

# 코딩 교육용 게임에 나타난 수행적 글쓰기 - 〈The Foos〉를 중심으로

윤현정  
인하대학교 한국학연구소  
hjun@inha.ac.kr

Performative Writing of Coding Game 〈The Foos〉

Hyun-Jung Yun  
Center for Korean Studies, Inha University

## 요 약

본 연구는 SW중심 사회를 맞아 주목받고 있는 코딩과 코딩 교육에 대한 새로운 관점을 제시하고자 하였다. 이에 코딩 교육용 게임 〈The Foos〉를 중심으로 코딩과 이를 통한 게임 플레이 안에서 나타나는 수행적 글쓰기로서의 양상을 분석하였다. 연구 결과 글쓰기 과정으로서 코딩은 코드 블록을 활용한 연쇄적 확장과 이의 순환이라는 작동 방식을 가지고 있었다. 또한 유닛 오퍼레이션으로서 의미를 생성하고 있었는데 이는 사용자의 창발적 플레이를 지시하는 동인이 되었다. 다음으로 수행적 글쓰기로서 코딩은 말하기/쓰기가 중첩되며 사용자의 플레이 전 과정에 체현되고 있었다. 이어 실패와 교정의 부재는 역설적으로 사용자의 문제 탐험 공간을 확장시키며 사용자의 자율적인 문제 해결 능력을 강화시키고 있었다. 본 연구는 융합교육 환경에서 코딩과 코딩 교육의 다양한 가능성을 모색하는 시도로서 의의가 있다.

## ABSTRACT

This study tries to present a new perspective for the coding and coding education. Therefore, this study analyzes coding game 〈The Foos〉 as performative writing. First, coding as writing process has the mechanism of serial expansion and circulation by utilizing basic code blocks. And coding as writing process produces a meaning as unit operation that generate an emergent game play. Second, coding as performative writing displays the speaking and writing at the same time and is embodied in game play. Also, absence of failure and correction makes possible users to explore the problem area and to strengthen the problem-solving ability.

**Keywords** : Coding, Performative Writing, Programing, S/W Education, The Foos  
(코딩, 수행적 글쓰기, 프로그래밍, 소프트웨어 교육, 더 푸즈)

Received: Jan. 24, 2016    Revised: Feb. 15, 2016  
Accepted: Feb. 18, 2016  
Corresponding Author: Hyun-Jung Yun(Inha University)  
E-mail: hjun@inha.ac.kr

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1598-4540 / eISSN: 2287-8211

## 1. 서 론

### 1.1 연구배경 및 연구목적

‘SW(Software)중심사회’로의 변화는 컴퓨터 프로그래밍 전공자 뿐 아니라 초·중·고등학생을 대상으로 한 코딩(coding) 교육의 확산으로 이어지고 있다. 아이디어의 구현과 문제점 해결에 SW가 중심적으로 활용되는 ‘SW중심사회’에서[1] SW교육은 전통적인 수학, 과학 등의 교과에서 다루지 않았던 새로운 유형의 사고방식인 ‘컴퓨팅적 사고(Computational Thinking, 이하 CT)’를 함양한다고 여겨진다[2]. CT는 읽기, 쓰기, 셈하기와 같이 21세기를 살아가는 이들에게 필요한 기본 소양으로 코딩 교육을 통해 프로그래밍 원리와 컴퓨터 정보처리의 방식으로 문제를 해결하는 과정을 배우으로써 함양할 수 있다[3].

영국은 2014년 SW교육을 정규 필수 과목으로 지정했으며, 같은 해 미국 역시 컴퓨터 프로그래밍을 정규 교과 과정으로 편성한 바 있다. 한국의 경우 2018년부터 초·중등 교과 과정에 SW의무 교육이 도입될 예정이다[4].

이러한 흐름아래 SW교육에 대한 다양한 학문적 논의가 이루어져 왔다. 이들은 대부분 컴퓨터 교육의 관점에서 실시된 연구들로 국내·외의 관련 교육 과정 및 그 운영 현황 등을 분석하거나[5,6,7,8], 운영된 교육 과정의 효과를 측정하는 연구[9,10], 실제 교육에 활용 가능한 콘텐츠 및 교재의 개발 방안과 사례 등을 소개하는 연구로 이루어진다[11,12,13]. 이들 연구들은 새롭게 실시된 교육 과정으로서 코딩 교육에 대한 초기적 연구로 의의를 지닌다. 때문에 그 연구의 목적과 내용이 교육 현장에서의 교과 운영, 효과 측정, 도구 개발이라는 실질적인 측면에 편중되어 있는 것 또한 사실이다. 이에 본 연구는 SW교육이 지니는 보다 근본적인 차별점과 특성을 살펴볼 필요가 있음에 주목한다.

SW교육은 프로그래밍 교육, 코딩 교육과 동일한 의미로 받아들여지고 있는데 이는 1980년대 Papert LOGO 프로그래밍 시대로 거슬러 올라가

는 역사를 지닌다[14]. 관련하여 박정호는 초기 SW교육이 교육현장에 정착하지 못한 이유 중 하나로 프로그래밍을 다양한 창작 표현의 도구가 아닌 프로그래밍 그 자체로 인식하였기 때문이라고 주장한다[15]. 이는 아이들의 흥미와는 거리가 있는 활동들이었다는 것이다.

때문에 오늘날 SW교육에 활용되고 있는 대표적인 도구 중 하나인 <스크래치(Scratch)>는 다양한 학습자 주도의 창작활동을 지원하고 있다. <스크래치>는 MIT 미디어랩에서 2007년 개발한 블록형 프로그래밍 언어로 작은 명령 단위인 블록 조각을 서로 조합해 프로그래밍할 수 있으며, 구글의 <블럭키(Blockly)>, 일본의 <프로그래밍(プログラミン)>, 한국의 <엔트리(Entry)> 등이 이 유형에 속한다[16].

