

Viscuit을 활용한 코딩 교육에 관한 연구

황서진[†] · 채지선^{††} · 김지현^{†††} · 박성빈^{††††}

요 약

본 연구는 일본에서 제작된 교육용 프로그래밍 언어 Viscuit의 장점들을 소개하고, 코딩 입문 수단으로서의 가능성을 제안한다. 특히 일본어에 대한 지식 없이도 본 연구에서 제작된 사용 안내서와 심화학습서만을 이용하여 학습자들을 대상으로 실험한 결과 쉽게 코딩에 대해 친숙해질 수 있음을 알 수 있었다. Viscuit은 잘 알려진 Scratch나 LightBot과 같은 언어에 비해 인지도는 낮은 편이지만 아주 간단하게 그림들을 조작하여 쉽게 코딩 연습을 할 수 있도록 해 주어서 학습자들에게 유용한 코딩 입문 수단이 될 수 있으리라 기대된다.

주제어 : 코딩 교육, Viscuit, 교육용 프로그래밍 언어

Research on Coding Education using Viscuit

Seojin Hwang[†] · Jiseon Chae^{††} · Jihyun Kim^{†††} · Seongbin Park^{††††}

ABSTRACT

In this research, we introduce advantages of Viscuit which is an educational programming language developed in Japan and propose a possibility of using Viscuit to introduce coding. Specifically, we conducted experiments against students using both the manual and the focused study guide and the results indicated that they became familiar with coding easily without the knowledge of Japanese. Even though Viscuit is less well-known than languages such as Scratch and LightBot, it is very easy to practice coding using Viscuit because all that users need to do is manipulating pictures simply and it is expected that Viscuit will serve as a useful vehicle by which students can learn how to code.

Keywords : Coding Education, Viscuit, Educational Programming Language

† 정 회 원: 고려대학교 일어일문학과 학부과정

†† 정 회 원: 고려대학교 언어학과 학부과정

††† 정 회 원: 고려대학교 행정학과 학부과정

†††† 종신회원: 고려대학교 컴퓨터학과 교수 (교신저자)

논문접수: 2018년 2월 20일, 심사완료: 2018년 3월 7일, 게재확정: 2018년 4월 11일

1. 서론

2018년부터 소프트웨어 교육이 공교육과정에 포함된다는 ‘2015 개정 교육과정’ 발표 이후, 학부모들의 코딩 교육에 대한 관심이 높아지고 있다[1]. 본 연구에서는 교육용 프로그래밍 언어(EPL)의 하나로 일본에서 개발된 Viscuit(www.viscuit.com)을 이용하여 프로그래밍 경험이 적거나 아예 없는 학습자들에게 Viscuit을 활용한 코딩 교육의 가능성을 제안한다. Viscuit은 웹에서 실행되기 때문에 컴퓨터 사용 환경에 구애받지 않으며, 사용방법이 매우 단순하다는 장점을 가진 뛰어난 EPL이지만 일본어만을 지원하기에[2] 인지도가 낮은 편이다. 그러나 Viscuit 사용화면에 일본어 텍스트는 없고 코딩에 사용되는 버튼이 모두 그림으로 표현되어 있을 뿐 아니라 일본어 능력은 사용자가 이전에 만든 작품을 찾아보거나 사용법을 익히는 데에만 필요하다.

실제로 본 연구에서 프로그래밍 경험이 아예 없거나 적은 사용자들을 대상으로 실험해 본 결과 모두 Viscuit을 이용하여 프로그래밍을 배우는 것에 대해 긍정적이었고 일본어 지식이 문제가 된다고 생각하지 않았다. 특히 Viscuit은 한 번 조작법을 익힌 뒤로는 일본어를 모르는 사람도 충분히 이용할 수 있고 웹에서 무료로 사용가능해, 가정에서 부모와 자녀가 함께 코딩을 학습하기에도 적합하다. 이는 코딩 경험이 전무해 코딩에 막연한 거부감을 느끼는 학부모의 인식을 전환할 수 있으며, 교과과정 개정으로 인해 급성장하고 있는 코딩 사교육의 열기를 낮추는 데에도 효과적일 것이다.

현재 일본에서는 학부모와 자녀가 함께 Viscuit을 배우는 수업이 진행되고 있으며, 2020년부터 초등학교의 코딩 교육이 의무화된다는 발표 이후 이에 대한 관심이 더 높아지는 추세이다. 또한 Viscuit은 일본의 사설 교육기관과 지방자치단체가 제공하는 IT 교실, 합동회사 디지털포켓(デジタルポケット), 개발사인 NTT에게 위탁받아 Viscuit 공식 사이트 운영 중)에서 주최하는 워크숍 등에서 교구로 사용되고 있다. 대상은 유아부터 초등학교, 중학생, 성인까지 다양하며, 각 대상에 따라 다른 커리큘럼이 적용된다[3].

본 논문은 다음과 같이 구성되었다. 2장은

Viscuit 언어의 특징을 기술하고 3장은 교구로서 Viscuit이 사용되고 있는 실제 사례들을 소개한다. 4장은 Scratch [9] 및 LightBot [14] 과 Viscuit 언어를 비교하고 5장은 본 연구의 일환으로 제작한 Viscuit 사용 안내서 및 Viscuit 심화 학습서를 기반으로 Viscuit 언어를 프로그래밍 언어 관점에서 분석한다. 6장은 본 연구에서 제작한 매뉴얼을 이용하여 사용자들의 의견에 근거한 Viscuit 언어 사용의 용이성 등에 대한 평가 결과를 서술하고 7장에서는 본 연구의 결론을 제시한다.

