



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

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

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

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

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

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

[Disclaimer](#) 

박사학위논문

창의성 신장을 위한 디자인씽킹 기반
소프트웨어 교육 프로그램

제주대학교 대학원

과학교육학부 컴퓨터교육전공

서영호

2022년 2월

창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램

指導教授 金 鍾 勳

徐 榮 皓

이 論文을 教育學 博士學位 論文으로 提出함

2021年 11月

徐榮皓의 教育學 博士學位 論文을 認准함

審査委員長	김 종 우	(印)
委 員	박 찬 정	(印)
委 員	김 용 인	(印)
委 員	오 정 철	(印)
委 員	김 종 훈	(印)

濟州大學校 大學院

2021年 11月

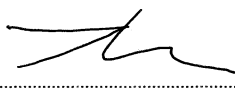
Design Thinking-based Software Education Program for Creativity Enhancement

Youngho Seo
(Supervised by professor Jonghoon Kim)

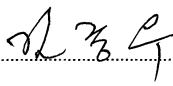
A thesis submitted in partial fulfillment of the requirement for the
degree of Doctor of Philosophy in Education

2021. 11.

This thesis has been examined and approved.

Jonghoon Kim 

Thesis director, Jonghoon Kim, Prof. Department of Computer Education

ChongWoo Kim 

Chan Jung Park 

Yong Min Kim 

Oh Jeong Cheol 

(Name and signature)

Date

Major in Computer Education
Faculty of Science Education
GRADUATE SCHOOL
JEJU NATIONAL UNIVERSITY

목 차

표 목 차	v
그림목차	vii
국문초록	ix
I. 서론	1
1. 연구의 필요성	1
2. 연구의 내용	4
3. 연구의 기대 효과	5
II. 이론적 배경	6
1. 창의성	6
1) 창의성의 정의	6
2) 창의성의 구성 요인	8
3) 창의성 검사 도구	9
2. 디자인씽킹(Design Thinking)	11
1) 디자인씽킹의 정의	11
2) 디자인씽킹의 절차	17
3) 디자인씽킹 활용 교육	20
4) 국내 디자인씽킹 선행연구	22

3. CSCL(Computer Supported Collaborative Learning, 컴퓨터 지원 협력학습) ……	24
1) CSCL의 정의 ……	24
2) CSCL에서 상호작용의 개념과 특징 ……	25
3) 디자인씽킹 기반 교육에서의 CSCL ……	29
4. 프로그램 개발 모형 ……	30
1) ADDIE 모형 ……	30
2) Rossett 요구 분석 모형 ……	32
III. 소프트웨어 교육 프로그램 설계 ……	33
1. 디자인씽킹 기반 소프트웨어 교육 프로그램을 위한 요구 분석 ……	33
1) 요구 분석 목적 결정 ……	33
2) 출처 확인 ……	33
3) 도구 선택 ……	34
4) 요구 분석 실시 ……	34
5) 요구 분석 의사 결정 ……	37
2. 디자인씽킹 기반 소프트웨어 교육 프로그램의 설계 ……	38
1) 성취 목표 명세화 ……	38
2) 프로그램 구조 설계 ……	38
3) 학습 내용 설계 ……	39
4) 교수 전략 설계 ……	40

5) 교육 프로그램 개발 원리	40
6) 교육 도구 선정	41
IV. 소프트웨어 교육 프로그램 개발 및 적용	43
1. 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램	43
1) 요구 분석	43
2) 연구 설계 및 개발	45
3) 적용	47
4) 연구 결과 분석	57
2. 애플벤티를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램	60
1) 요구 분석	60
2) 연구 설계 및 개발	62
3) 적용	64
4) 연구 결과 분석	74
3. 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램	78
1) 요구 분석	78
2) 연구 설계 및 개발	80
3) 적용	82
4) 연구 결과 분석	91
4. 디자인씽킹 기반 소프트웨어 교육 프로그램 창의성 결과 비교 분석	95

