

소프트웨어 프로그래밍교육을 통한 컴퓨팅 사고력 개발에 관한 국내 연구 고찰

서울대학교 교육학과 교육공학 전공
석사 과정 박주현

목차

1

서론

2

이론적 배경

3

연구 방법

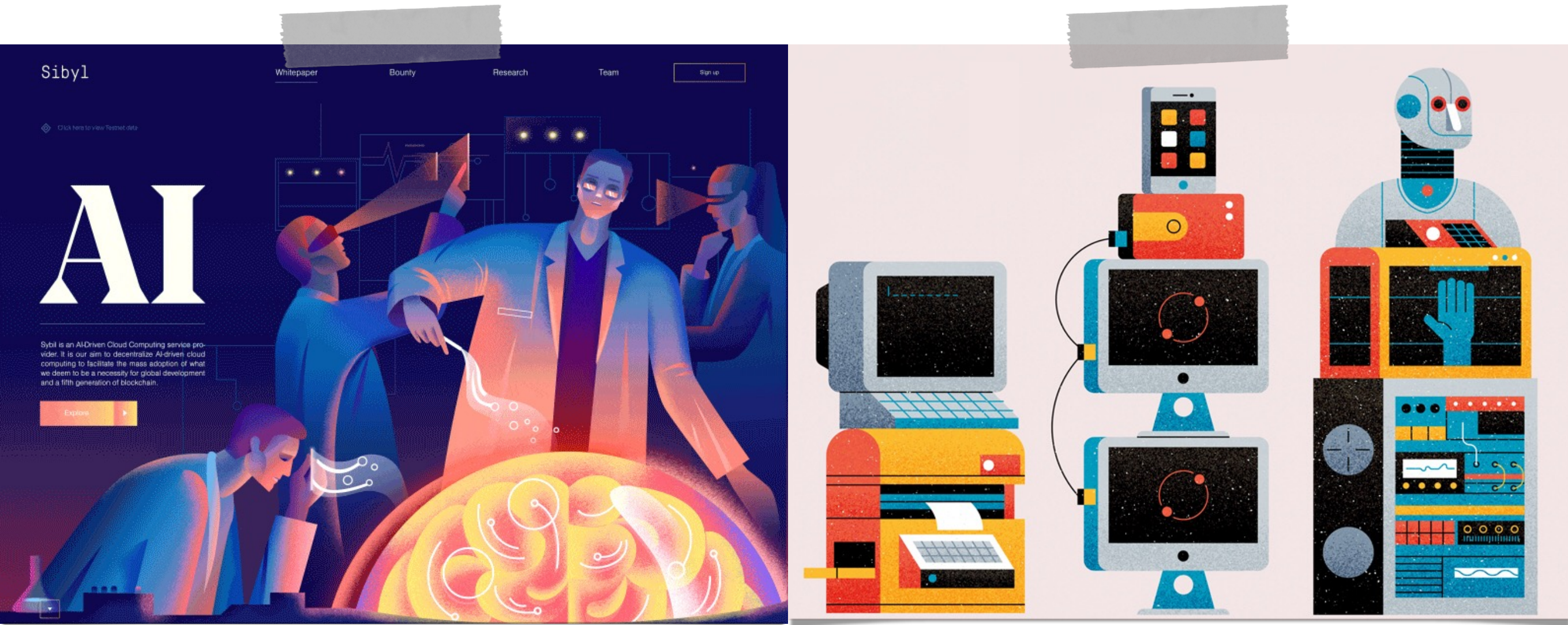
4

연구 결과

5

결론 및 논의

소프트웨어교육의 중요성



4차 산업혁명의 주요 기술로 여겨지는 인공지능, 사물 인터넷, 빅데이터, 3D 프린팅, 클라우드 서비스 등의 핵심은 **소프트웨어(Software: SW)**

소프트웨어교육의 중요성

Jeannette M. Wing

소프트웨어교육을 통해 배양해 주고자하는 역량은 “문제를 구성하고 그 문제의 해결책을 제시하는 사고 과정에서 정보처리 체계에서 효과적으로 처리될 수 있는 형태로 제시할 수 있는” 능력, 즉 **컴퓨팅 사고력(Computational Thinking: CT)**이다.

소프트웨어교육의 중요성

Jeannette M. Wing

컴퓨팅 사고력을 자신의 전공 영역에 적용하여 그전에는 생각하지 못했던
새로운 문제를 구성하고 해결하기 어려웠던 **복잡한 문제를 해결**



“컴퓨팅 사고력을 보편적 리터러시로 가르쳐야 한다”

소프트웨어교육의 종류와 프로그래밍 교육

사고력 교육

- 언플러그드 형태

프로그래밍교육

- 교육용 프로그래밍 언어를 활용

피지컬 컴퓨팅 교육

- 하드웨어와 소프트웨어를 연결하여 컴퓨팅 원리 구현

프로그래밍교육은 세부적인 컴퓨팅 사고력 역량을 배양할 수 있게 하는 잠재력을 가지고 있기 때문에 중요한 교육 방법

연구 문제

- 1 국내 소프트웨어 프로그래밍교육은 어떠한 교수 접근(interventions)을 활용하여 이루어지고 있는가?
- 2 국내 연구들은 프로그래밍교육을 통한 컴퓨팅 사고력 개발 효과로 무엇을 평가하고 있는가?

이론적 배경

컴퓨팅 사고력의 정의

Papert, 1980 -> Wing, 2006 -> Wing, 2010

컴퓨팅 사고력: 문제를 구성하고 그 문제에 대한 해결책을 제시함에 있어 정보-처리 체계(Information-processing agent)에 의해 효과적으로 수행될 수 있는 형태로서 제시하는 사고 과정(Thought processes)

이론적 배경

컴퓨팅 사고력 프레임워크의 3차원(Brennan & Resnick, 2012)

차원	세부 내용	설명
컴퓨팅 개념	순차 구조	과제를 위한 일련의 단계를 정의하는 것
	조건 구조	조건에 따라 결정하도록 만드는 것
	반복 구조	같은 순차를 필요한 만큼 실행하는 것
	연산자	수학적, 논리적 표현식에 따라 계산하는 것
	병행 처리	동시에 일이 일어나도록 만드는 것
	데이터(변수)	데이터 값을 저장, 검색, 수정하는 것
	이벤트	이것을 통해 다른 것이 일어나도록 만드는 것
컴퓨팅 수행	점진적인 시도-개발	조금 개발하며 시도해 보고, 다시 발전시켜서 개발하는 것
	테스팅-디버깅	정확하게 만들고 오류를 찾아내서 고치는 것
	재사용-재조합	자신이 만든 것이나 다른 사람이 만든 것을 이용해서 만드는 것
	추상화-모델화	전체와 부분들의 연관성을 탐색하거나 재사용이 가능한 코딩 블록들의 단위를 만드는 것
컴퓨팅 관점	표현하기	컴퓨팅을 하나의 도구로 인식해서 스스로 만들어 내는 것
	연계하기	다른 사람과 협력하고, 다른 사람을 위해 만들어 내는 것
	질문하기	세상의 의문점에 대해 컴퓨팅을 사용해서 탐구하는 것

이론적 배경

컴퓨팅 사고력의 중요성 및 프로그래밍과의 관계



소프트웨어는 잘 정의된 알고리즘이 **추상화**를 통해 만들어지고,
만들어진 결과물이 필요할 때 수행되거나 **자동화**되어 운영될 수 있는 실체

추상화

- 컴퓨팅 사고력에 내재된 **고차원 사고**에서 가장 중요한 것
- 특정 사례들로부터 일반화하여 패턴을 정의하고 **매개화 (parameterization)**하는데 사용됨
- 확장될 수 있는 능력(power to scale)과 복잡성(complexity)을 해결할 수 있는 능력을 가짐

자동화

- 추상화 과정에서 만들어진 해결 모델을 컴퓨터가 이해할 수 있는 프로그래밍 언어로 표현하는 것



컴퓨팅 사고력과 프로그래밍의 관계

프로그래밍 과정 중에 경험하는 고차원적 세부 역량들(문제의 분해, 추상화, 알고리즘적 사고, 평가, 일반화)은 컴퓨팅 사고의 핵심 구성요소들이므로 프로그래밍 및 코딩을 가르치는 것은 컴퓨팅 사고력을 길러주는 중요한 방법