2. Viscuit 언어의 특징

Viscuit의 목표는 코딩을 통해 ‘노는’ 것이고 Viscuit을 통해 완성한 프로그램은 ‘작품’이라고 불린다. 코딩 그 자체가 목적이기에, 배열이나 변수 같은 것을 다루지 않지만 프로그램과 컴퓨터의 기본원리를 익히기에는 충분하다[4]. Viscuit 코딩에 사용되는 핵심적인 버튼은 ‘안경’ 버튼이다. 안경처럼 생긴 부품의 좌우에 학습자가 직접 그린 그림을 넣어, 왼쪽 그림에서 오른쪽 그림으로 움직임을 변화시킬 수 있다. 물고기가 헤엄치는 Viscuit 프로그램을 통해 자세한 코딩 방법을 설명하면[5], 우선 각 부분의 명칭은 [그림 1]과 같다.

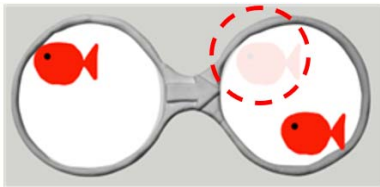


[그림 1] Viscuit 인터페이스

사용자는 먼저 움직일 물고기를 그리고, 회전 여부, 이동 방향 등의 구체적인 움직임을 결정해야 한다. 일반적인 프로그래밍에 비유하면 물고기 그림이 데이터, 구체적인 움직임이 알고리즘에 해당한다. Viscuit에서는 앞으로 움직일 그림, 즉 데이터를 ‘부품’이라고 부르며 생성한 부품은 부품창고에 나와 있다. 또한 알고리즘은 안경 버튼을 통해 구현한다.

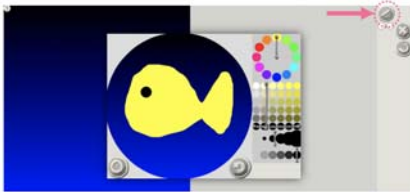
이용할 부품을 전부 그린 뒤에는 스테이지에 부품을 드래그 하고, 안경버튼을 꺼낸 후, 좌우에 움직이고 싶은 대로 부품을 넣는다. 안경 버튼을 확

대하면 [그림 2]와 같다. 왼쪽 원에 넣은 부품이 기준이 되어, 오른쪽 원 속 부품의 위치와 형태가 되도록 움직인다. 일반화하면, Viscuit 에서의 코딩은 그림을 그려 데이터를 작성한 뒤 안경으로 알 고리즘을 구현하는 것이라고 할 수 있다.



[그림 2] 안경 화면

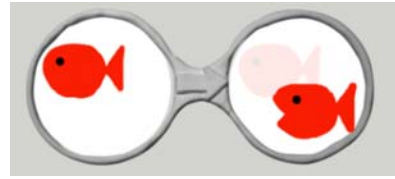
안경 사용법에 따라 부품이 바뀌는 모습은 스테이지에서 그때그때 확인할 수 있다. 즉 스테이지는 프로그램 초기화면 구성과 코드 실행 결과 미리보기라는 두 가지 기능을 담당한다. 스테이지 배경색은 여러 옵션 중 선택할 수 있으며, 배경색을 고른 후에는 연필 버튼을 클릭해 부품을 그린다. [그림 3]의 오른쪽 위에 표시한 부분이 연필 버튼이다.



[그림 3] 부품 생성 화면

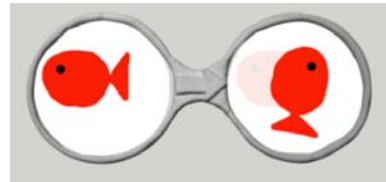
안경을 가지고 만들 수 있는 움직임의 종류는 크게 이동·변화·회전으로 나뉜다. 이동은 [그림 2]처럼 안경 왼쪽 그림이 오른쪽 그림 위치로 나아가는 것을 말하는데, [그림 2]의 경우, 오른쪽 그림이 왼쪽 그림 기준 대각선 아래쪽에 있으므로, 물고기가 대각선 아래 방향으로 이동할 것이다. 표시한 부분처럼 왼쪽 그림의 위치가 희미하게 나타나 있어 사용자가 그를 보고 물고기의 움직임을 예측할 수 있다.

변화는 [그림 4]와 같이 안경 좌우에 다른 그림을 넣어, 왼쪽에 넣은 그림이 오른쪽 그림으로 바뀌도록 만드는 것이다. 이 때 [그림 4]는 좌우 그림의 형상뿐만 아니라 위치까지 다르므로, 그림의 모습이 바뀌며 이동하게 된다. 즉, 물고기가 입을 벌린 채 대각선 아래 방향으로 움직일 것이다. 이렇게 Viscuit은 하나의 안경으로 동시에 두 가지 이상의 움직임을 만들 수 있다.



[그림 4] 이동 및 변화

회전은 [그림 5]처럼 사용자가 그린 그림을 시계 혹은 반시계 방향으로 회전시키는 것이다. [그림 5]는 안경 좌우에 동일한 부품을 넣은 뒤 시계방향으로 회전시킨 형태이기 때문에, 물고기가 시계 방향으로 회전할 것이다.