V. 결론 및 제언	99
참고문헌	101
Abstract	113
부록	116

표 목 차

<표 II-1> 창의성 관련 연구	6
<표 II-2> 창의성 관련 연구에서의 창의성 구성 요인	8
<표 II-3> 프로그래밍과 창의성 관련 연구	10
<표 II-4> 연구자별 디자인씽킹의 정의	11
<표 II-5> 디자인씽킹의 특성	16
<표 II-6> 창의성 관련 디자인씽킹 선행연구	22
<표 II-7> 소프트웨어 교육 관련 디자인씽킹 선행연구	23
<표 II-8> ADDIE 모형의 과정과 산출물	31
<표 III-1> 정보원과 추출 가능한 정보	33
<표 III-2> 초등교사 컴퓨터 관련 소지 학위	34
<표 III-3> 초등학교 학생들의 학년군별 성비	34
<표 III-4> 소프트웨어 교육 참여 경험	35
<표 III-5> 소프트웨어 교육 도구 선호도	36
<표 III-6> 소프트웨어 교육 학습 방법 선호도	36
<표 III-7> 디자인씽킹 기반 소프트웨어 교육을 통한 능력 신장	37
<표 III-8> 디자인씽킹 기반 소프트웨어 교육 성취 목표	38
<표 III-9> 디자인씽킹 기반 소프트웨어 교육 단계별 활동 내용 및 활용 도구	39
<표 III-10> 교수 전략 설계	40
<표 IV-1> 스크래치 사용 경험	43
<표 IV-2> 스크래치 학습자 디자인씽킹 교육 이해도	44
<표 IV-3> 스크래치 학습자 소프트웨어 교육 학습 방법 선호도	44
<표 IV-4> 초등 예비교사의 성별 구성	45
<표 IV-5> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제	46
<표 IV-6> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계	47
<표 IV-7> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안	51
<표 IV-8> 스크래치 활용 디자인씽킹 기반 문제해결 주제	52

<표 IV-9> 창의성 검사 정규성 검정	57
<표 IV-10> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)	58
<표 IV-11> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)	58
<표 IV-12> 앱인벤터 사용 경험	60
<표 IV-13> 앱인벤터 학습자 디자인씽킹 교육 이해도	61
<표 IV-14> 앱인벤터 학습자 소프트웨어 교육 학습 방법 선호도	61
<표 IV-15> 실험 집단 성별 구성	62
<표 IV-16> 성별에 따른 학년 구성	62
<표 IV-17> 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제	63
<표 IV-18> 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계	64
<표 IV-19> 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안	68
<표 IV-20> 앱인벤터 활용 디자인씽킹 기반 문제해결 주제	69
<표 IV-21> 창의성 검사 정규성 검정	74
<표 IV-22> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)	75
<표 IV-23> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)	75
<표 IV-24> 마이크로비트 사용 경험	78
<표 IV-25> 마이크로비트 학습자 디자인씽킹 교육 이해도	79
<표 IV-26> 마이크로비트 학습자 소프트웨어 교육 학습 방법 선호도	79
<표 IV-27> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제	81
<표 IV-28> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계	82
<표 IV-29> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안	86
<표 IV-30> 마이크로비트 활용 디자인씽킹 기반 문제해결 주제	87
<표 IV-31> 창의성 검사 정규성 검정	91
<표 IV-32> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)	92
<표 IV-33> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)	92
<표 IV-34> 디자인씽킹 기반 소프트웨어 교육 프로그램 도구별 창의성 결과 비교	95