SW 교육관련 문헌 연구 동향

4차 산업혁명시대의 도래로 국내 소프트웨어교육 관련 동향 연구가
양적인 증가를 보이고 있음

그러나

소프트웨어교육의 범위가 넓고 주제도 다양하여 컴퓨팅 사고력 증진을
목표로 설계한 수업은 일부이며, 소프트웨어교육을 통한 컴퓨팅 사고력 개발 효과로서
무엇이 평가되었는지 분석한 연구 부족

연구 대상 및 문헌 검색 절차

-2016~2018년도

-국내에서 컴퓨팅 사고력 향상을 위해 프로그래밍 교육을 설계 및 운영하고 컴퓨팅 사고력에 미치는 영향을 연구한 논문

-한국학술정보원서비스(Riss4u.net)에서 연도별로 국내 학술연구지 대상

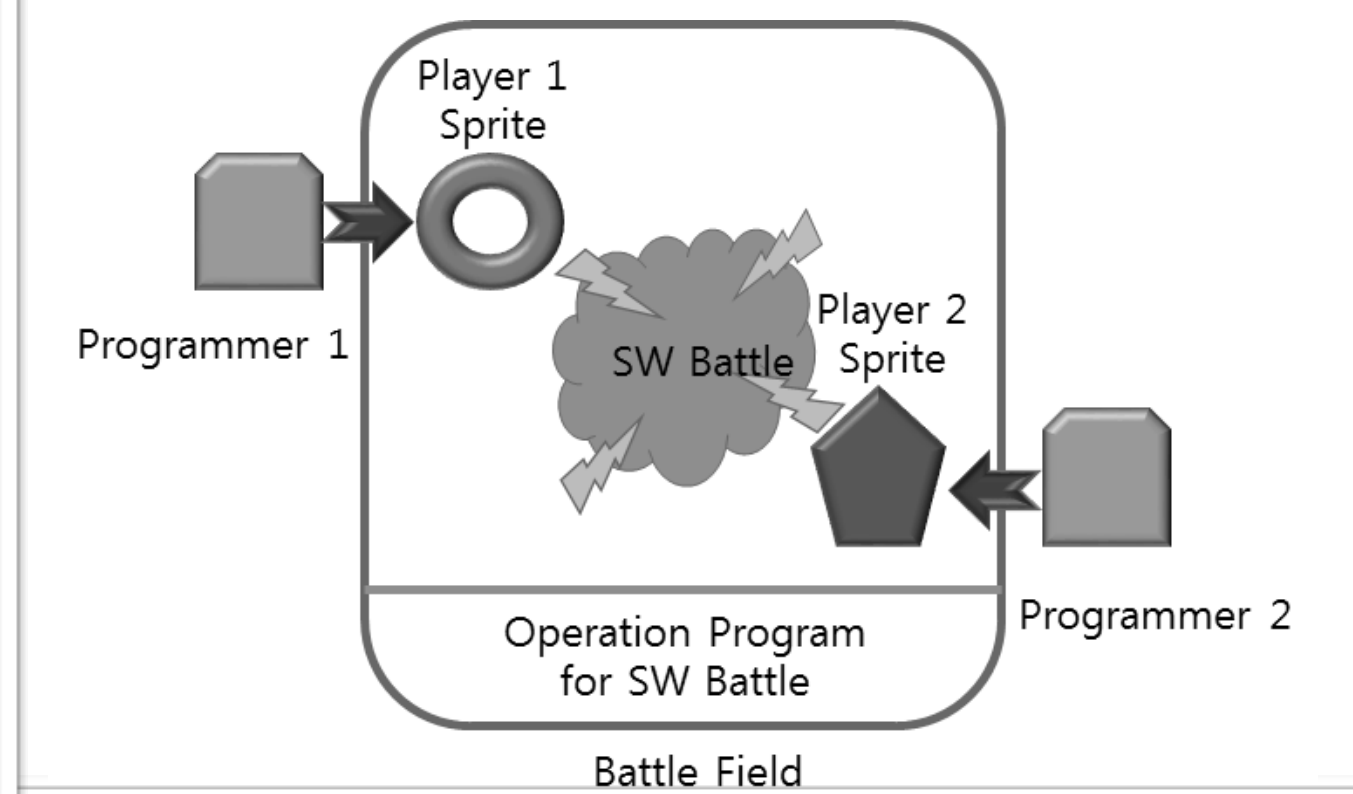
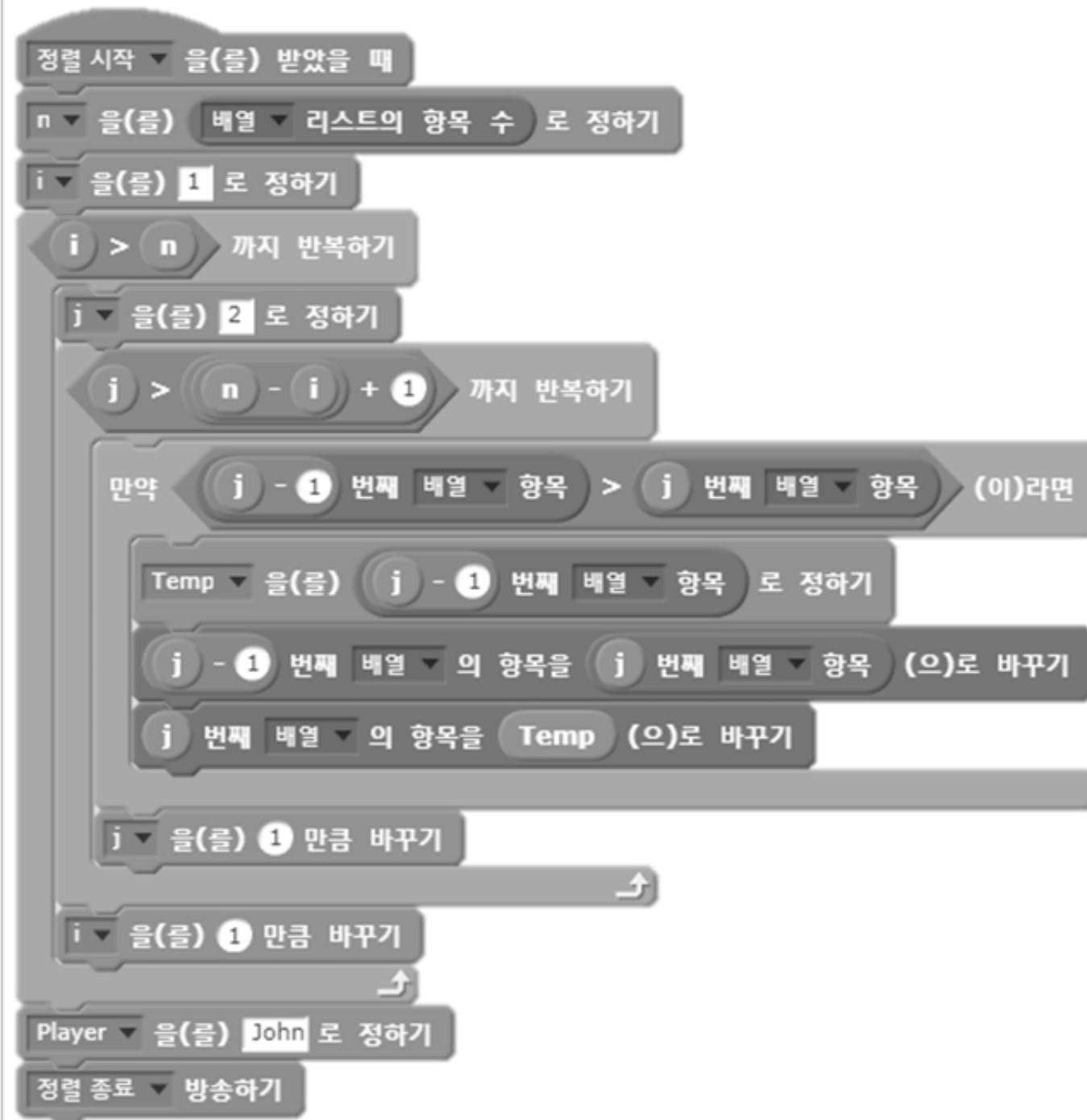
-총 32편