특히 미국을 중심으로 시작된 ‘아워즈 오브 코드(Hours of Code, <https://hourofcode.com/kr>)’ 캠페인은 2016년 1월 기준, 약 1억 9천 5백만 명의 학생이 경험한 대규모 코딩 교육 운동으로 전 세계 40개 이상의 언어로 서비스되고 있다. 이는 4세에서 104세까지 누구나 참여 가능한 1시간짜리 코드 교육 과정으로 코딩의 기초 개념과 원리를 쉽게 이해할 수 있도록 하는 것을 목표로 한다.

‘아워즈 오브 코드’ 역시 블록형 프로그래밍 언어를 통한 학습 프로그램을 지원하고 있는데 2016년 1월 현재, <스타워즈>, <마인크래프트>, <겨울왕국>, <앵그리 버드>의 스토리와 캐릭터를 활용한 테마가 존재한다. 플레이어는 각 테마별로 지시된 스토리를 따라가며 캐릭터를 도와 단계별로 코딩의 과정을 학습하게 된다. 예를 들어 <마인크래프트> 테마의 경우 14개의 단계로 이루어진 코딩을 학습하게 된다. 1단계에서는 이동하기와 같은 기본적인 코딩을 학습하며 마지막 단계에서는 실제 게임 <마인크래프트>처럼 채굴을 통해 다양한 아이템을 얻고 블록을 쌓는 등의 복잡한 코딩을 학습하여 자신만의 새로운 게임을 창조하고 실행해 볼 수 있도록 한다.

이와 같이 오늘날 주로 활용되고 있는 블록형

프로그래밍 언어를 통한 코딩 교육은 앞서 지적된 프로그래밍 그 자체로서의 SW교육의 한계를 벗어나 보다 창의적이고 다양한 활동을 통해 흥미를 제공하고 있음을 알 수 있다. 이들은 실제 컴퓨터 언어로 코드를 짜는 것을 학습하기보다는 CT적 사고의 함양을 주요 목적으로 하고 있는 것이다.

본 연구는 바로 이러한 지점에 주목한다. 코딩 교육용 도구들은 실제 프로그래밍과는 상이한 특유의 방식으로 설계되고 디자인되었으며, 학습자의 플레이를 통해 차별적인 성과를 획득하고 있다.

이에 이러한 특유의 방식을 만들어 내는 근본적인 메커니즘에 대한 고찰이 필요하며 이는 향후 코딩 교육 및 코딩 교육용 도구의 방향성을 설정하는데 중요한 시사점을 제공할 수 있으리라 사료된다. 또한 이를 통해 코딩의 다양한 가능성들을 탐색함으로써 SW중심사회에서 코딩 교육이 지니는 함의 역시 모색해 볼 수 있으리라 기대한다.

## 1.2 연구대상 및 연구방향

본 연구는 코드 스팩(CodeSpark)의 <The Foos>(http://thefoos.com/)를 구체적인 연구대상으로 선정하고자 한다. <The Foos>는 ‘아워즈 오브 코드’에서 과생한 별도 튜토리얼로 Computer Science Education Week에서 개발한 무료 코딩 교육 게임이다. 2014년 10월 베타테스트를 시작했으며 PC, 안드로이드, iOS, 킨들 파이어 등 다양한 기기와 플랫폼을 지원한다.

4세 이상, 주로 6-8세 아동을 대상으로 하며 2015년 Parents’ Choice 금상, Apple Editor’s Choice 및 최고 교육 앱 선정, LEGO’s Re-Imagine Learning Challenge 최종 후보 등에 오르며 그 완성도와 성과를 인정받았다.

<The Foos>는 앞서 살펴본 코딩 교육 도구들과 동일한 블록형 프로그래밍 언어를 사용하지만 이들과 비교해 하나의 독립된 게임 형태를 지닌다는 차이점을 보인다. ‘아워즈 오브 코드’에서 직접 제공하는 학습 프로그램들의 경우, 게임성을 내포하고는 있지만 실질적으로는 게임을 직접 플레이하

며 즐긴다기보다 코드의 결과를 재현해 본다는 정도의 의미를 지니는데 반해 <The Foos>는 독립된 게임의 구조를 지닌다. 때문에 코딩 교육용 도구의 차별적인 메커니즘을 살피고자 하는 본 연구의 연구대상으로 적합하다 할 것이다. 구체적으로 본 연구의 분석에 활용된 <The Foos>는 2015년 12월 1일 업데이트된 1.9.28 iOS버전이다.

본 연구는 <The Foos>의 게임 플레이를 통해 진행되는 코드 짜기 과정을 일종의 ‘수행적 글쓰기(performative writing)’로 보고 이를 구체화하고자 한다. 델라 폴록(Della Pollock)은 수행적 글쓰기는 인류학, 페미니스트 비평, 퍼포먼스 관련 글쓰기의 영역 등에서 길고 다양한 역사를 가지고 있으며 아방가르드, 스타일리쉬, 트렌디와 같은 수식어를 동반하지만 이를 통해 설명될 수 있는 어떤 고정된 장르나 형태가 아니라고 이야기한다[17]. 즉 수행적 글쓰기라는 개념은 여전히 문제적인데, 본 연구에서는 캐서린 N. 헤일즈(Katherine N. Hayles)의 관점을 받아들여 코드를 수행적 언어로 보고 코드를 짜나가는 코딩을 일종의 수행적 글쓰기 과정으로 전체하고자 한다.

이에 먼저 글쓰기 과정으로서 코딩의 의미를 살펴보고 <The Foos>에 구현된 그 작동 방식과 의미 생성의 양상을 분석하고자 한다. 이는 코딩 교육용 게임으로서 <The Foos>가 지니는 차별적 메커니즘을 밝히는 과정이 될 것이다. 다음으로 이러한 메커니즘이 유도하는 수행적 글쓰기로서의 특성을 살펴본다. 이를 통해 코딩과 코딩 교육의 다양한 가능성을 가늠해 볼 것이다.