그림 목 차

[그림 II-1] Standford d.school의 디자인씽킹 프로세스 모델	17
[그림 II-2] ADDIE 모형의 요소 및 개념화	30
[그림 II-3] Rossett 모형 절차	32
[그림 III-1] 교육 요소 간의 관계	39
[그림 IV-1] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계	48
[그림 IV-2] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계	48
[그림 IV-3] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계	49
[그림 IV-4] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계	49
[그림 IV-5] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계	50
[그림 IV-6] 스크래치 활용 디자인씽킹 공감 단계 활동 장면	53
[그림 IV-7] 스크래치 활용 디자인씽킹 문제 정의 단계 활동 장면	53
[그림 IV-8] 스크래치 활용 디자인씽킹 아이디어 발상 단계 활동 장면	54
[그림 IV-9] 스크래치 활용 디자인씽킹 시안 만들기 단계 활동 장면	54
[그림 IV-10] 스크래치 활용 디자인씽킹 검증하기 단계 활동 장면	55
[그림 IV-11] 디자인씽킹 기반 3그룹 ‘사회적 약자의 피해’ 활동 장면	55
[그림 IV-12] 디자인씽킹 기반 6그룹 ‘다문화 학생 차별 문제’ 활동 장면	56
[그림 IV-13] 디자인씽킹 기반 4그룹 ‘아프리카의 기아 문제’ 활동 장면	56
[그림 IV-14] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계	65
[그림 IV-15] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계	65
[그림 IV-16] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계	66
[그림 IV-17] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계	66
[그림 IV-18] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계	67
[그림 IV-19] 앱인벤터 활용 디자인씽킹 공감 단계 활동 장면	70
[그림 IV-20] 앱인벤터 활용 디자인씽킹 문제 정의 단계 활동 장면	70
[그림 IV-21] 앱인벤터 활용 디자인씽킹 아이디어 발상 단계 활동 장면	71
[그림 IV-22] 앱인벤터 활용 디자인씽킹 시안 만들기 단계 활동 장면	71

[그림 IV-23] 앱인벤터 활용 디자인씽킹 검증하기 단계 활동 장면	72
[그림 IV-24] 디자인씽킹 기반 1그룹 ‘게임 중독’ 활동 장면	72
[그림 IV-25] 디자인씽킹 기반 2그룹 ‘비만 줄이기’ 활동 장면	73
[그림 IV-26] 디자인씽킹 기반 3그룹 ‘쉬운 피아노 연습’ 활동 장면	73
[그림 IV-27] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계 ...	83
[그림 IV-28] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계 ...	83
[그림 IV-29] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계 ..	84
[그림 IV-30] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계 ...	84
[그림 IV-31] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계	85
[그림 IV-32] 마이크로비트 활용 디자인씽킹 공감 단계 활동 장면	88
[그림 IV-33] 마이크로비트 활용 디자인씽킹 문제 정의 단계 활동 장면	88
[그림 IV-34] 마이크로비트 활용 디자인씽킹 아이디어 발상 단계 활동 장면	89
[그림 IV-35] 마이크로비트 활용 디자인씽킹 시안 만들기 단계 활동 장면	89
[그림 IV-36] 마이크로비트 활용 디자인씽킹 검증하기 단계 활동 장면	90
[그림 IV-37] 디자인씽킹 기반 2그룹 ‘휴대폰 쉽게 찾기’ 활동 장면	90
[그림 IV-38] 디자인씽킹 기반 3그룹 ‘쓰레기통 비우기 알림’ 활동 장면	91

<국문초록>

창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램

서 영 호

제주대학교 대학원 과학교육학부 컴퓨터교육전공

지도교수 김 종 훈

본 연구의 목적은 창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하고 이를 현장에 적용하여 교육적 효과를 검증하는 데 있다.

4차 산업혁명 시대에서 과학기술을 포함한 인간의 삶의 변화 속도는 예측할 수 없을 정도로 빨라지며, 사회의 다양한 분야 간의 연계 및 융합이 점점 가속화 될 것이다. 그와 더불어 직업의 소멸, 분화, 생성의 속도는 빠르면서도 불확실한 상황으로 이어지고 있다. 이러한 급변하는 미래사회에서 학생들은 창의적 사고 능력, 지식 정보 처리 능력, 의사소통 능력, 자기 관리 능력, 공동체 능력, 심미적 감성 능력의 역량 습득이 매우 중요해졌다. 이러한 역량을 강화하기 위하여 협업 할 수 있는 형태의 수업이 필요하다.

본 연구에서는 초등학생과 초등 예비교사를 대상으로 창의성 신장을 위해 ADIE 모형에 따라 디자인씽킹 기반 소프트웨어 교육 프로그램을 설계·개발하고 이를 현장에 적용하여 교육적 효과를 검증하였다.

분석 단계에서는 창의성, 디자인씽킹 기반 소프트웨어 교육, CSCL 등에 대한

선행연구 분석과 Rossett 요구 분석 모형에 맞춰 초등학생과 교사를 대상으로 학습자 요구 분석을 진행하였다.