[그림 5] 회전

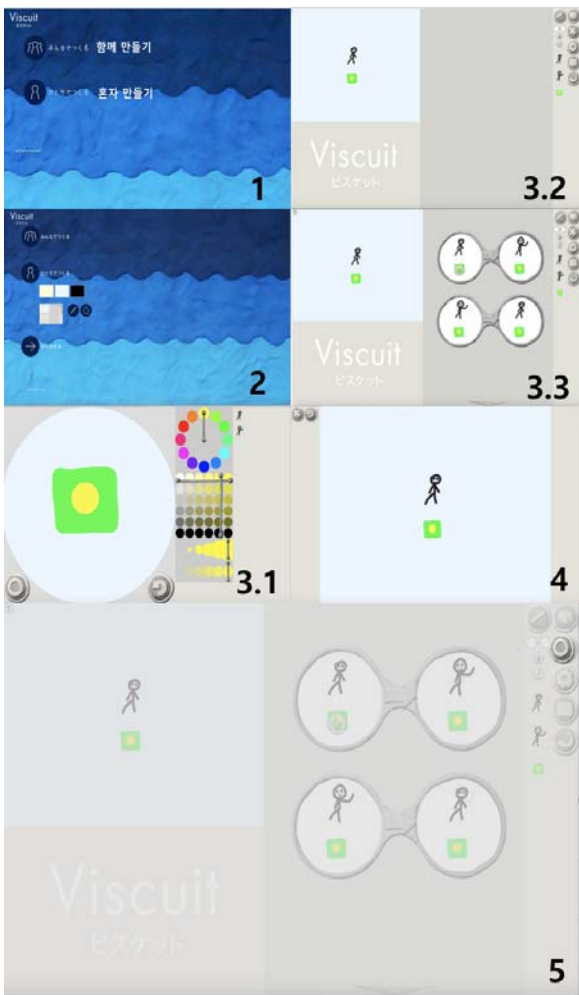
Viscuit 코딩 과정을 좀 더 구체적으로 설명하기 위해 초인종을 눌렀을 때 사람이 인사하는 프로그램을 작성하는 과정을 보면 다음 다섯 단계들을 거친다.

1. 초기화면에서 모드 선택: ‘혼자 만들기’ 모드
2. 놀이터(작품의 배경) 선택: 원하는 배경색
3. 작품 제작
 - 3.1 필요한 부품 그리기: 사람, 인사하는 사람, 초인종’
 - 3.2 프로그램 초기화면 구성: 스테이지에 ‘사람’과 ‘초인종’ 배치
 - 3.3 안경조작
 - 3.3.1 초인종을 클릭 시 ‘사람’이 인사하도록 만드는 안경:
 - 3.3.1.1 안경을 ‘안경 두는 곳’에 드래그
 - 3.3.1.2 안경 왼쪽에 ‘사람’, ‘초인종’, ‘클릭 버튼’ 넣기
 - 3.3.1.3 안경 오른쪽에 ‘인사하는 사람’과 ‘초인종’ 부품 넣기
 - 3.3.2 인사 후 원래 모습으로 돌아오도록 만드는 안경:
 - 3.3.2.1 새 안경을 ‘안경 두는 곳’에 드래그
 - 3.3.2.2 안경 왼쪽에 3.3.1.3의 부품 넣기
 - 3.3.2.3 안경 오른쪽에 3.3.1.2의 부품 넣기
4. 작품 확인: ‘전체화면’ 버튼을 클릭해 작품의 정

상 작동 여부 확인

5. 작품 저장

‘클릭 버튼’은 왼쪽 그림의 특정 지점을 클릭 했을 때 오른쪽 그림으로 바뀌도록 도와주는 장치이다. ‘클릭 버튼’을 사용하지 않을 경우 왼쪽 그림은 자동으로 오른쪽 그림으로 바뀐다. ‘전체화면’은 작품 미리보기 기능을 제공하며, 3.3.2는 작품을 반복적으로 실행할 수 있게 해준다. [그림 6]은 전체 과정을 보여준다.



[그림 6] Viscuit 코딩 과정

이렇게 표현하고자 하는 것을 이미지로 나타낼 수 있는 경우라면, Viscuit으로도 얼마든지 복잡한 알고리즘을 구현할 수 있다. 숫자를 써야 할 경우에는 그를 특정 도형의 개수로 표현하면 된다. 예를 들어 8은 동그라미 여덟 개로 나타낼 수 있다.

3. 교구로서의 Viscuit 사용 실례

Viscuit은 각종 사설 교육업체에서 코딩 워크샵 교구로 사용되고 있으며, 일본 총무성에서 2016년부터 진행한 ‘소년층에 대한 프로그래밍 교육 보급 추진(若年層に對するプログラミング教育の普及推進)’ 2차 사업에 교구로 선정되기도 하였다.

‘소년층에 대한 프로그래밍 교육 보급 추진’ 사업은 “지리 및 신체적 조건과 상관없이 모든 학생이 질 높은 프로그래밍 교육을 받을 수 있도록 현직 인재를 멘토로 육성하는 동시에, 교재와 강의 노하우 등을 인터넷 상에서 공유·활용하고 프로그래밍 교육을 실시하는 모델”의 시험 운영 사업이다[6]. 2016년부터 시작하였으며, 2차 사업은 지난해 7월부터 11월까지 일부 비영리단체와 영리단체, 지역단체를 포함한 24개교에서 진행되었다. Viscuit은 초등학교 8곳과 외부 단체 3곳, 총 11곳에서 사용되었다.

구체적인 사용 실례를 들면 2016년 12월 27일에 열린 ‘동계 비스킷 워크숍 어린이 망년회(冬のビスケットWS コドモ忘年會)’의 대상과 그에 따른 수업내용은 <표 1>과 같다[7].

