. 또한 스토리텔링 기법을 적용하기 위한 여러 가지 교육용 도구를 비교하여 Alice가 최적의 도구임을 뒷받침 하였다.

프로그래밍을 통해 3D 객체의 움직임을 관찰하며 프로그래밍을 학습할 수 있는 도구로서 앨리스는 큰 의미를 갖는다. 더불어 스토리텔링 기법을 적용한다면 학습자 스스로 이야기를 기획하고 그 아이디어를 쉽게 구현해 볼 수 있다. 스토리텔링을 통해 창의적인 문제해결 능력과 프로그램의 기초 개념을 기를 수 있다고 여겨진다.

이를 구현하기 위한 앨리스의 기본 구조인 월드와 객체에 대해 소개하고 장면을 구성하는 방법을 살펴보았다. 또한 월드를 구성하는 객체를 다양한 클래스 컬렉션을 통해 생성하고 프로시저를 통해 움직임을 제어하는 방법을 소개하였다. 프로시저와는 별개로 다양한 객체 표현 방식을 통해 이야기를 풍성하게 꾸밀 수 있도록 속성을 적용하는 방법과, 객체 관절의 특이성과 제어 방법에 관한 몇 가지 방법을 기술하였다.

스토리텔링을 구현하는 방법에는 여러 가지가 있는데 그 중에서 스토리보드를 활용한 방법을 소개하였다. 스토리보드로 전체적인 줄거리를 기획하고 세부적인 시나리오를 작성하는 방법을 제시하였다. 시나리오 작성을 위해 줄거리 단위인 Plot Units에 관하여 소개하였고 이를 바탕으로 문장을 Affect States 요소와 Casual Links 요소로 분해하였다.

앨리스에 적용하기 위한 시나리오를 작성하였으며 시나리오의 내용은 15가지 Primitive Plot Units 예제를 활용하였다. 이를 바탕으로 등장인물 간의 내부 및 외부 상황을 앨리스를 통해 등장인물(객체)의 행동으로 구현하였다. 등장인물의 심상을 앨리스에서 나타내기 위하여 조건문과 제어문 및 기본 프로시저를 조합하여 코드를 완성하였다.

구현된 코드는 클래스 기반 프로시저로 저장하여 재활용할 수 있도록 하였다. 이렇게 만들어진 프로시저를 한 개의 클래스 파일로 저장하여 라이브러리를 구축하였으며 이를 다른 프로젝트 및 객체에 적용한 사례를 제시하였다.

텍스트 기반 언어를 통한 프로그래밍 교육은 문법을 학습하고 오류를 수정하는 데 많은 시간이 소요된다. 반면 Alice를 통해 애니메이션을 작성하면 문법 오류 없이 객체지향 개념을 자연스럽게 습득할 수 있을 것이다. 또한 이 논문에서 제시한 라이브러리를 활용한다면 등장인물의 심상과 행동을 표현하는데 소요되는 시간을 단축하여 프로그래밍 교육 자체에 더 집중할 수 있을 것으로 기대된다.

이번 연구에서는 Primitive Plot Units만을 활용하여 등장인물 한 사람의 움직임을 표현하는 라이브러리를 제작하였다. 이후에는 Complex Plot Units를 기반으로 라이브러리를 확대할 것이다. Complex Plot Units를 적용하면 훨씬 다양한 스토리를 구현할 수 있을 것으로 여겨진다.

이 논문에서는 라이브러리 구축에 초점을 맞추었기 때문에 실제로 학습자가 얼마만큼의 시간적 효과를 거두었는지에 대한 연구가 부족하다. 추후 교육을 통한 효과를 검증할 수 있는 기회를 만들어 연구를 이어가고자 한다.

앨리스를 활용한 스토리텔링 기반 프로그래밍 교육이 기초 학습자에게 효과적이라는 것이 다수의 연구 결과로 검증되었으나 여전히 많은 학습자들이 비주얼 프로그래밍 언어에서 텍스트 언어로 전환하는데 어려움을 겪고 있다. 앨리스에서 제공하는 넷빈즈 플러그인을 이용한다면 자바(Java) 프로그램으로의 전환이 수월할 것으로 여겨진다. 이를 활용한 교수법 및 VR 기기를 연동한 교수안을 기획해 보고 싶은 바람이 있다.