설계 단계에서는 성취 목표 명세화, 프로그램 구조 설계, 학습 내용 설계, 교수 전략 설계, 교육 프로그램 개발 원리, 교육 도구 선정 과정을 통해 교육 프로그램을 설계하였다.

개발 단계에서는 설계 단계의 내용을 바탕으로 3가지의 교육 도구에 따라 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하였다.

적용 단계에서는 디자인씽킹 기반 소프트웨어 교육 도구에 따라 개발된 3가지의 디자인씽킹 기반 소프트웨어 교육 프로그램을 총 3회에 걸쳐 초등 예비교사와 초등학생에게 적용하여 프로그램의 교육적 효과를 검증하였다. 디자인씽킹 과정에서 협력이 필요한 상황에서는 CSCL 도구로 패들렛, 구글 프레젠테이션을 사용하였다. 3가지의 교육 프로그램은 ‘스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’으로 구성하였다.

평가 단계에서는 창의성 검사 도구로 TTCT 검사지 도형 A, B형을 사용하였다. 평가 단계를 통해 분석된 본 연구의 ‘스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’의 학습 효과는 모두 초등학생의 창의성 신장에 효과적인 것으로 나타났다.

본 연구의 시사점은 다음과 같다.

첫째, 디자인씽킹 기반 소프트웨어 교육 프로그램은 창의성 신장에 효과적이다. 공감 단계, 문제 정의 단계, 아이디어 발상 단계, 시안 만들기 단계, 검증하기 단계로 5가지 단계에 따라 학생 활동 중심의 디자인씽킹 기반 소프트웨어 교육 문제 해결 과정을 통해 창의성이 신장되었다.

둘째, 디자인씽킹 기반 소프트웨어 교육 프로그램의 도구로 활용된 스크래치, 앱인벤터, 마이크로비트는 창의성의 하위 요소의 특징에 따라 구분하여 사용할 수 있음을 확인하였다. 스크래치는 ‘독창성’과 ‘정교성’, 앱인벤터는 ‘유창성’, ‘독창성’, ‘정교성’, ‘성급한 종결에 대한 저항’, 마이크로비트는 ‘유창성’, ‘제목의 추상

성' 신장에 효과적이었다.

셋째, 디자인씽킹 기반 소프트웨어 교육 프로그램을 적용할 수 있는 교육 교재와 교수·학습 과정안 개발을 통해 초등학생과 초등 예비교사를 대상으로 한 디자인씽킹 기반 소프트웨어 교육을 일반화할 수 있는 가능성을 제시하였다.

향후 후속 연구를 통해 본 연구에서 개발한 디자인씽킹 기반 소프트웨어 교육 프로그램을 지속적으로 개선하여 대상을 확대 적용해 간다면 창의성 신장을 위한 일반화된 또 하나의 소프트웨어 교육 프로그램이 될 것이다.

주요어: 창의성, 디자인씽킹, 소프트웨어 교육 프로그램, CSCL, ADDIE 모형

I. 서론

1. 연구의 필요성

2016년 World Economic Forum(WEF)에서 Schwab(2016)은 제 4차 산업혁명을 언급하며 인공지능, 빅데이터, 로봇 기술 등 스마트 기술의 등장으로 사회, 경제, 교육 등 사회 전반의 변화를 가져올 것이라고 하였다. 이에 맞물려 제 4차 산업혁명의 시대를 살아갈 미래 인재들의 역량에 대해서도 관심을 가지게 되었다.

이전까지 교육은 3R's 중심의 기초 기능 교육인 읽기, 쓰기, 셈하기 교육에 치중해 왔다. 지식 위주의 단편적인 학습, 학습 내용의 암기 등의 방법으로 많은 문제를 해결할 수 있었으며, '아는 것이 힘' 이라는 말처럼 사회는 지식 기반의 사회였다.