<표 1> 동계 비스킷 워크숍 어린이 망년회

대상	프로그램 내용
4살 이상	‘선물’을 주제로 한 단순 애니메이션 제작
초등학교 1학년 이상	일본 전통 카드 놀이 ‘가루타(かるた)’ 제작
초등학교 3학년 이상 · 보호자	「부모와 자녀가 함께 배우는 컴퓨터과학입문(親子で学ぶコンピュータサイエンス入門)」 수업으로 Viscuit 작품을 제작함과 동시에 컴퓨터 과학 지식 학습

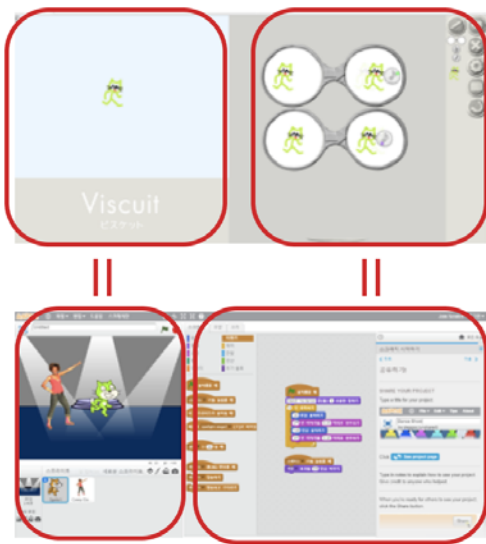
또한 Viscuit 교수법에 대해서 2014년 7월 13일, ‘제 1회 비스킷 유저 컨퍼런스’가 개최되었고, 이듬해인 2015년부터 개발자인 하라다 야스노리(原田康德) 박사가 직접 Viscuit 퍼실리테이션 강좌를 열고 있다. 강좌의 목적은 초등학생을 대상으로 Viscuit 수업이 가능한 강사를 양성하는 것으로, 2017년 기준 200명 이상이 수료했다[3].

일본에서 프로그래밍은 중학교부터 배우기 시작하며 유치원과 소학교(한국의 초등학교에 해당)에서는 컴퓨터 다루는 법과 같은 기초 지식만을 배

운다[8]. 그러므로 Viscuit 주 학습대상인 초등학생과 미취학아동의 경우 일본과 한국의 교육환경이 유사하다고 볼 수 있으며, 일본에서의 Viscuit 교육 방식을 한국에도 적용할 수 있을 것이다.

4. Scratch 및 LightBot과 비교

Viscuit은 상대적으로 간단한 기능들을 제공하지만 대표적 EPL인 Scratch로 작성할 수 있는 프로그램들과 비슷한 기능의 프로그램도 작성 가능하다. 예를 들어 [그림 7]은 Viscuit과 Scratch로 고양이 고양이가 앞뒤로 움직이며, 움직일 때 마다 소리가 나는 프로그램을 만든 것이다[9]. 두 언어 모두 화면 왼편에서는 프로그램 실행 모습을 미리 확인할 수 있고, 오른편에서는 코드를 작성한다.



[그림 7] 고양이 움직임에 따라 소리가 나는 프로그램 (위: Viscuit, 아래: Scratch)

Scratch는 프로그램에 문자를 삽입할 수 있고, 화면의 물체를 수치를 입력하여 이동시키기 때문에 언어 지식과 좌표, 음수 개념을 알아야 한다. 오직 그림으로만 표현되어 선행 지식이 불필요한 Viscuit과 대비되는 점이다. Scratch는 ‘OO만큼 움직이기’같은 기본으로 제공되는 명령문의 공란에 원하는 수치나 문자만 입력하여 명령문이 쓰인 블록을 연결하면, 물체가 명령문대로 움직인다. 따라서 언어와 수학에 익숙한 사람이라면 Scratch를 이용해 프로그램을 작성하는 쪽이 효율적일 수 있다. 코드가 눈에 보이기 때문에 작동 방식을 확인

하기 용이하고, 다양한 이미지 소스를 제공해주어 사용자가 그림을 직접 그리지 않고도 프로그램을 완성할 수 있다는 점 또한 장점이다.

반면 Viscuit은 사전에 정의된 형태나 움직임이 없기 때문에, 호기심과 흥미를 유발하는 동시에 학습자가 창의력을 최대한으로 발휘할 수 있다. 또한 Viscuit 작품의 움직임은 오직 안경을 통해서만 만들어지기 때문에, 안경 툴의 사용법만 숙지하면 문법 없이도 코딩에 직관적인 접근이 가능하며, Viscuit에만 존재하는 ‘함께 만들기’ 모드는 협동 학습을 유도할 수 있다. 함께 만들기 모드를 사용하면 개개인이 만든 여러 작품을 한 화면에서 볼 수 있기 때문에 다수의 학생이 조화로운 하나의 작품을 만들 수 있어 단체 교육에 특히 효과적이다.

LightBot은 전진·회전·점프·점등 버튼을 조합하여 로봇을 움직임으로써, 최종적인 목표지점인 파란 타일에 불을 켜는 퍼즐 게임이다. LightBot의 로봇 조작 버튼은 명령문, 그를 나열하여 만든 로봇의 움직임은 프로그램에 해당한다. 반면 버튼을 놓을 수 있는 공간에 제한이 있는데 그 공간의 길이는 대부분 버튼이 가장 효율적으로 배열되었을 때와 같다. 그러나 특정 프로그램을 만드는 데 사용되는 코드는 얼마든지 다르게 표현될 수 있고, 길이가 짧다고 꼭 좋은 코드라고 할 수도 없다. 학생들에게 다양한 방식으로 동일한 목적을 이룰 수 있다는 점을 가르치기 위해서는 길이에 제한을 두지 않는 것이 바람직하다[10].

프로시저와 조건문, 오버로딩 개념을 학습할 수 있는 것은 LightBot만의 장점이지만 변화된 내용을 매번 확인해가며 코딩할 수 있게 해주는 기능은, Viscuit에서는 실행 버튼을 누르지 않아도 디폴트로 제공되는 것에 비하면 불편할 수 있다. Viscuit은 게임은 아니지만 학습자가 만들고 싶은 프로그램을 고안하여 스스로 코딩하는 과정에서 흥미를 느낄 수 있고, LightBot 학습 목표의 일부인 시퀀스와 조건문 개념을 학습할 수도 있다.