문헌 분석 방법

- 1 연구 대상 논문에서 기술한 연구 방법, 연구 절차, 수업 내용, 교수 모형, 교수 방법 및 전략 등을 상세히 검토하고 교수 접근들의 유사점 및 차이점을 분석하여 유목화
- 2 Brennan과 Resnick(2012)의 컴퓨팅 사고력 프레임워크를 기반으로 3개의 차원으로 분석

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

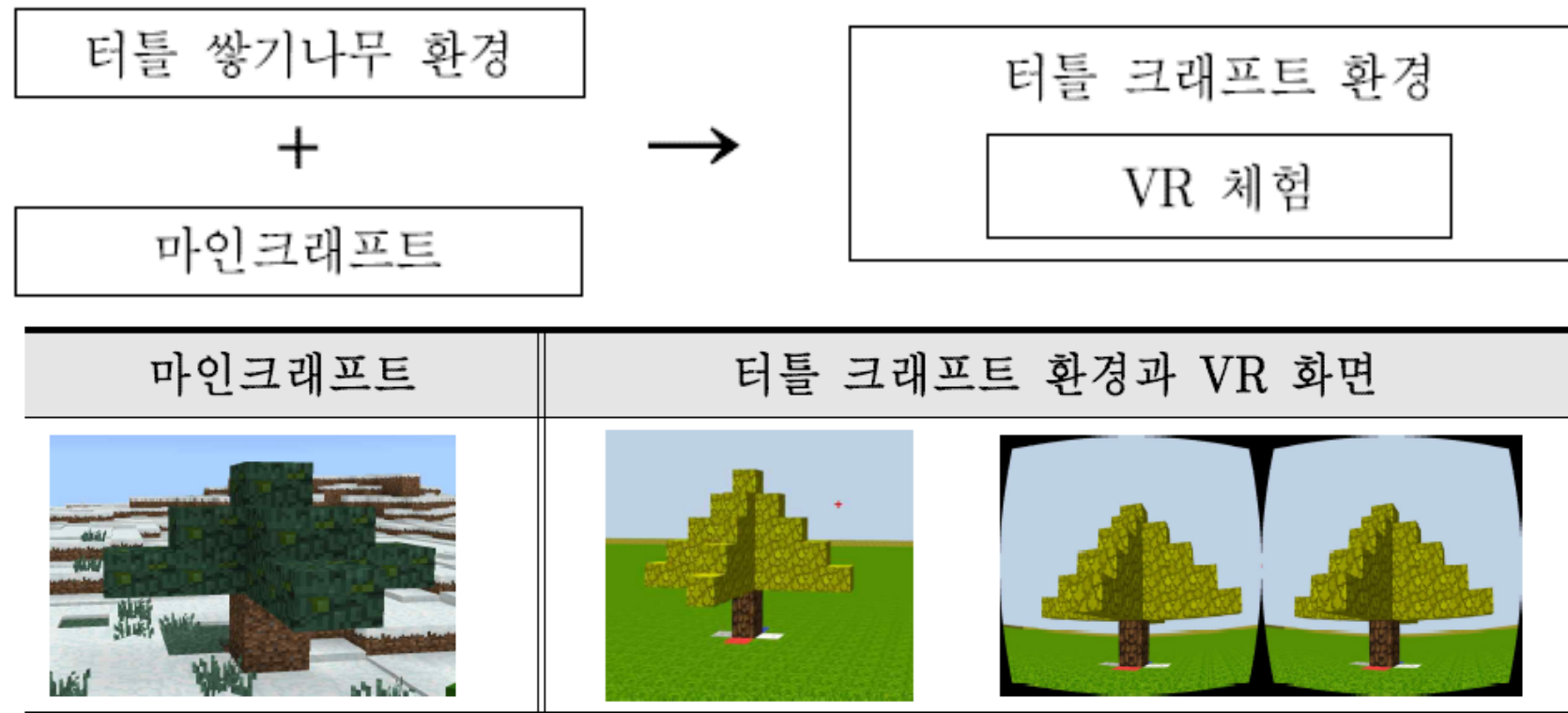
1. 문제해결적 접근



퍼즐, 배틀, 미션을 제공하는 문제해결적 접근

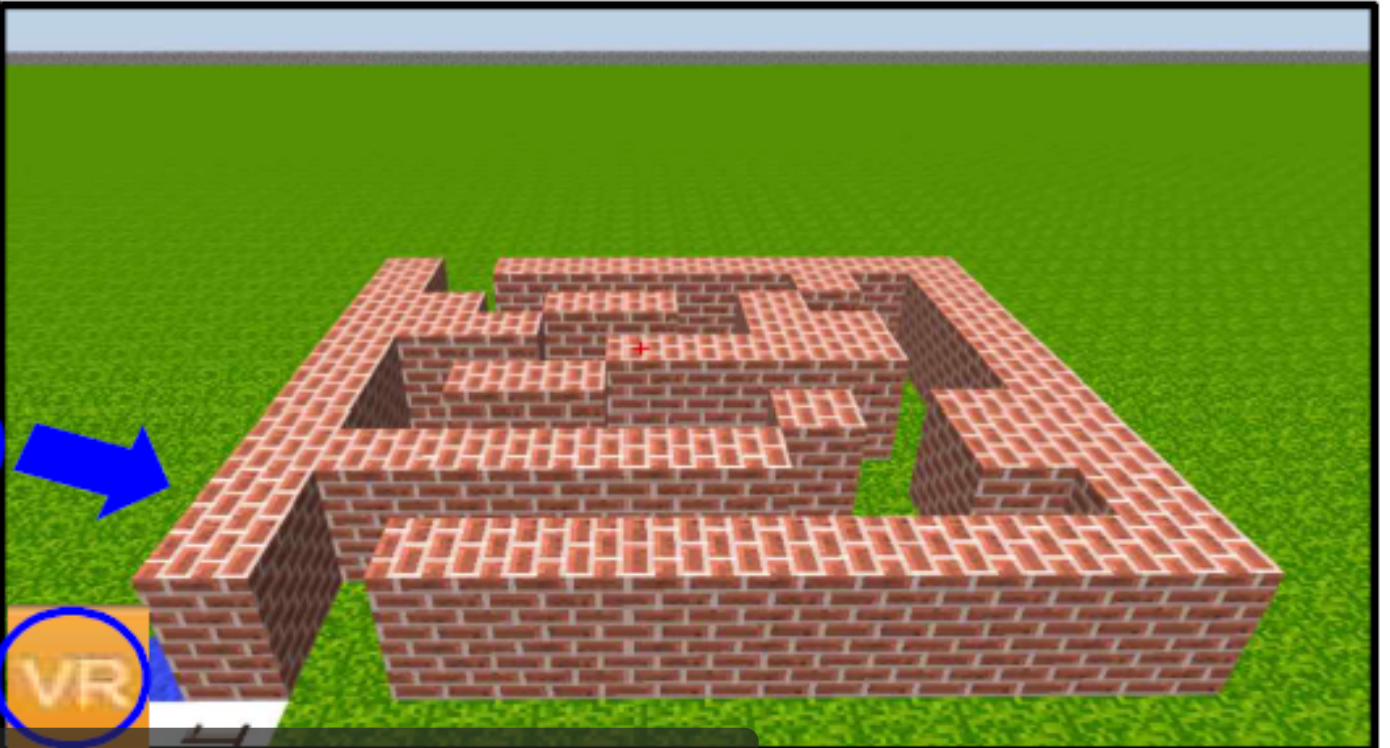