21세기 들어서면서 빠르게 변화하는 세상에 대응할 수 있는 역량 중심의 미래 인재가 필요함을 많은 연구자들이 언급하게 되었다. 먼저 Partnership for 21st Century Skills[P21](2009)에서는 3R's를 이해함과 동시에 미래 핵심 역량인 창의력(Creativity), 협업능력(Collaboration), 비판적 사고능력(Critical thinking), 의사소통능력(Communication)의 4C를 요구하고 있다.

이에 더하여 WEF는 <New Vision for Education: Unlocking the potential of technology> 보고서에서도 핵심역량으로 비판적 사고/문제해결능력, 창의력, 의사소통능력, 협업능력을 요구하고 있다(World Economic Forum, 2015).

우리나라에서도 2015 개정 교육과정에서 4C와 연관된 내용을 주요 역량으로 다루고 있으며 정보과 교육과정의 교과 역량에서는 창의·융합능력, 협력적 컴퓨팅 사고력, 디지털 의사소통 능력, 공유와 협업 능력 등을 반영하였다(교육부, 2015).

이상에서 연구자들은 공통적으로 창의력과 협업능력을 강조하고 있는 것을 알 수 있다.

「미래의 속도」의 저자들은 4차 산업혁명이 가져올 사회 변화 속도가 산업혁명(18세기 후반의 1차 산업혁명)의 10배, 규모는 100배, 효과는 3,000배가 될 것이라고 주장하고 있다(Dobbs, Manyika, & Woetzel, 2016). 이런 급격한 변화에

적응할 수 있도록 학생들에게 생각의 유연성과 근력을 키워줘야 한다. 한국의 교육 시스템은 산업화 시대에 남을 따라잡는 추격형 인재(Fast follower) 교육에 최적화되어 있다. 4차 산업혁명 시대에는 인간보다 뛰어난 사고능력을 가진 인공지능이 나타나서 연산이나 암기력, 눈치 보기 실력은 더 이상 경쟁력이 될 수 없을 것이다. 인공지능 바둑 프로그램인 알파고만 하더라도 인간이 평생 공부를 해도 다 학습하지 못할 분량인 프로그램인 프로기사 기보 16만 개를 심층학습(Deep learning)이라는 기술을 바탕으로 단 5주 만에 독파했다. 암기력이 뛰어난 인재는 곧 다가올 미래에 기계와의 대결에서 살아남기 어렵다. 왜냐하면, 기계의 기억장치가 인간의 것을 압도하기 때문이다.

4차 산업혁명 시대에서 과학기술을 포함한 인간의 삶의 변화 속도는 예측할 수 없을 정도로 빨라지며, 사회의 다양한 분야 간의 연계 및 융합이 점점 가속화될 것이다. 그와 더불어 직업의 소멸, 분화, 생성의 속도는 빠르면서도 불확실한 상황으로 이어지고 있다. 이러한 급변하는 미래사회에서 학생들은 창의적 사고 능력, 지식 정보 처리 능력, 의사소통 능력, 자기 관리 능력, 공동체 능력, 심미적 감성 능력의 역량 습득이 매우 중요해졌다. 이러한 역량을 강화하기 위하여 협업할 수 있는 형태의 수업이 필요하다.

과거에는 창의성이 소수의 특별한 사람만이 가지는 선천인 특성으로 인식되었으나, 창의성이란 모든 인간이 가지는 보편 특성이며 교육을 통해 개발이 가능하다는 연구 결과가 보고되면서(Zabelina & Robinson, 2010) 창의성 교육에 대한 관심이 높아지고 있다. 또한 특수한 분야만이 아니라 다양한 영역에서 창의 문제 해결 능력을 요구하게 되면서 창의성 교육의 중요성은 더욱 강조되고 있다. 창의성 함양에 있어서 실생활의 문제를 직접 찾고 문제에 대한 다양한 아이디어와 창의적 해결 방법을 도출하기 위하여 디자인씽킹 기반 프로그램을 제안한다.