프로그래밍을 어렵게 만드는 요소는 7가지로 분류할 수 있다 [11]. 첫째는 프로그래밍이 단일 스킬이 아니라 Syntax와 시맨틱, 구조, 스타일 각각에 대한 이해를 요구하는 멀티-스킬 활동이라는 점이며, 둘째는 그런 작업을 하기 위해서는 먼저 알고리즘을 설계해야 한다는 점이다. 또 초보자에

게 어렵게 느껴질 수 있는 프로그래밍 언어와 프로그래밍이라는 새로운 과목을 개설해야 하는 교육적인 부담감, 코딩이 흥미와 재미를 유발하기 힘들다는 점도 그에 포함된다. 여섯째는 프로그래밍이 어렵다는 평판과 이미지를 가지고 있어 접근 자체를 꺼리게 된다는 점이며, 마지막으로 대학의 커리큘럼에 따라 코딩을 배우기 때문에 학생 개인의 학습 속도에 맞추기 힘들다는 사실이 프로그래밍을 어렵게 만든다.

Viscuit은 안경 조작이라는 고유의 문법을 통해 학습자 스스로 구조와 스타일을 결정하여 프로그램을 만든다. 즉, 쉬운 난이도로 코딩에 요구되는 여러 스킬을 동시에 훈련함으로써, 알고리즘을 포함해 멀티 스킬에 대한 이해가 필요하다는 코딩의 첫 번째와 두 번째 문제점을 해결할 수 있다. 원래 교육용으로 제작된 언어이기에 초보자에게 매우 적합하며, 익숙해지기까지 걸리는 시간이 짧고 웹에서 무료로 제공되기 때문에 프로그래밍 과목 개설 시에도 비용을 최소화할 수 있다. 또한 단순한 인터페이스는 프로그래밍에 대한 부정적인 인식 전환에 도움이 될 것이고, 본 연구과정에서 제작한 매뉴얼을 이용한다면 일본어를 모르는 사람도 언제 어디서든 Viscuit을 학습할 수 있다.

Viscuit 코딩은 학습자가 만들고자 하는 프로그램을 정한 뒤 필요한 부품을 직접 그리는 것에서 시작한다. 사용하는 안경 개수의 제한도 100개로 거의 없다고 할 수 있고 사용자는 부품 디자인을 정하는 것부터 알고리즘 설계까지 전부 스스로 하며 창의력과 상상력을 키울 수 있다. 반면, LightBot의 프로그램은 타일에 불을 켜는 것 하나 뿐이다. 타일의 패턴은 퍼즐에 따라 다르지만 궁극적인 목적은 모든 파란 타일에 불을 밝히는 것이다.

동기 부여의 경우 LightBot은 모든 예제를 풀면 답을 다 알고 있는 상태가 되어 반복 학습 시 흥미가 급격히 떨어지고, 위 단계로 갈수록 난이도가 가파르게 상승해 포기하는 사람이 발생할 수 있다. 반면 Viscuit은 달성해야 할 과제가 유한하지 않고 스스로 만들어내기 때문에 동기 부여에 더 효과적이며, 프로그래밍 언어가 그림으로 대체되었을 뿐 코딩의 전 과정을 다루기 때문에 프로그래밍의 이미지를 상상하는 데에도 더 유용할 것이다.

5. 활용 및 적용

본 연구에서는 Viscuit 활용 방안의 일환으로 기본적인 조작법을 다루는 ‘Viscuit 사용안내서’와 좀 더 어려운 예제를 포함하는 ‘Viscuit 심화학습서’를 제작하였으며, 각각의 구성은 <표 2>, <표 3>와 같다.

<표 2> 사용안내서

목차	세부내용
I. Viscuit이란?	Viscuit 프로그램 소개
II. 사용 준비	Viscuit 작품 제작 페이지 접속 방법 및 모드(‘함께 만들기’, ‘혼자 만들기’) 선택 방법 설명
III. 따라 해보기	1. 함께 만들기: 예제를 통해 ‘함께 만들기’ 모드의 해업치는 물고기 기본 조작법 설명[11]
	2. 혼자 만들기: 예제를 통해 ‘혼자 만들기’ 모드의 초인종을 누르면? 기본 조작법 설명
Q. 코딩이란 무엇일까?	Viscuit을 통한 프로그래밍 설명

코딩은 추상화·알고리즘·코딩의 세 단계로 이루어지며 문제를 정확하게 이해하여 원하는 바를 달성하기 위해, 문제의 핵심 내용만을 가려내는 것이 추상화이다. 이를 통해 문제를 요약한 뒤, 구체적으로 어떤 알고리즘을 사용하여 문제를 해결할지 결정한다. 그런 뒤에야 비로소 툴을 선택해 코딩작업에 들어갈 수 있다[12].

<표 4>에 각 단계를 세부적인 요구사항으로 나누고, 그 때 필요한 사고능력과 환경 등의 내용을 추가하여 코딩의 필수 조건을 나타내었다. ‘O’, ‘△’, ‘X’는 각각 해당되는 능력이 코딩에 ‘꼭 필요하다’, ‘필수는 아니나 유용하다’, ‘필요하지 않다’를 나타내는 기호이다. <표 4>의 ‘Viscuit 사용 효과’ 열은 ‘Viscuit을 사용함으로써 얻을 수 있는 능력’을 나타내기 때문에, ‘O’, ‘X’가 각각 Viscuit 사용을 통해 이 능력을 ‘충분히 얻을 수 있다’와 ‘얻기 힘들다’를 의미한다.

다음은 본 연구과정에서 제작한 심화학습서의 3번째 심화학예제 ‘슈팅 게임’에 주어진 시나리오이다.