## 2. 글쓰기 과정으로서의 코드 짜기

코드는 정보를 전달하는 언어이며, 코딩은 정보가 매체 간에 옮겨질 때 이를 그 매체가 이해할 수 있는 새로운 패턴으로 번역하는 과정이다[18]. 공학적 관점에서 코드는 추상적 논리의 표현이자 실행을 위한 도구로 분석 텍스트가 될 수 없다고 여겨

져 왔지만[19] 위와 같이 코딩은 그 개념과 역할 자체에 이미 글쓰기로서의 가능성을 내포하고 있다.

도날드 크누스(Donald Knuth)는 ‘코딩은 다양한 의미를 표현하기 위해 주의 깊게 선택된 단어들을 작가가 그러하듯이 인간이 잘 이해할 수 있도록 개념화하고, 정형화되거나 정형화되지 않은 방법들을 섞어 사용한다[20].’고 말한 바 있다. 그는 코드 짜기를 단어의 선택, 이를 통한 의미의 표현, 다양한 사용 방식 등의 일련의 과정으로 나타냄으로써 글쓰기 과정으로서 코드 짜기의 가능성을 구체화한다.

코드를 미학적 자산으로 고려하는 코드 비평 연구 역시 코드를 사용한 창조적인 글쓰기 작업을 ‘코드 워크(code work)’라 칭하고 코드는 배열되어 보이는 작품의 레시피일 뿐 아니라 그 자체로 전시 가능한 작품이 되기도 함을 주장한다[21].

한편 매체 연구가 빌렘 플루서(Vilém Flusser)의 논의는 글쓰기 과정으로서 코드 짜기를 살펴보는 데 좀 더 구체적인 시사점을 던진다. 플루서에 따르면 디지털 시대의 문예창작가는 더 이상 ‘독창적인 창작자’가 아닌 ‘언어 배열자’로서 자신을 인식해야 한다[22]. 이들은 더 이상 전통적 이야기꾼처럼 경험을 나누는 자가 아닌 정보제공자이기 때문이다. 플루서가 직접적으로 코딩을 언급하고 있지는 않지만 디지털 시대에 나타날 새로운 글쓰기의 방식은 코딩의 그것과 닮아 있음을 알 수 있다.

이러한 논의들을 종합해 볼 때 코딩은 전통적 글쓰기와 유사한, 언어를 통한 창조적인 행위인 동시에 새로운 유형의 글쓰기로서 의미를 지닌다 할 것이다.

이러한 전제하에 본 장에서는 <The Fools>에 나타난 글쓰기 과정으로서의 코드 짜기를 구체적으로 분석한다. 이때 코드 교육용 게임인 <The Fools>에서는 사용자의 게임 플레이 행위 전체가 코드 짜기를 지시하게 된다. 이에 사용자의 게임 플레이를 하나의 글쓰기 과정으로 보고 그 기저에 놓인 작동 방식과 의미 생성의 과정을 차례로 살펴본다.

## 2.1 작동 방식 : 연쇄적 확장과 순환

<The Fools>는 하나의 완결된 게임으로 서비스되고 있기 때문에 다양한 요소들을 포함하고 있다. 그 중 가장 핵심적인 요소는 코드 블럭을 활용한 코딩을 통해 게임을 플레이하는 44개의 스테이지라 할 것이다.

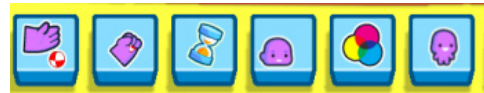
이때 게임 플레이는 ‘푸즈’ 캐릭터들을 조종하며 진행된다[Fig. 1]. 이들은 컴퓨터 칩 속에 살고 있는 언제나 즐겁고 행복한 가상 존재들이다.



[Fig. 1] Game Characters of <The Fools>

<The Fools>에는 언어적 부가설명이 존재하지 않는다. 메뉴를 비롯한 인터페이스는 글자를 모르는 아이들도 직관적으로 이해할 수 있는 아이콘으로 표시되며 배경 스토리 역시 간단한 컷씬으로 대사 없이 전달된다. 때문에 코딩을 활용하는 44개의 스테이지에서 캐릭터들은 더욱 강력한 힘을 가지게 된다. 왜냐하면 캐릭터의 움직임을 통해서만 모든 것을 확인하고 이해할 수 있기 때문이다.

44개의 스테이지는 총 5개의 묶음으로 다시 구성된다. 이때 각 스테이지의 주인공 푸즈들은 상이하며 이들은 각각의 목표를 부여받는다. 사용자는 코드 블럭을 활용한 코딩을 통해 이 목표를 완수하여야 한다[Fig. 2].



[Fig. 2] Code Block Examples

이때 코드 블럭을 배열한 후 실행하여 캐릭터를 적절하게 움직이도록 하는 것이 게임 플레이의 핵심이다. [Table 1]은 각 스테이지 단계별 요소들을 분석한 표이다.

[Table 1] Coding Stages of <The Fools>

1-8 stage	space	european village
	character	cap
	story	finding the donut
	item	none
9-16 stage	coding	forward/backward move, jump
	space	construction site
	character	builder
	story	finding the tools
	item	box, dynamite, tire
17-24 stage	coding	loop statement
	space	japanese house
	character	ninja
	story	finding friend's kites
	item	key, bigger, threw, smash
25-34 stage	coding	loop statement
	space	the cosmos
	character	astronaut
	story	finding my puppy
	item	dynamite
35-44 stage	coding	if statement
	space	restaurant
	character	chef
	story	cooking the recipe
	item	shake, tomato spaghetti, cheese toast, egg toast
	coding	while switch case

예를 들어 ‘1-8 스테이지’의 주인공 캐릭터인 경찰관 푸즈는 악당 푸즈가 훔쳐간 도넛을 찾아야 한다. 서양풍 마을을 배경으로 플레이가 진행되며 튜토리얼 단계에 걸맞게 앞/뒤로 걸거나 뛰는 방법의 코드 블록을 적절히 사용, 배열하여 도로에 놓인 도넛을 회수해야 한다.