연구 결과 - 프로그래밍 교수 접근 유형 및 사례

1. 문제해결적 접근



교당수학	init	TT	cls	STOP	큐브식	좌표식	터복이
기호 입력 창	$S = ' S[u] '$ do $S[L9S] \cdots SL9S$						

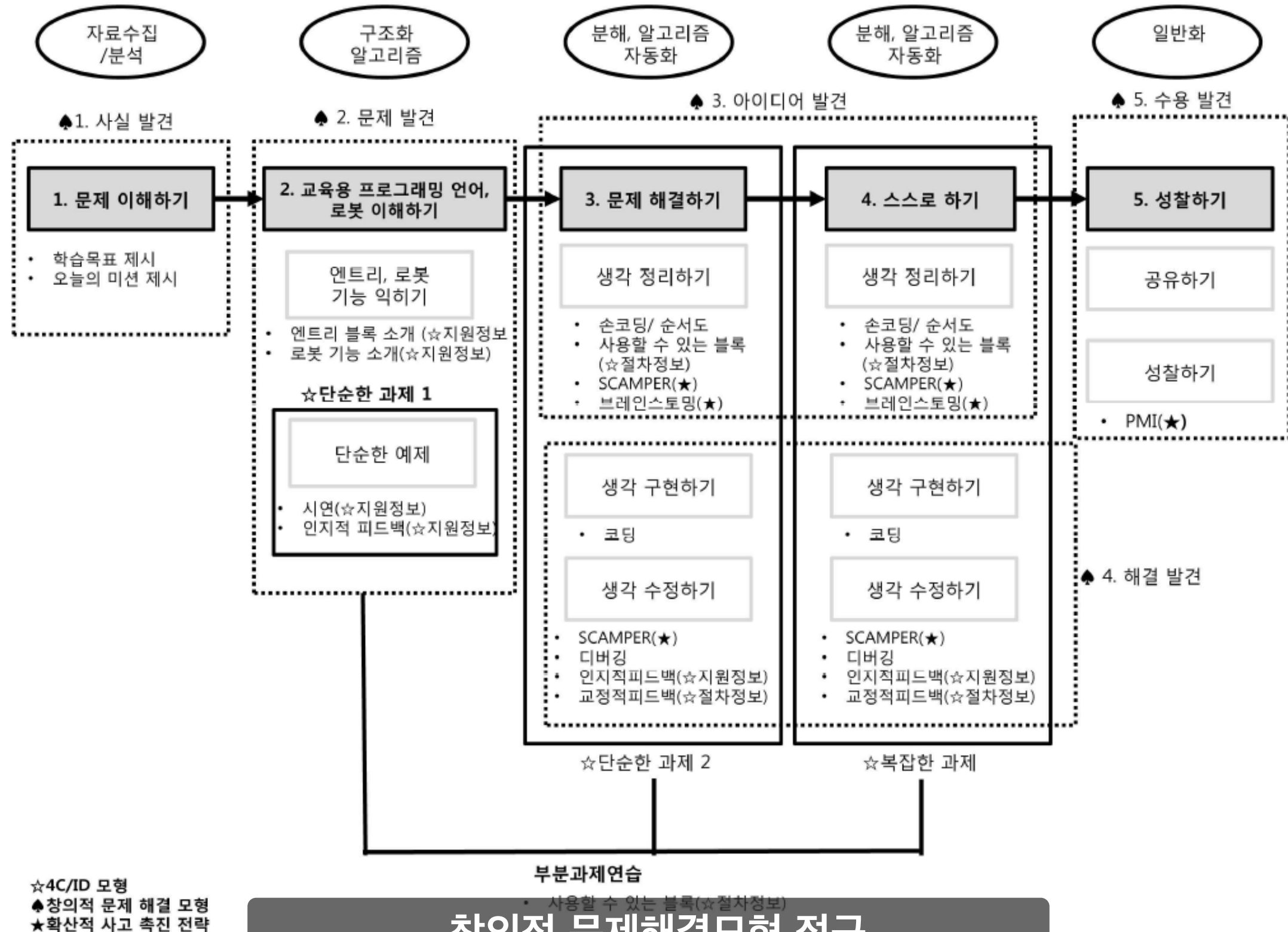
VR



퍼즐, 배틀, 미션을 제공하는 문제해결적 접근

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

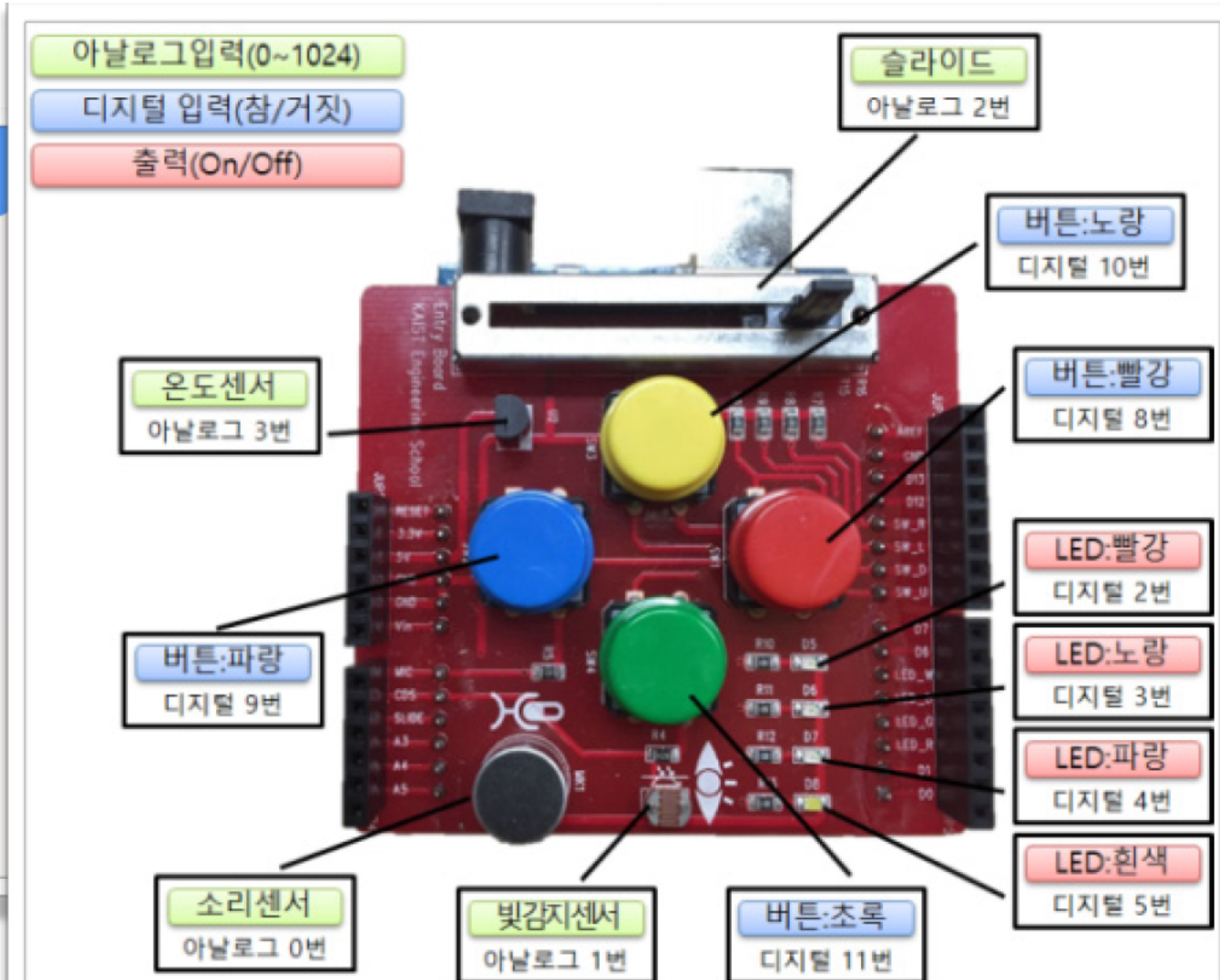
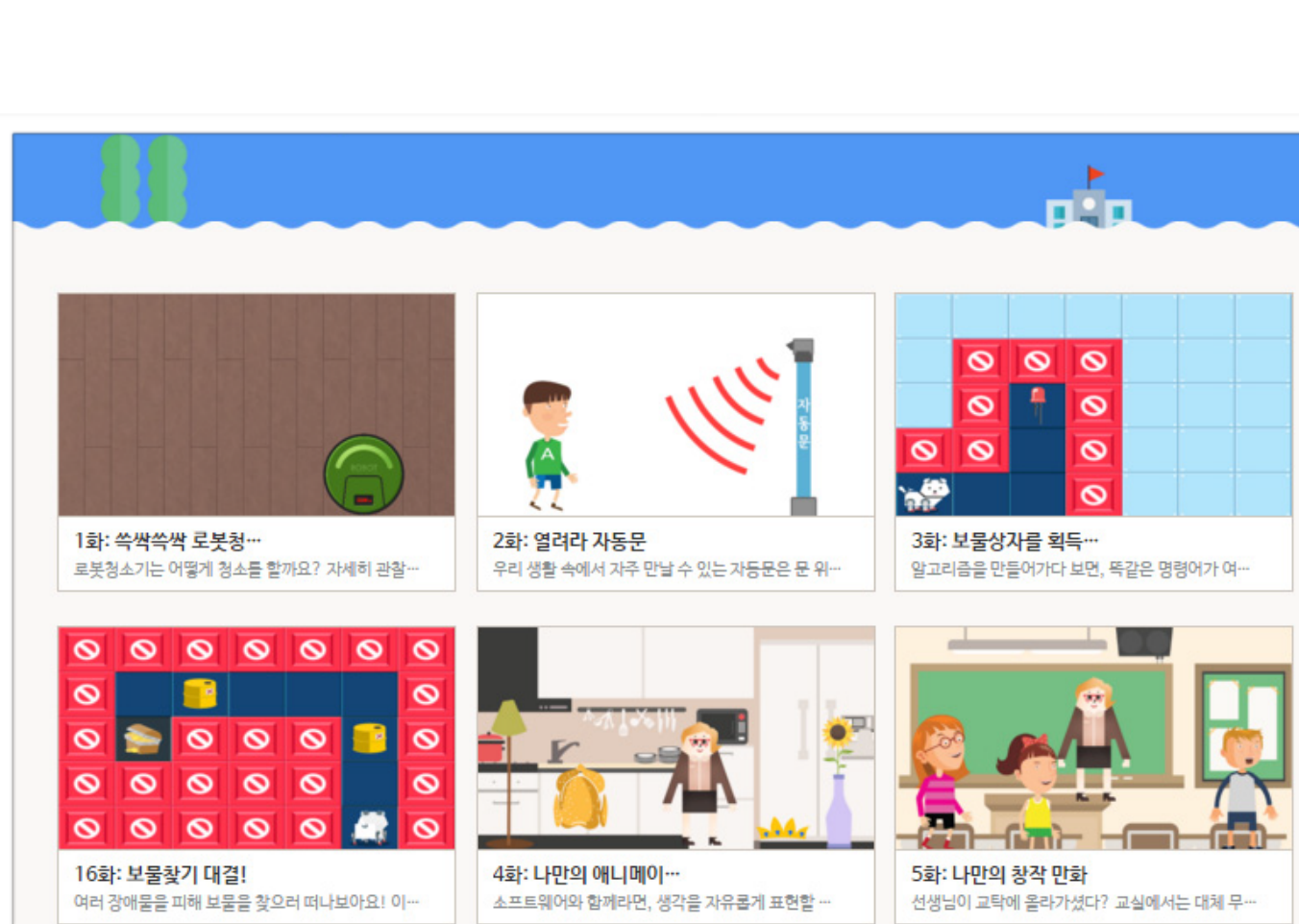
1. 문제해결적 접근