[참고문헌]

- 1) Wing, J. M. "Computational thinking. Communications of the Association for Computing Machinery", 19(3), 33-35, 2006.
- 2) 김광열, 송정범, 이태욱, "디지털 스토리텔링 기반 프로그래밍 교육이 학습자의 동기 및 학업 성취도에 미치는 영향", 한국컴퓨터정보학회논문지 14(1), 47-55, 2009.
- 3) 김광열, 박정호, 이태욱, "스토리텔링을 활용한 초등학교 프로그래밍 교육의 방법 탐구", 한국컴퓨터교육학회 하계 학술발표논문지 3(2), 1999.
- 4) 문외식, "초등학생들이 프로그래밍 학습 시 발생하는 오류유형 분석", 한국 컴퓨터정보학회, 11(2), 2006.
- 5) Lehnert, W.G., Loiselle, C.L., "Plot Units and Narrative Summarization", Cognitive Science 4, 293-331, 1981.
- 6) 서성은, "메타버스 개발동향과 발전전망 연구", 한국HCI학회 학술대회, 1450-1457, 2008.
- 7) "Where Millennials end and Generation Z begins | Pew Research Center", <https://www.pewresearch.org/fact-tank/2019/01/17/where-millennials-end-and-generation-z-begins/>.
- 8) "Roblox - Wikipedia", <https://en.wikipedia.org/wiki/Roblox>.
- 9) "Unity 시작하기 - Unity Store", <https://store.unity.com/kr/lego-microgame>.
- 10) "Official Minecraft Wiki - The Ultimate Resource for Minecraft", https://minecraft.fandom.com/wiki/Minecraft_Wiki.
- 11) B. Jost, M. Ketterl, R. Budde, T. Leimbach, "Graphical Programming Environments for Educational Robots: Open Roberta - Yet Another One?", IEEE International Symposium on Multimedia, 381-386, 2014.
- 12) "Scratch - 소개", <http://scratch.mit.edu/about>.
- 13) "엔트리 소개 : 엔트리", <https://playentry.org/about>.
- 14) "Alice - Tell Stories. Build Games. Learn to Program.", <http://alice.org>.
- 15) "Kodu Game Lab | KoduGameLab", <https://www.kodugamelab.com/>.
- 16) "MIT App Inventor | Explore MIT App Inventor", <https://appinventor.mit.edu/>.
- 17) Rahman, F., "From App Inventor to Java: Introducing Object-oriented Programming to Middle School Students Through Experiential Learning", ASEE Annual Conference & Exposition, 2018.
- 18) Selmer Bringsjord, David Ferrucci, "Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine", Psychology Press, 1999.
- 19) "AI Writer - The best AI Text Generator, promised.", <https://ai-writer.com/>.
- 20) Roger McDermott, Gordon Eccleston & Garry Brindley, "More than a good story - can you really teach programming through storytelling?", Innovation in Teaching and Learning in Information and Computer Sciences, 7(1), 34-43, 2008.

- 21) Joel Adams, "Alice3 in Action Computing Through Animation", 2e, Cengage Learning, 2015.
- 22) 김곤, 양재균, 배재학, "개연구칙의 활용 : Plot Units의 발견", 한국정보과학회 학술발표논문집 31(1B), 652-654, 2004.

[Abstract]

Implementation of Alice Library for Software Education based on Storytelling

Lee, Se Young

Department of Smart IT Convergence Engineering

University of Ulsan

The coding ability requires creativity based on 'Computational Thinking' to prepare for the era of the 4th Industrial Revolution. Coding is a complex problem solving process, which is similar to 'Writing'. This is because the 'Writing' ability that proceeds in the form of solving problems is similar to an algorithm that is also the process of complex thinking activities. This is the reason why it is necessary that programming education should be based on storytelling. In this thesis, Alice is introduced among numerous educational programming languages. Alice is an innovative block-based programming environment that makes it easy to create animations, build interactive narratives in 3D. Learners would be able to develop computational thinking and creativity by making animations and games using Alice. It was presented that a method of analyzing a story based on Plot Units to implement interactive storytelling. Sample sentences in Primitive Plot Units were used to decompose the sentences into Affect States and Casual Links to show the interaction between the characters. Scenarios were presented along with an animation to express the character's state of mind in Alice based on

this. However, it takes time to create procedures for characters to be moved naturally. In order to reduce the time and trial and error in Alice, a group of procedures based on Plot Units were written to be provided for Alice learners. The set of procedures written based on the examples of Primitive Plot Units was saved as an Alice class file and was provided into a library. The learner can easily express the interaction between the characters and their state of mind by reusing the procedures in this library. The learners will be able to obtain the intended result easily.

Keywords : Alice3, Plot Units, Storytelling, Software Education, Block Based Programming Language, Metaverse, Learn to Code, Alice procedures