스테이지가 진행되면서 이는 점점 심화된다. ‘25-34 스테이지’의 주인공은 우주 비행사로 우주 미아가 된 애완견을 구하라는 목표를 부여받는다. 해당 스테이지에서는 이전에 학습한 코드 뿐 아니라 ‘if statement’ 코드 블록을 새롭게 배우게 된다. 실제 게임 플레이 상에서는 해당 코드 블록을 사용해 ‘이동 경로에 방해물이 있을시 밀거나/폭파시키거나/피하는 방법’을 선택할 수 있다.

흩어진 공구를 찾는 건설가 푸즈가 등장하는 ‘9-16 스테이지’와 친구의 놓쳐버린 연을 찾아주는

닌자 푸즈가 등장하는 ‘17-24 스테이지’, 손님이 원하는 메뉴에 맞게 요리를 대접해야하는 셰프 푸즈가 등장하는 ‘35-44 스테이지’에 이르기까지 사용자는 게임 플레이 과정에 설정된 목표를 이루기 위해 동일한 형태의 코딩을 반복하여야 한다.

구체적으로 ‘1-8 스테이지’는 ‘기본 이동과 관련된 코드’, ‘9-16 스테이지’는 ‘loop statement’, ‘25-34 스테이지’는 ‘if statement’, ‘35-44 스테이지’는 ‘while switch case’를 학습하게 되는데 사실상 각 스테이지는 앞 선 스테이지의 코딩을 조금씩 반복하며 확장해 나가는 형태를 보인다.

예를 들어 기본 이동과 관련된 코딩을 학습하는 ‘1-8 스테이지’에서는 ‘앞으로 한번 이동->앞으로 두 번 이동->앞으로 세 번 이동->앞으로 한 번 이동 후 앞으로 한 번 뛰기->앞으로 한 번 이동 후 앞으로 세 번 뛰기->뒤로 한 번 이동->뒤로 한 번 이동 후 뒤로 한 번 뛰기...’와 같은 방식으로 코딩을 반복하게 된다. 스테이지의 진행에 따라 미리 학습한 코딩을 반복적으로 연습하며 동시에 새로운 코딩을 덧붙여 나가게 되는 것이다. 이때 코드 짜기는 연쇄적으로 덧붙여지며 확장되고 전체 스테이지에 걸쳐 반복되며 순환된다.

결국 사용자는 연쇄적 확장과 순환의 과정을 통해 각 스테이지가 목표한 코딩을 자연스럽게 습득하게 되며 이를 다음 단계의 스테이지에도 계속해서 반복, 적용하면서 또 다른 새로운 코딩을 학습하게 된다. 이러한 작동 방식을 통해 사용자는 게임의 플레이 과정에서 한 번 학습한 코딩을 점진적으로 강화시켜 나가게 되는 것이다.

## 2.2 의미 생성 : 유닛 오퍼레이션

게임 연구자인 이안 보고스트(Ian Bogost)는 결정론인 시스템을 넘어서는 불연속적이고 단절된 움직임이 특징인 의미 생성 방식으로 유닛 오퍼레이션(unit operation)을 제시한다. 그에 따르면 유닛 오퍼레이션은 시스템 오퍼레이션의 반대 개념으로 독립적 자질을 가진 불연속적이고 우발적인 성격의 유닛이 목적이 있는 어떤 행위를 하는 전 과정을

의미한다[23].

시스템 오퍼레이션이 정적, 선형적, 지속성, 예측가능함 등의 특징을 지닌 것과 달리 유닛 오퍼레이션은 역동적, 불연속적, 상호적, 즉시적이며 복잡함과 다양성을 추구한다. 때문에 시스템 오퍼레이션이 발견적이라면 유닛 오퍼레이션은 탐험적이다. 유닛 오퍼레이션은 이러한 역동적 내부 관계에서 의미를 도출하는 보다 창조적이고 동적인 의미 생성의 구조이다. 보거스트에 의하면 유닛 오퍼레이션은 복잡계 및 문학과 예술 비평을 위한 도구로도 활용될 수 있다.

<The Foes>에서는 두 가지 차원에서 유닛 오퍼레이션적 의미 생성이 일어나는데 첫 번째 차원은 코드 블록을 활용해 게임을 플레이해 나가는 것이다.

앞 절에서 살펴본 바와 같이 <The Foes>는 단순한 움직임의 지정하는 최소한의 코딩 블록들을 배열해 게임을 플레이한다. 이때 이동, 뛰기와 같은 단순 블록들은 플레이가 진행됨에 따라 상호관계 안에서 역동적으로 변화하는데 ‘상하좌우’라는 방향 유닛이 이동 블록에 추가되는 등의 변화가 그 예이다.

이동을 지시하는 코딩 블록 뿐 아니라 수식과 관련된 코딩 블록 역시 마찬가지다. 반복문, 조건문을 나타내는 코딩 블록들은 각각의 유닛으로 다양하게 조합할 수 있으며 자유롭게 배열가능하다.

<The Foes>는 목표를 제시하긴 하지만 그 방식에 있어 다양성을 인정한다. 이는 동일한 하나의 실행에도 수 백 가지 다른 방식의 코딩이 가능한 실제 코딩과 유사하다. 그러나 실제 코딩의 경우 잘못된 코드 짜기를 한다면 오류로 인해 실행이 불가할 수도 있지만 <The Foes>는 사용자가 어떠한 방식의 배열, 즉 글쓰기를 하더라도 이를 게임 플레이화한다. 이런 맥락에서 <The Foes>는 창조적이고 동적인 의미 생성이라는 유닛 오퍼레이션의 기초를 잘 반영하고 있다 할 것이다.