창의적 문제해결모형 접근

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

1. 문제해결적 접근



피지컬 컴퓨팅을 활용한 실제생활 연계 메이킹 활동

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

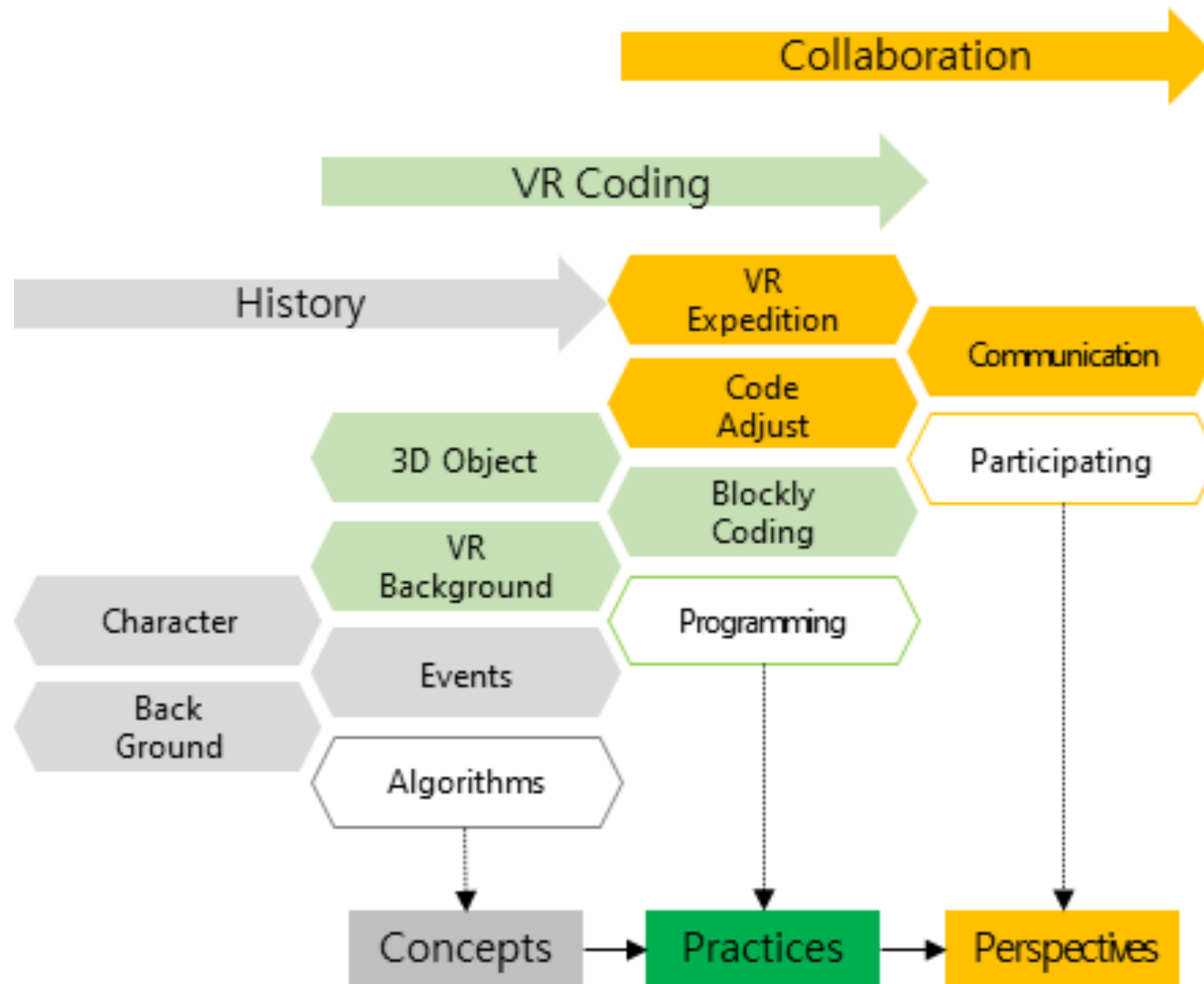
2. 협력적 상호작용 접근

<Table 1> The role of Pair Programming

Role	Work	etc
Driver	<ul style="list-style-type: none">– Working with keyboard and mouse– Programming code to work– Design Coding	<ul style="list-style-type: none">– Communicating with each other
Navigator (Observer)	<ul style="list-style-type: none">– Finding the wrong character on the coding– Finding grammatical errors– Visualize algorithms that We need to	<ul style="list-style-type: none">– Exchanging roles between colleagues