“서기 2100년. 인류는 우주로 진출하여 다양한 행성에서 살게 되었고, 당신은 물자를 운송하는 우주선의 함장을 맡고 있다. 언제나처럼 지구에서 재

배한 작물을 다른 행성으로 운반하던 중, 해적이 물자를 뺏기 위해 당신의 우주선을 가로막았다! 해적을 피해 무사히 탈출하고 물자를 운송하라!”

학습자는 이 시나리오를 보고 게임을 제작한다. 게임 구조를 정하기 위해 시작점과 목표 상태를 설정할 것이고, 게임의 장애물로서 해적이 필요하다고 분석할 것이다. 그 외의 서기 2100년이라는 시간적 배경 혹은 함장이라는 지위는 고려할 필요가 없으므로 배제한다. 일련의 과정을 통해 학습자는 ‘문제 이해’와 ‘문제 분석’ 능력을 향상시킬 수 있다. 이는 Viscuit 자체보다는 문제를 얼마나 많이 접해보는가와 관련 있기 때문에, 발전을 위해서는 Viscuit으로 다양한 과제를 해결해보는 연습이 필요하다.

다음으로 ‘알고리즘 이해’는 기존에 존재하는 알고리즘을 이해할 수 있는 능력을 말한다. Viscuit은 작품을 만들기 전에 ‘놀이터’라고 불리는 작품의 밑바탕을 고를 수 있는데, 그를 기준으로 각 놀이터 별로 제작된 모든 작품과 작품 코드를 볼 수 있는 사이트를 제공한다. 이곳에서 타인이 만든 Viscuit 코드를 보며 알고리즘을 분석하는 활동을 통해 ‘알고리즘 이해’ 능력을 습득할 수 있다.

직접 작품을 만들며 ‘알고리즘 설계’ 실력 또한 키울 수 있다. Viscuit에서는 안경 조작 방법과 순서의 조합이 알고리즘에 해당한다.

Viscuit은 쉬운 사용법을 통해 코딩에 대한 친근감을 효과적으로 높일 수 있다. 반면 낮은 난이도로 인해 ‘프로그래밍 언어 이해’력 향상에는 적합하지 않다. 프로그래밍 언어의 이해가 변수, 반복 등의 개념과 문법 학습은 물론, 프로그래밍 언어의 개발 환경과 특성 이해를 포괄하는 능력을 일컫기 때문이다[13].

Viscuit은 고유 문법과 프로그래밍 언어로서의 개발 환경·특성이 존재하고, 이를 분석할 수도 있으나, 구조가 단순하여 변수나 반복문 등의 실제 코딩에서 중요한 개념을 다룰 수는 없고 ‘프로그래밍 언어 이해’에는 한계가 있다. ‘코드 작성’과 ‘코드 이해’는 알고리즘의 경우와 마찬가지로 Viscuit으로 직접 작품을 만들고 타인 작품의 코드를 분석함으로써 능력을 기를 수 있다.

코드 오류를 수정하는 ‘디버깅’의 경우, 발생하는 오류가 다양한 만큼 디버깅 종류 또한 다양하다. Viscuit 코딩에서도 단순 문법 실수, 알고리즘설계 오류 등 여러 오류가 발생하기 때문에 실제 코딩

<표 3> 심화학습서

목차	세부내용	
I. 심화 예제	1. 함께 만들기: 어떤 꽃이 필요요?	여러 색의 꽃 중 하나가 랜덤으로 피게 만드는 애니메이션 제작
	2. 혼자 만들기: Viscuit Run	인기 모바일 게임 ‘쿠키런’을 참고하여 달리는 캐릭터가 장애물을 넘는 게임 제작
	3. 혼자 만들기: 슈팅 게임	우주선이 해적선을 피하는 슈팅 게임 제작
II. 제작 가이드: 나만의 작품 만들기	Q1. 무엇을 움직이고 싶나요? Q2. 어떻게 움직이고 싶은가요? Q3. 2번에 쓴 움직임을 작게 나눠봅시다. Q4. 1번의 대상을 그려봅시다. Q5. 3번에서 나눈 각각의 움직임을 안경으로 만들어봅시다. Q6. 프로그램을 전체화면에서 실행한 뒤, 2번에서 생각한 대로 움직이는지 확인해 봅시다. 예상과 다르다면, 프로그램의 어디에서 차이가 생기는지 확인하고 그 부분을 고쳐서 다시 실행시켜봅시다. Q7. 생각한대로 움직일 때까지 6번을 반복합니다.	
III. 생각해 볼 거리	Q1. Viscuit으로 만든 것은 ‘프로그램’이고, 그를 만드는 과정을 ‘코딩’이라고 합니다. 여러분이 만든 작품을 바탕으로 생각했을 때, ‘프로그램’이란 무엇이고, ‘코딩’은 무엇일까요? Q2. 평소에 컴퓨터 또는 스마트폰에서 사용하고 있는 프로그램에는 어떤 것이 있습니까? 그런 프로그램의 기능을 Viscuit에서의 움직임과 같다고 볼 수 있을까요? 만약 그렇다면, 그 움직임을 Viscuit에서 한 것처럼 작은 단위로 나눌 수 있습니까? 나눈다면 어떤 움직임이 될까요? Q3. ‘프로그램’은 다양한 도구를 통해 만들 수 있습니다. Viscuit은 그 도구 중 하나로, 많은 기능이 있지만, 프로그램에 음악이나 글자 등을 넣을 수 없는 한계도 있습니다. 비스킷으로 만들 수 없는 프로그램에는 어떤 것이 있을까요? Q4. ‘함께 만들기’와 ‘혼자 만들기’ 모드 선택 화면에서 다른 사람이 만든 작품을 감상해봅시다. 그 사람이 만든 작품이 어떤 움직임을 하는지 파악할 수 있습니까? 좋은 코드란 어떤 코드일까요? Q5. Viscuit으로 만들고 싶은 프로그램이 있습니까? 여러분이 혼자 그 프로그램을 만드려면 시간이 얼마나 걸릴까요? Q6. 앞서 답한 ‘코딩’과 ‘프로그램’의 의미를 생각했을 때, Viscuit으로는 할 수 없지만 다른 프로그램으로는 가능한 일에는 어떤 것이 있을까요? 만약 여러분이 Viscuit 외의 다른 프로그래밍 언어를 사용할 수 있다면, 어떤 프로그램을 만들고 싶나요?	