예를 들어 ‘35-44 스테이지’에서 토마토 스파게티를 만들기 위해서는 메뉴를 묻는 구문 안에 ‘스

파게티 삶기’, ‘토마토 소스 붓기’, ‘서빙’에 해당하는 코드 블록을 순서대로 배열하여야 한다. 하지만 사용자는 ‘스파게티 삶기’만을 연속 배열할 수도 있으며 그 결과는 ‘타 버린 스파게티’로 산출된다. 이때 손님은 메뉴를 거부하고 떠나며 음식은 애완동물의 차지가 된다. 정해지지 않은 유닛들의 조합이 예상치 못한 경우의 수를 산출하며 새로운 의미를 생성해 내는 것이다.

이런 유닛화된 방식을 통해 생성되는 탐험과 발견, 그리고 창조성의 즐거움은 <The Foes>의 메인 화면 메뉴와 인터페이스에도 고스란히 반영되어 있다. 이는 단순한 외부적 형태가 아니며 <The Foes>의 유닛 오퍼레이션적인 인터페이스 구조는 사용자의 게임 플레이 매커니즘과 긴밀한 관계를 맺으며 특수한 플레이 형태를 유도한다.

먼저 <The Foes>의 메뉴에는 ‘위에서 아래로’, 혹은 ‘아래에서 위로’와 같은 시스템 구조가 존재하지 않는다. <The Foes>는 상호 관련적으로 새로운 메뉴와 맵이 열리며 사용자는 자유롭게 이들 간을 오가며 플레이할 수 있다.

또한 사용자는 <The Foes>를 플레이함에 있어 직접적인 코드 교육과 관련된 44개의 스테이지 메뉴를 반드시 마스터할 필요가 없다. [Table 2]에서 정리한 바와 같이 <The Foes>의 메인 메뉴는 총 5개인데, 44개 스테이지 메뉴 외에도 다른 메뉴를 통한 다양한 활동이 가능하기 때문이다.

<The Foes>를 처음 실행할 때는 코딩을 학습하는 44개의 스테이지 메뉴만이 존재한다. 그러나 ‘1-8 스테이지’를 완료할 경우 미니게임 메뉴가 추가된다. 미니게임 메뉴는 세 가지 목적을 지니고 있는데 첫째, 게임 그 자체를 즐기기 위한 미니게임 본연의 역할이 그것이다. 하지만 더 중요한 목적은 사용자가 코드 블록을 활용하여 자신만의 게임을 제작하는 방법을 미리 경험해보도록 하는 것이다. 때문에 미니게임 메뉴에서 미니게임을 플레이하기 위해서는 반드시 사전에 코드 블록을 활용해 게임을 플레이할 수 있는 환경을 만들어야 한다. 마지막 목적은 다른 메뉴를 위한 열쇠로서의

미니게임이다. 미니게임을 진행함에 따라 상점 메뉴, 나만의 게임 제작 메뉴, 친구 방문 메뉴가 생성되기 때문이다.

[Table 2] Main Menu of <The Fools>

coding stage	space	european village
	purpose	learning coding
	etc.	44 stage
mini game	space	hollywood studio
	purpose	mini game, pre-learning about using code block for my game
	etc.	to connect with store, my game craft, social
store	space	treasure house
	purpose	purchasing items
	etc.	no cash item
my game craft	space	workbench
	purpose	crafting own game with code block
	etc.	save & share
social	space	world map
	purpose	to play other player's game, rating
	etc.	modify others game & save

이러한 메뉴 생성의 과정은 단계적이라기보다 유닛적이다. 사실상 기본적인 단계만 통과하면 각각의 메뉴가 생성되며 각각 메뉴들은 별도로 존재한다. 이들은 각자의 의미를 생성하지만 또 한편으로 사용자에게 의한 새로운 게임의 제작이라는 <The Fools>의 궁극적인 목표를 위해 긴밀하게 상호협력하고 있기 때문이다. 예를 들어 상점에서는 게임 제작에 활용 가능한 아이템을 구매할 수 있으며 게임 제작 메뉴에서는 코드 블럭을 활용해 나만의 게임을 제작하고 이를 공유할 수 있다. 친구 방문 메뉴에서는 다른 사용자가 제작한 게임을 플레이하고 평가하거나 수정할 수 있다.

이와 같이 <The Fools>에서는 코딩 교육의 궁극적인 목표인 CT적 사고를 통한 새로운 창의성의 발현이 유닛 오퍼레이션적 의미 생성의 과정을 통해 완수되고 있는 것이다.

### 3. 수행적 글쓰기로서의 특성

#### 3.1 말하기/쓰기 중첩을 통한 체현

플루서는 “모든 글쓰기는 인간이 자신의 사유들을 정확한 궤도 속에 안내하기 위해 문자를 질서 짓고 배열하는 것”이라고 이야기한 바 있다. 때문에 그는 이런 의미에서 기계들이 더 글쓰기에 적합하다고 주장하기도 한다[24]. 플루서의 이러한 논의는 글쓰기로서의 코드 짜기에 적용되어 기계에 의해 쓰이는 새로운 글쓰기의 특성을 포착하게 하는 단서가 된다.

실상 코드는 글자와 달라서 그 자체로, 그리고 이것을 단순히 배열하는 것만으로 의미가 생성되지 않는다. 코드는 그것이 실행될 때 비로소 의미를 가진다. 즉 코드가 가진 수행성은 텍스트로서 코드의 차별적 가능성을 밝히는 데 중요한 지점이 된다. 코드는 수행 가능한 언어로서 의미를 가지며 [25] 코딩과 전통적 글쓰기의 차이점은 코드가 매우 특별한, 수행하기 위한 언어라는 데 있기 때문이다[26].

이와 관련하여 헤일즈는 코드가 지닌 새로운 수행성은 말하기와 쓰기 중 어느 것에도 종속되지 않는다고 주장한다. 코드의 수행성은 쓰기가 말하기로부터 비롯된다고 보는 소쉬르의 입장이나 그 반대의 테리다의 입장과는 다르며 어떤 다른 수평적 영향 속에서 인간과 컴퓨터 간의 역동적 상호 관계를 만들어 낸다는 것이다[27].