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

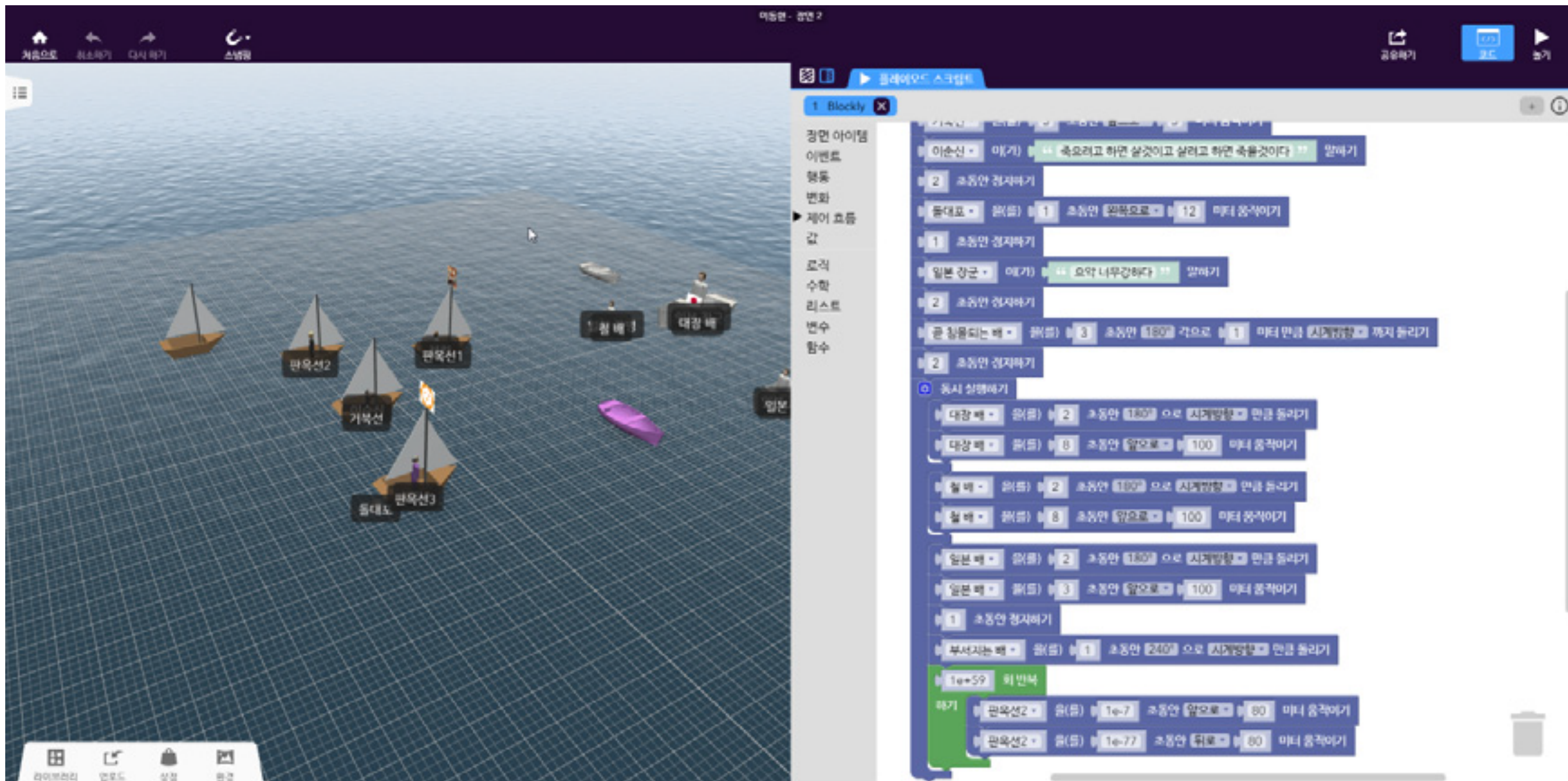
2. 협력적 상호작용 접근



팀프로젝트 및 팀워크 접근

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

2. 협력적 상호작용 접근



팀프로젝트 및 팀워크 접근

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

3. 성찰기반 접근

〈표 4〉 성찰기반 코딩교육에 대한 효과

범주	개념빈도	백분율
내용 재생 및 파지	64	35.6%
사고력 및 문제해결력	52	28.9%
성취감	30	16.7%
창의력	18	10.0%
소통 및 협업능력	16	8.9%
계	180	100%

성찰일지 활용

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

3. 성찰기반 접근

〈표 5〉 “내용 파지 및 전이” 관련 진술

“성찰일지 작성으로 배웠던 내용을 되새길 수 있었고 나중에 기억이 안 나는 부분에서 성찰 일지를 다시 찾아보면 그때의 기억이 확실하게 되살아납니다.”

“학습내용을 적으면서 배운 내용을 복습할 수 있었다. 그리고 종합의견을 적으면서 수업 때 놓친 부분이나 생각하지 못한 내용을 떠올릴 수 있었고 개선할 점을 적으면서 나의 부족한 점과 고칠점을 찾을 수 있었다. 또한 피드백을 받으며 내가 몰랐던 나의 개선할 점과 잘한 점을 알 수 있었다.”

“익숙하지 않고 자주 사용하지 않는 것이라 쉽게 까먹기 일수입니다. 하지만 성찰을 하며 다시 되새겨 봄으로써 조금 더 오래 기억할 수 있게 되었습니다.”

성찰일지 활용

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

3. 성찰기반 접근

<Table 2> Self-Assessment Rubric

Category	Level 1	Level 2	Level 3	Level / Evidence
Algorithmic thinking	The procedures for problem solving have not defined clearly.	The procedures for problem solving have defined clearly.	The procedures for problem solving have defined clearly and the flow is visually presented.	
Algorithm Evaluation	There are some inaccuracy in algorithms.	The algorithms are accurate but there are some inefficiency.	All algorithms are accurate and efficient.	
Problem Decomposition	No custom blocks are used.	Problems are divided in to logical units in custom blocks but there seems more decomposition is necessary.	All custom blocks are logically constructed.	
Abstraction	No abstraction was attempted.	Abstraction was made but some of them is not appropriate.	Abstraction was made properly overall.	
Generalization	No reuse of previously used algorithms was made.	Reuse of algorithms was made but some inefficiency is found.	Reuse of algorithms was made for different purposes.	

자기평가 루브릭 활용

연구 결과 - 프로그래밍 교수 접근 유형 및 사례

3. 성찰기반 접근

1. 문제 상황

엘리스는 이상한 나라에 도착하였습니다. 엘리스가 이상한 나라에서 자기 집으로 안전하게 돌아가기 위해서는 우리가 도와주어야 합니다.

엘리스는 아래의 조건과 같은 방법으로 이동할 수 있습니다.

엘리스가 이상한 나라에서 자기 집으로 돌아가기 위해서 위, 아래, 왼쪽, 오른쪽으로 이동하는 프로그램을 작성해 봅시다.