과 유사점이 있다. 이 때, 단순 문법 실수로 인한 경 사용법이 잘못된 경우를 들 수 있고, 알고리즘 실수로 사전에 설계한 안경을 실제로 만들었을 때 움직임이 예측과 다른 경우를 들 수 있다. Viscuit 초심자용 모드인 ‘함께 만들기’ 모드에서는, 단순 문법 실수에 한해 하얀 안경이 붉게 변해 오류를 범했음을 알려주는 기능을 제공한다[5].

<표 4> Viscuit 사용의 기대 효과

		코딩 필요조건	Viscuit 사용 효과
추상화	문제이해	O	O
	문제분석	O	O
알고리즘	알고리즘 이해	O	O
	알고리즘 설계	O	O
코딩	프로그래밍 언어 이해	O	X
	코드 작성	O	O
	코드 이해	O	O
	코딩을 통한 알고리즘 구현	O	O
	디버깅	O	O
	코딩에 대한 친밀감	△	O
사고력	논리적 사고	O	O
	추론력	O	O
	창의력	O	O
	호기심	△	O
기타	협동 능력	O	O
	동기 부여	△	O

‘호기심’은 ‘코딩에 대한 친밀감’과 비슷한 맥락으로, Viscuit에서는 안경 속에 부품을 어떻게 넣는가에 따라 하나의 부품으로도 무궁무진한 움직임을 만들 수 있기 때문에, 호기심을 자극하는 데에 유용하다. ‘기타’의 ‘동기부여’ 또한 같은 이유이다. 쉬운 사용법과 작품 제작 과정의 재미는 학습자에게 코딩에 대한 흥미를 느끼게 함으로써 그를 자발적인 코딩 학습으로 유도할 것이다.

‘협동 능력’은 코드를 공유·비교·분석함으로써 프로그램을 수정·보완하여 효율적인 프로그램을 완성하는 능력으로, 작품 감상 사이트를 통해 훈련할 수 있다. 또한 ‘함께 만들기’ 모드를 통해 개별적인 작품이 모여 하나의 큰 작품을 형성하도록 계획할 수도 있다.

6. 평가 및 분석

5장에서 예측한 효과가 실재하는지 알아보기 위해, 자체 제작한 매뉴얼을 ‘프로그래밍 경험이 전혀 없거나 약간 접해본¹⁾’ A, B, C, D, E에게 사용하도록 한 뒤 평가받았다. A, B, C는 대학교 인문계열 학생들이며, D는 고등학생, F는 중학생이다. Viscuit에 대한 사용자 의견은 <표 5>와 같다.

<표 5> Viscuit에 대한 사용자 의견

대상	평가내용	
A	흥미	재미있었다. 실제로 프로그래밍 언어를 공부해보고 싶어졌다.
	적합 연령	10대 후반에게 적합할 것 같다.
B	흥미	재밌었다.
	적합 연령	‘그림판’같은 기본 프로그램과 비슷하기에, 유아부터 초등학교 저학년까지 적합할 것 같다. 성인은 간단해 보이는 외관으로 인해 Viscuit을 무시할 것 같다.
C	흥미	생각보다 훨씬 재미있었다. 여러 움직임을 만들거나, 내가 원하는 캐릭터를 만들 수 있다는 점도 흥미로웠다. 실제 프로그래밍 언어를 공부해보고 싶다는 생각이 들었다.
	적합 연령	사용안내서의 ‘헤엄치는 물고기’ 같은 예제는 초등학교 저학년도 얼마든지 가능하다고 생각한다.
D	흥미	작품을 만드는 것 자체는 재밌었지만 매뉴얼의 설명을 읽는 시간이 오래 걸렸기 때문에, 프로그램을 만드는 일이 힘들다고 생각되어 공부해보고 싶은 생각이 들지 않았다.
	적합 연령	연령에 상관없이 컴퓨터만 사용할 수 있다면 전부 가능하다고 생각한다. 성인이라면 코딩을 처음 배우는 초보자에게 유용할 것 같다.
F	흥미	프로그래밍을 전혀 몰랐는데 어떤 것인지 조금 이해하게 되었다. (Viscuit이) 신기했지만 실제 프로그래밍 언어가 무엇인지 잘 몰라서 (실제 프로그래밍 언어) 공부를 해야 할지는 잘 모르겠다.
	적합 연령	초등학교 고학년부터는 다 사용해도 괜찮을 것 같다.

평가 내용을 살펴보면, Viscuit은 흥미롭고 코딩 학습으로의 유도에 도움이 되며, 예제 난이도에 따라 유아에게도 사용가능하다. 또한 미흡한 점을 수정하여 새로운 매뉴얼을 만들고 그를 초등학생 E에게 사용하도록 하여 <표 6>과 같은 코멘트를 받을 수 있었다. E의 평가를 보면, 안경 속에 부품을 넣는 위치와 관련하여 매뉴얼의 예제를 쉽게 해결하기 어려운 것으로 보인다. 언뜻 단점으로 보

1) E의 경우 EPL인 ‘엔트리’ (<https://playentry.org>)의 사용 경험이 있는 초등학생이었다.