전통적 작가의 글쓰기에서 작가는 쓰기를 반복할 뿐 말하기를 경험할 수 없다. 때문에 글을 쓰는 순간 작가에 의해 의미가 생성되는 동시에 작가에게도 자신이 표현하고자 했던 의미가 현화해 된다.

그러나 <The Fools>와 같이 코딩 블럭을 활용한 코딩 게임에서 사용자는 쓰기 행위 그 자체로 의미를 생성시키지 못한다. 코드 블럭이 단어이고 블럭을 배열하는 것이 쓰기라면, 의미는 쓰기를 마친 후 캐릭터를 플레이하는 순간에 현화해 된다. 플레이 전까지 어떠한 의미도 지니지 못했던 코드 블럭은 바로 그 순간 특별한 의미를 생성하게 된

다. 즉 게임 플레이가 일종의 말하기 과정이라면 코드 블록을 배열하는 쓰기의 완료와 게임을 플레이하는 말하기가 중첩되어 현행화 되면서 그 의미를 생성시키는 것이다.

이는 전통적 작가의 글쓰기와 차별화되는 지점이다. 수행적 글쓰기로서 코딩은 쓰기와 말하기를 동시에 포함하기 때문이다. 인간-기계의 글쓰기인 코딩은 그 자체로 행위하며 특별한 어떤 것을 만들어 내는 것이다[28]. 전통적 작가의 글쓰기에서 쓰기는 독자에 의해 발생하는 말하기와 분명히 분리되어 있지만 코딩 교육 게임에서 주체는 쓰기를 경험하는 작가이자 동시에 이를 말하는 사용자로 역할 한다. 결국 사용자는 게임 플레이 전 과정에서 말하기와 쓰기의 중첩을 코드 짜기를 통해 체현하게 되는 것이다.

이와 같은 말하기/쓰기의 중첩이라는 수행적 글쓰기의 특성은 의미의 수정과 삭제의 차원에서 전통적 글쓰기와 변별되는 양상을 만들어낸다.

전통적 작가의 글쓰기에서 일차적인 의미의 수정과 삭제가 작가의 몫이었다면 수행적 글쓰기로서의 코딩에서 일차적 의미의 수정과 삭제는 사용자의 몫이 된다. 코딩에서는 플레이의 순간에만 의미가 존재하며 플레이가 완료되면 다시 의미는 잠재태의 상태로 변환되기 때문이다. 이는 전통적 글쓰기의 텍스트에서 말하기의 주체인 독자가 새로운 의미를 생성시킬 수는 있지만 작가의 일차적 의미는 여전히 남아있는 것과는 대조된다.

마지막으로 전통적 작가의 글쓰기는 쓰기와 말하기의 완전한 분리로 인한 쓰기의 종료가 존재한다. 독자에 의해 작가의 글이 읽히는 순간, 쓰기는 종료되며 말하기의 상태로 작가의 글은 변환된다. 그러나 코딩에서 쓰기와 말하기는 동시적으로 중첩되어 나타나며 순간적인 의미를 발생시킨다. 때문에 끝없는 플레이어의 반복을 통해 지속적인 새로운 의미의 생성이 가능하다.

결국 사용자는 코딩 교육 게임의 플레이 과정을 통해 말하기와 글쓰기의 선행관계나 분리관계가 존재하는 전통적 글쓰기 행위보다 더욱 강력한 새로

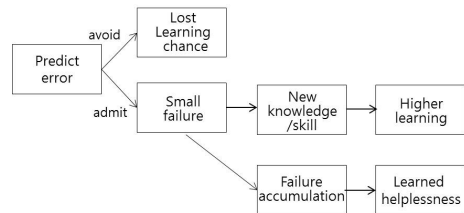
운 글쓰기를 체현하게 된다. 이는 의미의 발생, 수정 및 삭제, 종료라는 전 영역에서 이루어지며 수행적 글쓰기로서 코딩이 지닌 특성을 나타내고 있었다.

### 3.2 실패의 부재로 인한 문제공간의 확장

게임에서 실패는 동기를 부여한다. 우리는 실패를 극복하기 위해 학습하기 때문이다. 결국 좋은 게임은 잘 정의된 목적과 피드백을 가진 구조화된 놀이이다[29].

게임에서 뿐 아니라 학습 전반에 실패와 실수는 중요한 의미를 지닌다. 실수 기반 학습(Error-Driven Learning)에서 실수는 피해야 할 행동 혹은 회복해야 할 대상이다[30]. 또한 실수로 인한 부정적인 결과는 학습을 위한 정보(information)라기보다는 처벌(punishment)로 작용하여 실수를 감소시킨다[31]. 이러한 관점에서 실수는 올바른 행동의 반복적 실천을 통한 교정의 대상이다[32].

때문에 ‘실수의 교육적 역할 모델’[33][Fig. 3]은 실수를 허용했을 때 나타나는 작은 실패가 의도적 노력을 통해 새로운 지식과 기술을 만들어내고 이것이 통합되면서 고차원적 학습이 이루어짐을 보여준다. 만약 이러한 작은 실패에 무관심하다면 실패는 누적되고 소외현상이 일어나 학습된 무력감이 나타난다.



[Fig. 3] Educational Role Model from Error

결국 학습에 있어 실수가 만들어내는 실패, 이에 대한 교정과 반복은 학습자의 동기와 성취 모두에 영향을 미치는 주요한 요인이라 할 것이다.



한편 게임에서 ‘실패의 수사학’을 논한 윌슨은 더 많은 실패와 더 적은 처벌이라는 구조화된 방식이 결합될 때 더 많은 기회가 발생한다고 본다[34]. 실패에 대한 처벌이 적을수록 게임의 사용자는 적극적으로 참여하기 때문에 실수와 실패를 통해 학습할 있도록 더 많은 시도를 제공하라는 것이다. 그의 논의는 실수와 실패를 성공을 위해 교정해야 할 중독적 대상이 아닌, 독자적 의미를 지닌 요소로 파악했다는 점에서 의의를 지닌다.