※ 조건) 키보드의 키를 사용하여 위, 아래, 왼쪽, 오른쪽으로 이동

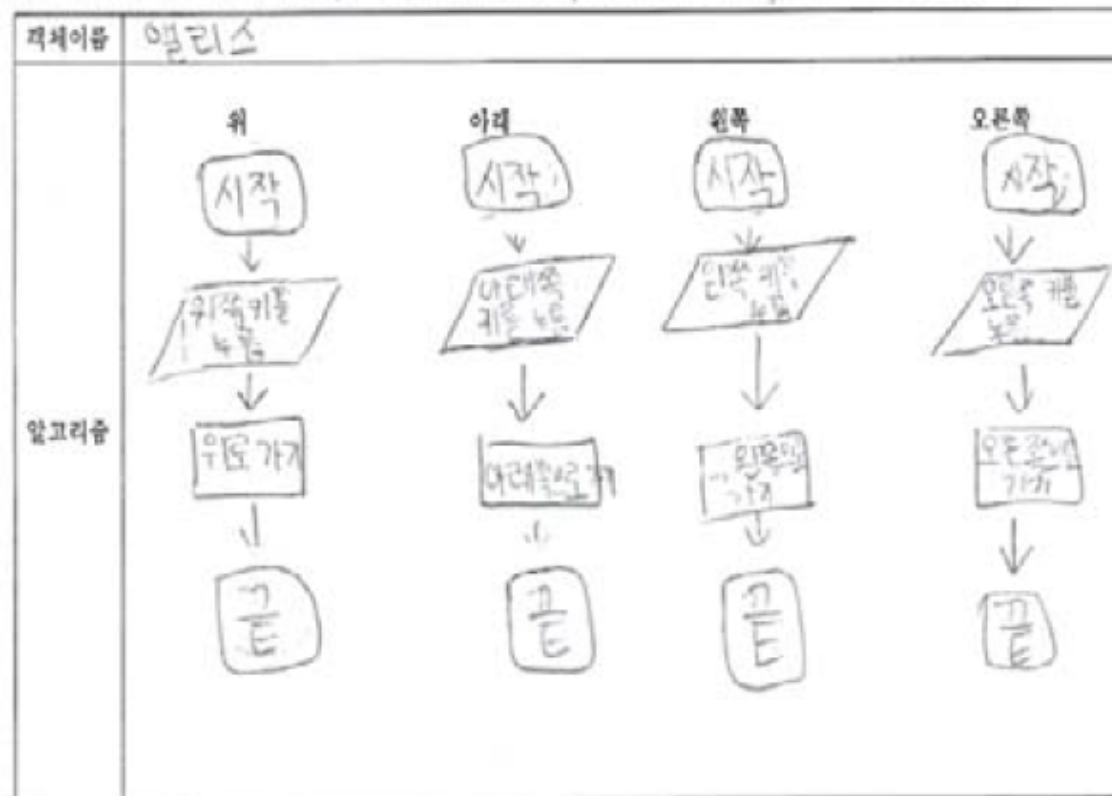
2. 필요한 객체 및 자원

엘리스, 이상한 나라(배경)

3. 객체별 알고리즘과 순서도

■ 문제 상황을 해결하기 위해서 필요한 알고리즘은 모두 몇 개입니까? (4개)

■ 각각의 키 정하기: 위 (위쪽) 키, 아래 (아래쪽) 키, 왼쪽 (왼쪽) 키, 오른쪽 (오른쪽) 키



1. 문제 상황

엘리스는 이상한 나라에 도착하였습니다. 엘리스가 이상한 나라에서 자기 집으로 안전하게 돌아가기 위해서는 우리가 도와주어야 합니다.

엘리스는 아래의 조건과 같은 방법으로 이동할 수 있습니다.

엘리스가 이상한 나라에서 자기 집으로 돌아가기 위해서 위, 아래, 왼쪽, 오른쪽으로 이동하는 프로그램을 작성해 봅시다.

※ 조건) 키보드의 키를 사용하여 위, 아래, 왼쪽, 오른쪽으로 이동

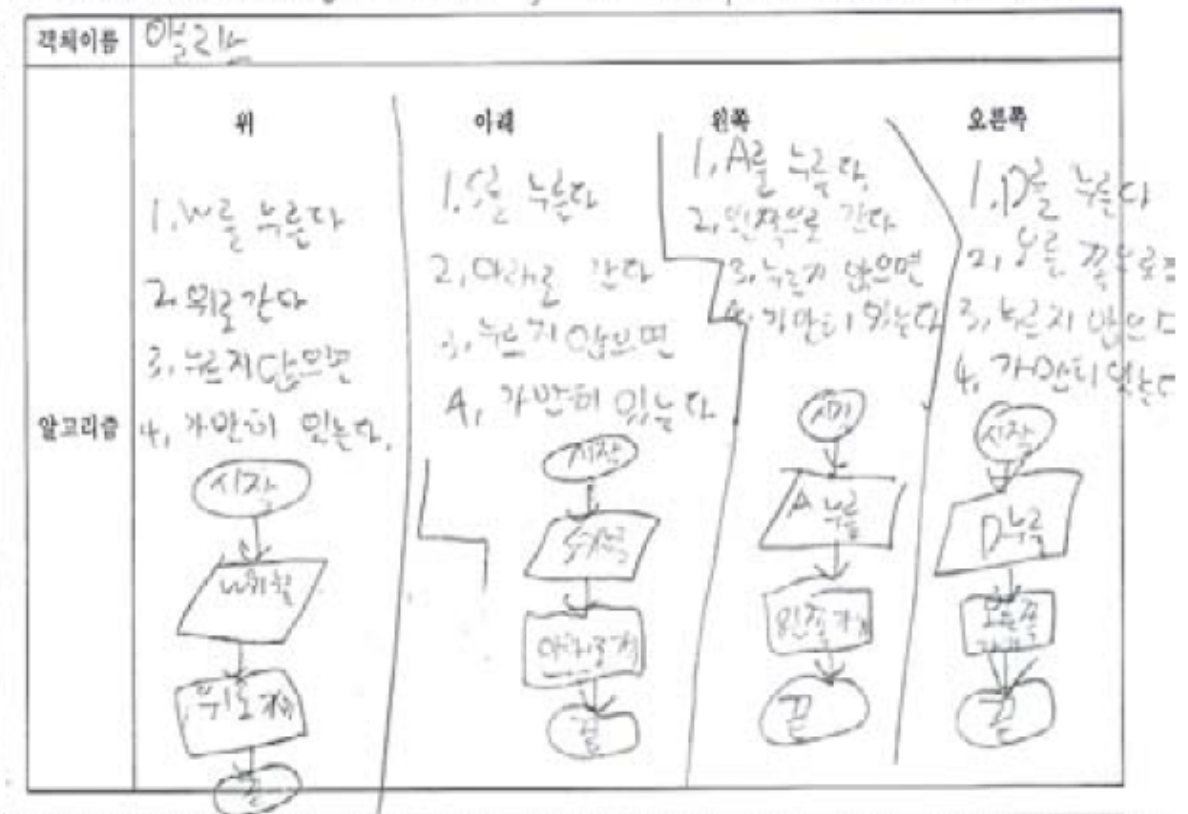
2. 필요한 객체 및 자원

엘리스, 이상한 나라(배경)

3. 객체별 알고리즘과 순서도

■ 문제 상황을 해결하기 위해서 필요한 알고리즘은 모두 몇 개입니까? (4)