디자인은 우리가 상상한 것을 현실로 구현해 줄 수 있는 큰 그림이며 그것을 가능하게 해주는 행위이다. 현재 상태의 수정이나 변형이 아닌 이상향(Desired state)에 대한 창조의 과정이다(홍정순, 2020). 우리는 과거와 달리 미래를 예측하기 어려운 시대에 살고 있으며 이러한 불확실한 시대에는 급변하는 환경에 빠르게 대처하는게 중요하며 이러한 시기에도 실리콘 벨리의 ‘에어비앤비(Airbnb)’와 ‘인스타그램(Instagram)’과 같은 기업은 디자인을 통한 창의와 혁신으로 큰 성공

을 거두고 있다(배성환, 2017). 디자인씽킹은 인간에 대한 관찰, 공감을 통해 소비자를 이해하고 다양한 해결 방법을 찾는 확산적 사고와 주어진 상황에 가장 합리적인 방법을 찾는 수렴적 사고의 반복 과정으로 혁신적인 결과를 내는 창의적 문제해결 방법이다(이재용, 2012).

또한 팀원 간의 협업, 다른 가치 속에서 공통된 가치를 발견하게 되는 적극적이고 급진적인 협력 관계는 디자인씽킹의 또 핵심이다(Adams et al., 2011; Brown, 2008; Noel & Liu, 2017). 하지만 협력학습이 늘 성공적인 결과를 보장하는 것은 아니다(Dillenbourg & Hong, 2008; Koller, Fisher, & Hesse, 2006). 이에 디자인씽킹의 성공적인 협력학습을 위하여 Computer Supported Collaborative Learning(CSCL, 컴퓨터 지원 협력학습)을 적용하려고 한다. CSCL은 컴퓨터 또는 인터넷을 통한 사회적 상호작용을 통해 학습이 이루어지는 교육학적 접근 방식이다. 이러한 종류의 학습은 기술을 주요 의사소통 수단 또는 공통 자원으로 사용하여 참가자 간의 지식 공유 및 구성이 특징이다(Stahl, Koschmann, & Suthers, 2006).

이에 본 연구는 창의성 신장을 위해 CSCL을 적용한 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하고 적용해 봄으로써 소프트웨어 교육에 주는 시사점을 밝히고자 한다.

2. 연구의 내용

본 연구의 내용을 요약하여 제시하면 다음과 같다.

첫째, 창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하기 위해 창의성, 디자인씽킹 기반 소프트웨어 교육에 관한 선행연구를 검토하고 그 특징을 비교·분석하여 제시한다.

둘째, 창의성을 신장시키기 위한 디자인씽킹 기반 소프트웨어 교육 프로그램을 설계·개발한다.

셋째, 개발한 교육 프로그램을 학습자에게 투입하고 창의성을 관련 검사 도구로 검사하여 본 연구에서 개발한 디자인씽킹 기반 소프트웨어 교육 프로그램의 투입 효과를 검증한다.

ISD(Instructional System Development)모형은 기본적으로 ADDIE 모형을 적용하였다. 또한, 분석은 실제 요구 분석 실행자들이 적용하기 쉬운 안내를 제공하는 Rosset의 요구 분석 모형을 사용하였다(이재무, 2014). ADDIE 모형 단계에 따라 연구 내용 및 연구방법을 구체적으로 살펴보면 다음과 같다.

- 분석(Analysis)
 - 창의성, 디자인씽킹, CSCL에 대한 연구(문헌 연구)
 - 선행연구 분석(사례 연구)
 - 학습자의 요구 분석(조사 연구)
- 설계(Design)
 - 성취 목표 명세화(조사 연구/문헌 연구)
 - 프로그램 구조 설계(조사 연구/문헌 연구)
 - 학습 내용 설계(조사 연구/문헌 연구)
 - 교수 전략 설계(조사 연구/문헌 연구)
 - 교육 프로그램 개발 원리(조사 연구/문헌 연구)
 - 교육 도구 선정(조사 연구/문헌 연구)
- 개발(Development)

- 디자인씽킹 기반 소프트웨어 교육 프로그램 개발(개발 연구)
- 실행(Implementation)
 - 디자인씽킹 기반 소프트웨어 교육 프로그램 적용(실험 연구)
- 평가(Evaluation)
 - 교육 프로그램 적용 결과 분석(실험 연구/조사 연구)

3. 연구의 기대 효과

본 연구의 기대 효과는 다음과 같다.