<The Fools>에는 이러한 실패의 가능성이 애초에 존재하지 않는다. 캐릭터는 생명력 수치를 가지고 있지 않으며 코드 블록이 부적절하게 배열된 채 실행으로 이어졌다 하더라도 부정적인 피드백이 발생하지 않는다. 의도된 고난 및 지연이라 할 수 있는 아이템의 손실이나 시간제한과 같은 일반적인 패널티도 존재하지 않기 때문에 실패를 통한 처벌 역시 있을 수 없다. 실수를 했을 경우 나타나는 별도의 시그널도 없다.

그럼에도 불구하고 <The Fools>는 실패와 처벌, 그리고 교정이라는 구조화된 방식 없이도 사용자들의 동기를 발생시키며 이것이 다양한 시도로 이어지도록 만든다. 때문에 역설적으로 사용자는 실수와 실패에 대한 두려움 없이 자유롭게 다양한 코딩을 실험, 플레이하고 즉각적으로 그 결과를 확인할 수 있다. 이것이 교육용 게임에 나타난 수행적 글쓰기로서 코딩이 지닌 또 다른 특성이다.



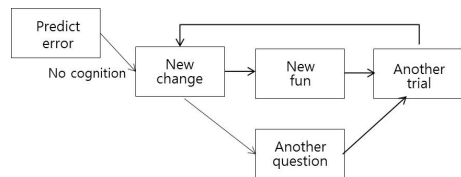
[Fig. 4] ‘Pile Up Box’ Loop Statement

수행적 글쓰기로서 <The Fools>의 코드 짜기가지니는 흥미로운 지점은 실수를 반복하는 행동 자체가 새로운 의미를 만들어 낸다는 것이다. 예를 들어 [Fig. 4][35]와 같이 상자를 쌓아 올리는 행위를

를 무한 반복하는 코드를 쓸 경우, 쌓여 올라가던 상자는 일정 높이에서 흔들리며 무너진다. 무너진 상자 위로 다시 상자가 쌓이고 이것이 다시 무너지기를 반복한다. 때문에 사용자는 상자더미에 깔린 캐릭터를 구출시킬 방법을 고민해 보아야 한다.

교정과 피드백은 실수를 인지하고 그 필요성을 스스로 인지할 때 효과적이다. 그러나 <The Fools>에는 사용자에게 실수를 자각시키기 위해 다른 게임처럼 행동이 정지되거나 처음으로 돌아가는 시스템이 없다. 실수로 인해 또 다른 의미 있는 변화가 공간에 생성될 뿐이다.

결국 <The Fools>에서 ‘실수의 교육적 역할 모델’은 다음과 같이 수정된다[Fig. 5]. 먼저 실패의 부재는 다양한 실수를 만들어 낼 수 있지만 이것은 실수로 인식되지 않기 때문에 실수의 허용과 회피라는 개념 자체가 성립되지 않는다. 때문에 작은 실패가 아닌 새로운 변화가 일어날 뿐이다. 이는 다시 새로운 게임의 재미로 연결된다. 이를 통해 하나의 목표를 성취하고자 하는 선형적 학습이 아닌 사용자가 자신들만의 목적을 설정하고 해결해 나가는 순환적 학습이 가능하게 된다.



[Fig. 5] Modified Educational Role Model from Error

이때 처음에는 모두 같은 지점에서 시작하였지만 각자 문제를 해결하고 도달할 수 있는 범위는 상이하며 그 방향과 가능성이 광범위하다는 점에서 문제 공간은 계속해서 확장되며, 사용자는 이러한 문제 공간을 자유롭게 탐험할 수 있다.

넓고 복잡한 문제공간을 주도적으로 탐색, 경험하면서 만나는 다양한 장애물은 학습에 긍정적인 역할을 하며 이러한 학습의 과정은 일종의 ‘즐거움 혼돈’의 과정이다[36]. 실패가 부재하는 <The

Foes>의 코드 짜기에서 이러한 문제 공간은 특유의 방식으로 무한히 확장되면서 다양한 코드 짜기의 가능성을 만들어낸다. 동시에 이에 따른 새로운 변화, 새로운 재미, 또 다른 시도라는 순환적 학습의 구조는 코딩 교육이라는 본연의 목적을 효과적으로 달성시킨다.

#### 4. 결 론

본 연구는 <The Foes>를 연구대상으로 하여 코딩 교육용 게임에 나타난 수행적 글쓰기에 대해 고찰하였다. SW중심 사회로의 변화가 이끈 코딩 교육에 대한 관심은 다양한 교육적 논의들로 이어졌지만 본 연구는 코딩을 수행적 글쓰기로 상정하고 코딩이 글쓰기로서 가지는 가능성을 살펴보았다. 이를 통해 기존의 코딩이 지니는 도구적, 공학적 관점을 인문학적 관점에서 좀 더 확장시키고 코딩이 오늘날 우리에게 가질 수 있는 구체적인 의미를 모색하고자 하였다.

연구 결과 글쓰기 과정으로서 코딩은 연쇄적 확장 순환의 작동 방식을 가지고 있었다. 이는 특히 기본적인 코드 블록을 활용한 학습의 방식에서 두드러지게 나타났다. 이어 코딩은 유닛 오퍼레이션으로서 의미를 생성하고 있었는데 이는 코드 블록을 활용해 게임을 플레이해가는 과정 및 인터페이스와 이것이 지시하는 플레이 메커니즘에서도 발견되었다.