■ 각각의 키 정하기: 위 (W) 키, 아래 (S) 키, 왼쪽 (A) 키, 오른쪽 (D) 키

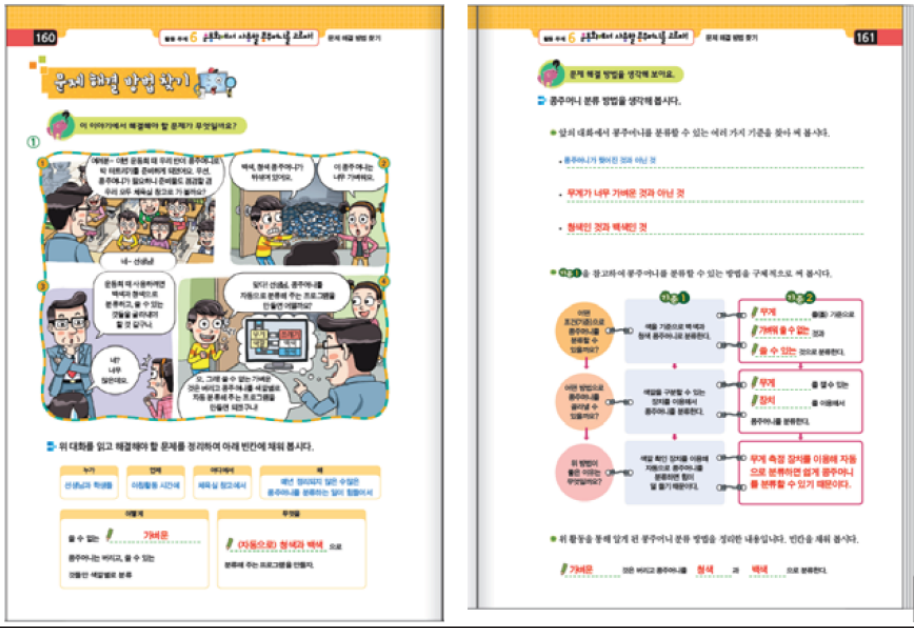
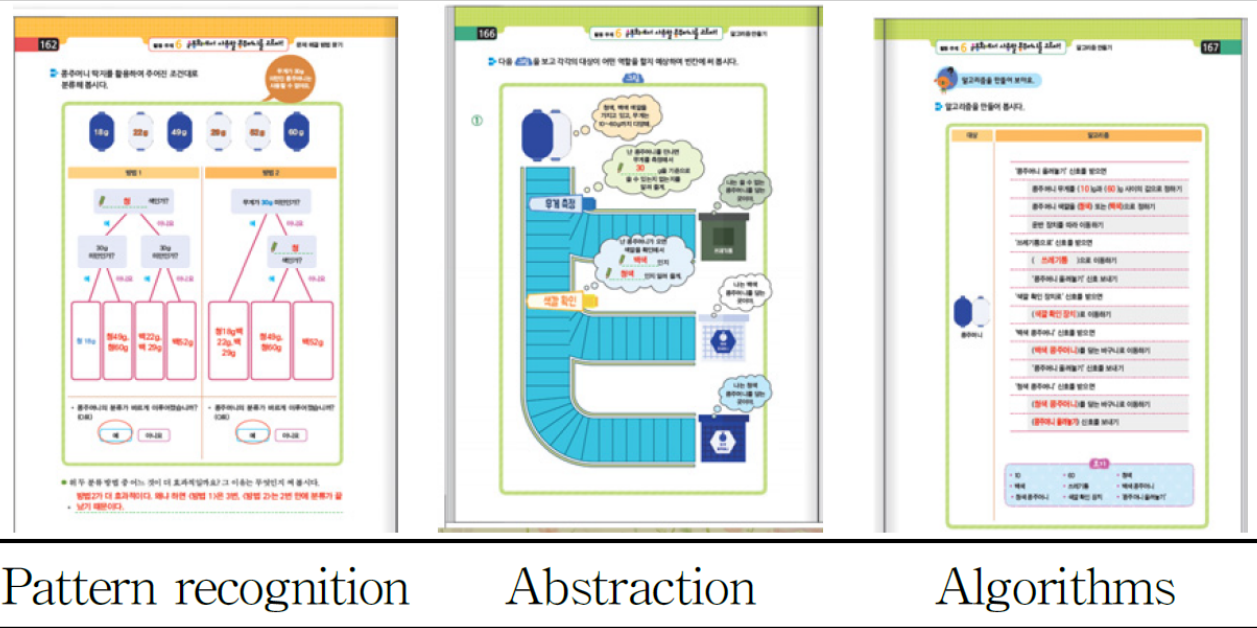


연구 결과 - 프로그래밍 교수 접근 유형 및 사례

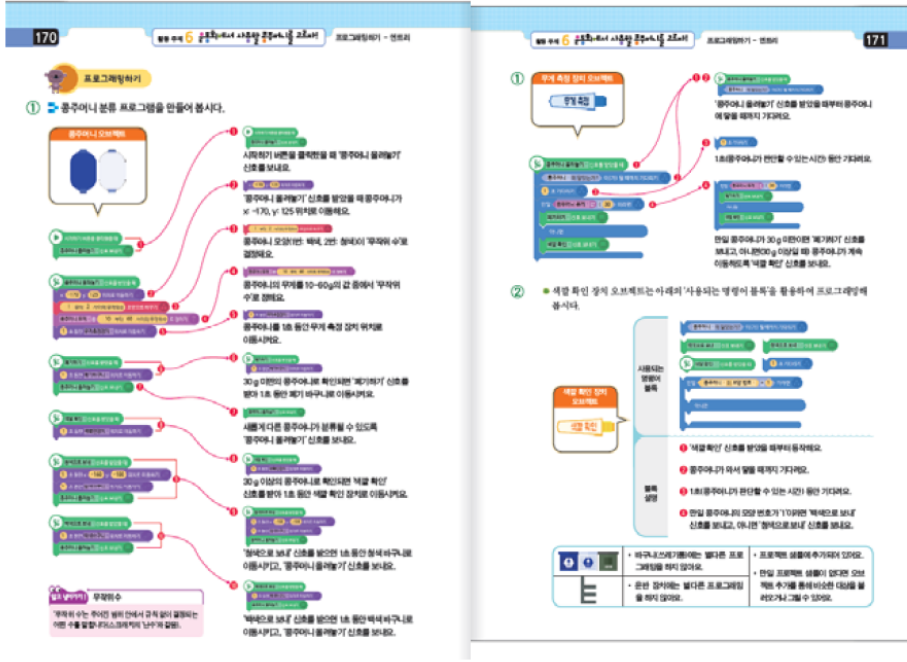
4. 놀이중심 발견학습 및 디자인 접근

<Table 3> Examples of CT in the Discovery step

<Table 4> Examples of CT in the Design step

CT	Project example (bean bag classifier[8])	Project example (bean bag classifier[8])
Decomposition		
	Pattern recognition	Abstraction
		Algorithms

<Table 5> Examples of CT in the Development step

CT	Project example (bean bag classifier[8])
Automation	

연구 결과 - 프로그래밍교육의 효과 평가

평가 내용	평가 방법	연구 방법	연구 결과
컴퓨팅 개념	<ol style="list-style-type: none"> 1. 설문 조사 방식 2. 선다형 지필 시험 방식 	<ol style="list-style-type: none"> 1. 실험집단 사후검사 2. 실험집단 사전-사후검사 3. 실험-비교집단 사전-사후검사 4. 질적 분석 	<ol style="list-style-type: none"> 1. 컴퓨팅 사고력 수준 2. 컴퓨팅 개념 수준 3. 성별에 따른 차이 4. 컴퓨팅 수행 수준 5. 기타 <p>통계적으로 유의했다 혹은 유의하지 않았다</p>
컴퓨팅 수행	<ol style="list-style-type: none"> 1. 설문 조사 방식 2. 전문기관에서 개발한 검사지 3. 프로젝트 평가 및 서술형 수행 평가 4. 자동화된 수행 평가 도구 		
컴퓨팅 관점	<ol style="list-style-type: none"> 1. 설문 조사 방식 2. 성찰 일지 		

결론 및 논의

1 국내 소프트웨어 프로그래밍교육은 어떠한 교수 접근(interventions)을 활용하여 이루어지고 있는가?

- 국내의 경우 컴퓨팅 사고력에 기본이 되는 개념 교육을 기반으로 하되 이를 수행할 수 있는 역량 즉 **문제해결력 증진**을 지향하고 실제 실천적 노력이 이루어지고 있는 추세
- 컴퓨팅 사고력을 토대로 문제를 해결할 때 **협력적 상호작용**이 요구되므로 이를 교수 접근에 도입하고 있음

- **디자인 사고(design thinking) 접근** 활용하여 학생들이 문제를 스스로 정의하는 기회를 가질 수 있도록 하는 것이 필요
- **자동화된 교수 접근**에 대한 연구 필요

결론 및 논의

2 국내 연구들은 프로그래밍교육을 통한 컴퓨팅 사고력 개발 효과로 무엇을 평가하고 있는가?

- 수행 및 관점 측면에서 다소 연구가 부족
- 국내 프로그래밍교육의 방향이 컴퓨팅 개념 습득을 넘어 컴퓨팅 수행 측면이 강조되고 있다는 고무적인 현상
- 향후 소프트웨어 프로그래밍교육 연구 방법론 측면에서 보다 체계적인 접근이 마련되어야 함
- 설계기반 연구방법론(Design-based Research)을 활용한 연구가 수행될 필요가 있음

- 컴퓨팅 사고력이 가진 구성요소가 다면성을 가지고 있으므로 다양한 방법들이 상호보완적으로 활용되고 이들 방법들간의 수렴타당도를 보고하는 연구들이 공유될 필요가 있음
- 자동화된 검사에 대한 검토, 도입 및 자체 개발 필요



Q & A

감사합니다