일 수 있으나 이는 사실 Viscuit의 장점 중 하나이다. 실제 프로그래밍 언어로 코딩을 하는 과정에서도 콜론(:)이나 세미콜론(;), 온점(.) 등을 빠뜨리는 사소한 실수로 인해 프로그램이 작동하지 않는 경우가 많다. 부품의 위치에 엄격한 Viscuit의 특성은 이러한 문제의식을 반영한 것으로 보인다. 부품의 위치로 코딩에 요구되는 정확성을 표현한 것이다. 그럼만 사용하는 특성상 프로그래밍 언어 학습과 관련이 없어 보일 수도 있지만, 부품의 위치를 주의 깊게 조정해야하기 때문에 Viscuit 작품을 만드는 과정에서도 실제 프로그래밍 언어의 문법에서 정확성이 요구된다는 사실을 깨달을 수 있다.

반복문의 부재 또한 비슷한 맥락에서 이해 가능하다. 반복문이 없을 경우 동일한 코드를 입력해야 하므로 상당히 번거로울 것이다. 실제로 반복문이 없는 Viscuit에서는, 반복이 필요할 경우 일일이 새 안경을 만들어야 하며, 이는 지루하게 느껴질 수 있다. 그러나 동시에 학습자에게 반복문의 필요성을 알려준다. 반복문이 없을 때의 불편함을 느끼게 하여 반복문의 유용함과 필요성을 인식시키는 것이다.

<표 6> 수정 매뉴얼과 Viscuit에 대한 사용자 의견

대상	구분	평가내용
E	매뉴얼에 대한 평가	'Viscuit Run'과 '슈팅게임'에서 안경 속 부품의 위치를 정확하게 맞추지 않으면 부품이 인식되지 않는 점이 어려웠다.
	Viscuit에 대한 평가	재미있으며 이후 취미로 사용할 의향이 있으나, 안경 속 부품의 위치를 맞추는 것이 전부 이므로 실제 프로그래밍 언어 학습에는 큰 도움이 되지 않을 것 같다. (전반적으로) 쉬우나 부품 위치 선정이 까다로웠다. 반복문을 포함한 명령어가 없다는 점이 (Viscuit의) 특징이다.

7. 결론

Viscuit은 잘 알려진 EPL들과 달리 인지도가 상대적으로 낮은 편인데 그 이유 중 하나는 일본어로만 서비스가 제공되기 때문이다. 그러나, 실제 Viscuit을 이용하여 코딩을 하는데 일본어 지식이 많이 필요하지 않을 뿐 아니라 언어 자체는 그림들을 이용하여 쉽고 재미있게 코딩 연습을 할 수 있도록 여러 장점들을 가지고 있다. 본 연구에서는 Viscuit의 장점들을 소개하고 코딩 입문 수단으로 Viscuit의 사용 가능성을 제안하였다. 이를 위해 본

연구의 일환으로 제작된 사용자 설명서와 심화학습서를 바탕으로 학습자들에게 실험을 한 결과를 분석하였고 학습자들에게 코딩에 대한 친밀감을 느끼게 해 줄 뿐 아니라 흥미를 유발시켜준다는 것을 알 수 있었다. 향후 더 많은 학습자들이 Viscuit을 이용하여 코딩에 입문할 수 있으리라 기대된다.

참고 문헌

[1] 교육부. (2015). 2015 개정 교육과정 안내. <https://www.moe.go.kr>

[2] 한국컴퓨터교육학회. (2014). 국내외 SW교육 운영현황 및 요구사항 조사. **정보통신산업 진흥원**, 1-178.

[3] Digitalpocket, <http://www.digitalpocket.org>

[4] 原田康徳. (2012). 生徒児童によるプログラミング-ビスケットの挑戦. **情報処理**, 1, 64.

[5] 合同會社デジタルポケット, 原田康徳, 渡辺勇士, 井上愉可里. (2017). **ビスケットであそぼう 園児・小學生からはじめるプログラミング**. 東京: 翔泳社, 8-19.

[6] 總務省. (2017). 若年層に対するプログラミング教育の普及推進 (平成28年度~). http://www.soumu.go.jp/main_sosiki/joho_tsu sin/kyouiku_joho-ka/jakunensou.html

[7] Peatix. (2015). 冬のビスケットワークショップ コドモ忘年會. <http://vis20151227.peatix.com/>

[8] 조규복. (2015). 일본의 프로그래밍 교육과 정보교과의 현황과 과제. **한국교육개발원**, 8, 1-39.

[9] Scratch <https://scratch.mit.edu>

[10] Caroline Ormesher, Dannynay. (2016). App Review: Lightbot. **Mathematics Teaching**, 251, 46.

[11] T. Jenkins. (2002). On The Difficulty of Learning to Program. **3rd Annual LTSN-ICS Conference, Loughborough University**, 53-58.

[12] 국가교육과정정보센터. (2015). 실과(기술가정)_정보과 교육과정(제2015-74호). <http://ncic.go.kr>

[13] Viscuit, <http://www.viscuit.com>

[14] Lightbot, <http://lightbot.com>



황 서 진

2015~현재 고려대학교
일어일문학과 학부과정
관심분야: 코딩교육, 컴퓨터교육
E-Mail: prol@naver.com



채 지 선

2014~현재 고려대학교
언어학과 학부과정
관심분야: 시맨틱웹, 전산언어학
E-Mail: wltjscow@naver.com



김 지 현

2016~현재 고려대학교
행정학과 학부과정
관심분야: 시맨틱웹, 하이퍼텍스트
E-Mail: jodyzzang77@naver.com



박 성 빈

2003~현재 고려대학교
컴퓨터학과 교수
관심분야: 하이퍼텍스트,
컴퓨터교육, 시맨틱웹
E-Mail: hyperspace@korea.ac.kr