다음으로 수행적 글쓰기로서 코딩은 말하기/쓰기의 중첩을 통해 의미를 생성, 수정, 종료 시키고 있었다. 이는 전통적 작가의 글쓰기와 변별되는 지점으로 사용자는 이러한 코드의 수행성을 통해 더 강력한 새로운 글쓰기 행위를 경험할 수 있었다. 한편 기존의 게임 및 학습에서의 경험과 달리 실패와 교정의 부재는 오히려 사용자의 문제 탐험 공간을 창발적으로 확장시키고 있었다. 사용자는 실수의 결과를 실패가 아닌 예상치 못한 변화나 재미로 받아들여지게 되며 이는 또 다른 경험을 이끄는 동인이 되었다.

본 연구는 수행적 글쓰기의 관점에서 구체적인 사례 분석을 통해 코딩과 코딩 교육이 지니는 변별적 특성을 모색하는 탐색적 연구로 그 의의가 있다. 본 연구가 융합교육 환경에서 코딩과 코딩 교육에 대한 더욱 다양한 가능성들을 논하는 단초가 될 수 있기를 기대한다.

#### ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government(NRF-2007-361-AM0013) of 2007.

#### Reference

- [1] SPRI, Software Centered Society: The Meaning and Direction of Reaction, SPRI Issue Report 2014-003, 2014.
- [2] Park, J. H., "Effects of Storytelling Based Software Education on Computational Thinking", JOURNAL OF The Korean Association of information Education, 19(11), p.58, 2015.
- [3] Wing, J. M., "Computational thinking", Communications of the ACM, 19(3), pp.33-35, 2006.
- [4] Park, H. M., "Global SW education trend and tools", Internet & Security Focus, p.41, 2014.
- [5] Song, T. O., "The Purpose and Direction of Software Education", The Korean Association of Computer Education, 19(2), pp.23-26, 2015.
- [6] Kim, J. M., & Lee, W. G., "China's informatics curriculum to consider equality and excellence", The Journal of Korean association of computer education, 18(2), pp.11-20, 2015.
- [7] Shin, S. K., & Bae, Y. K., "Review of Software Education based on the Coding in Finland", JOURNAL OF The Korean Association of information Education, 19(1),

- pp.127-138, 2015.
- [8] Sung, J. S., & Kim, H. C., "Analysis on the International Comparison of Computer Education in Schools", The Journal of Korean association of computer education, 18(1), pp.45-54, 2015.
- [9] Park, J. H., (2015), pp.57-68.
- [10] Jin, Y. H., & Kim, Y. S., "A Meta-Analysis on the Effects of Educational Programming Language", The Journal of Korean association of computer education, 14(3), pp.25-36, 2011.
- [11] Lee, E. H., & Lee, T. W., "Instruction Model for Elementary School on Programming Induction Education Using ENTRY", The Korean Association of Computer Education, 19(1), pp.43-46, 2015.
- [12] Lyou, M. Y., & Han, S. K., "Development of Computational Thinking-based Educational Program for SW Education", JOURNAL OF The Korean Association of information Education, 19(1), pp.11-20, 2015.
- [13] Kim, S. H., et al., "Analysis about User Log for Development of Online SW Education Platform in Korea", The Korean Association of Computer Education, 19(2), pp.63-67, 2015.
- [14] Park, J. H., (2015), p.59.
- [15] ibid.
- [16] Park, H. M., "Global SW education trend and tools", Internet & Security Focus, pp.51-53, 2014.
- [17] Pollock, D., "Performing writing", The ends of performance, pp.75-76, 1998.
- [18] Baeyer, v., & Christian, H., Information, New language of science, Sungsan, p.49. 2007.
- [19] Yun, H. J., THE USER STORYTELLING OF <MINECRAFT>, Doctoral dissertation, EWHA WOMANS UNIVERSITY, pp.23-24, 2015.
- [20] Berry, David M., The Philosophy of Software: Code and Mediation in the digital age, NY: Palgrave Macmillan, pp.29-30, 2011.
- [21] Joasia, K., & Grzesiek, S., "Source Code", Software Studies: A Lexicon, MA: The MIT Press, p.239, 2008.
- [22] Flusser, V., Writing in the digital age, Munye, p.139, 1998.
- [23] Bogost, I., Unit Operation, MA: The MIT Press, 1997.
- [24] Flusser, V., (1998), p.19.
- [25] Galloway, A. R., Protocol: How Control Exists after Decentralization, MA: The MIT Press, 2004.
- [26] Fredkin, Edward, "Introduction to digital Philosophy", International Journal of Theoretical Physics, 42(2), pp.189-247, 2003.
- [27] Hayles, N. K., My mother was a computer: digital subjects and literary texts, Univ of Chicago Press, pp.39-42, 2005.
- [28] Cox, G., & Mclean, A., Speaking Code: Coding as Aesthetic and Political Expression, MA: The MIT Press, pp.34-35, 2012.
- [29] Japer, J., The art of failure: An essay on the pain of playing video games, MA: The MIT Press, pp.9-10, 2013.
- [30] Lannin, J. K., & David D. B., Brian E. T., "How students view the general nature of their errors", Educational Studies in Mathematics 66(1), pp.43-59, 2007.
- [31] Ohlsson, S., "Learning from errors", Psychological Review, 103, pp.241-262, 1996.
- [32] Kim, J. B., "Educational Implications of Error-Driven Learning", Korean Educational Psychology Association, 24(4), p.897, 2010.
- [33] Kim, J. B., (2010), p.900.
- [34] Japer, J., (2013), p.72.
- [35] <https://www.kidscodecs.com/eak-the-foos-tynker-run-marco/>
- [36] Kim, J. B., (2010), pp.908-909.



윤현정(Yun, Hyun Jung)

약력 : 2004 이화여자대학교 광고홍보학과  
2009 이화여자대학교 대학원 디지털미디어학부 석사  
2015 이화여자대학교 대학원 디지털미디어학부 박사  
현재 인하대학교 한국학연구소 HK연구교수

관심분야 : 게임, 디지털스토리텔링, 소프트웨어 연구

— 코딩 교육용 게임에 나타난 수행적 글쓰기 - <The Foes>를 중심으로 —