첫째, 공감, 문제 정의, 아이디어 발상, 시안 만들기, 검증하기 단계를 거치는 디자인씽킹 기반 소프트웨어 교육 프로그램을 통해 창의성이 신장될 것이다.

둘째, 실생활 문제 상황을 직접 찾고 스크래치, 앱인벤터, 마이크로비트를 활용한 소프트웨어 교육 경험을 통해 능동적으로 문제 해결하는 과정을 체험할 수 있을 것이다.

셋째, 디자인씽킹 기반 소프트웨어 교육 프로그램은 초등학생 및 초등 예비교사를 대상으로 개발 및 적용되어 실제 교육 현장에 적용할 수 있는 소프트웨어 교육 방안으로 활용될 수 있을 것이다.

II. 이론적 배경

1. 창의성

‘창의성’이라는 용어는 다양한 의미를 가지며 처음 심리학 영역에서 논쟁적으로 토론이 이루어졌다. 창의성에 대한 연구는 창의성에 대한 정의, 평가 및 증진에 관해 다양한 관점에 초점을 맞추고 있다. 또한, 창의적 인간, 창의적 과정, 환경 요소의 영향, 창의적 생산물이 무엇인가에 대한 탐구가 관련 연구에서 계속되고 있다 (Runco, 2001). 본 연구에서도 창의성에 대한 정의가 필요하며 창의성의 실체를 알기 위해 그 구성 요인들을 파악할 필요가 있다.

1) 창의성의 정의

창의성에 대한 정의는 학자마다 조금씩 다르다. 창의성과 관련된 기존의 연구들을 <표 II-1>을 통해 창의성에 대한 정의를 살펴보았다.

<표 II-1> 창의성 관련 연구(김정아, 2021)

연구자 (발행연도)	연구 내용	
Torrance (1966)	제목	Torrance tests of creativity thinking
	정의	창의성이란 문제, 지식의 부족, 부족한 요소들, 부조화 등에 대한 민감성, 문제와 장애를 규명하는 것, 문제 해결책을 찾는 것, 문제나 해결되어야 할 결점에 대한 추측, 가설을 형성하고 이러한 가설을 검증하고 수정, 재검토하여 결과를 얻어내는 것이다.

연구자 (발행연도)	연구 내용	
Guilford (1967)	제목	The nature of human intelligence
	정의	창의성이란 인간의 지적, 정의적 요인을 모두 포함하고 있으며, 지적요인은 지능 검사로 측정되기 어려우나 지능의 한 중요한 측면이면서 인간의 보편적인 잠재력이고, 창의적 요인은 창의적 행동을 발휘하는 개인의 인성적, 기질적 특성이며, 학교 교육에서 훈련을 통해 개발할 수 있는 것이라고 하면서 인간의 지적 능력 중 창의성의 분명한 지침이 되는 사고 유형을 확산적 사고(divergent thinking)라고 하였으며, 이것을 측정함으로써 창의성을 측정할 수 있다고 하였다.
Runco (1989)	제목	The creativity of children's art
	정의	창의성은 문제의 정의 또는 발견, 그것의 해결책으로 묘사된다. 창의성이란 이 해결책이 다른 가능한 해결책과 비교하였을 때 확실히 확산적이거나 독특해야 한다.
김영채 (1999)	제목	창의적 문제 해결: 창의력의 이론 개발과 수업
	정의	창의성을 협의, 광의, 과정으로서의 창의성으로 나누어 정의하였다. 협의의 창의성은 확산적 사고로 어떤 문제에 대한 반응의 수가 많고, 다양하고, 독특한 것일수록 창의적으로 보았다. 광의의 창의성은 새롭고 유용한 어떤 것을 생산해 내는 행동 또는 정신 과정을 창의성이라고 하였다. 과정으로서의 창의성은 기존의 정보들을 특정한 요구 조건에 맞거나 유용하도록 새롭게 조합시킨 것이라고 보았다.
김혜숙 (1999)	제목	창의성 진단 측정 도구의 개발 및 타당화
	정의	창의성이란 새롭고 가치 있는 유용한 것을 만들어 내는 능력(힘)으로서 이는 개인의 정의적 성향과 인지적 능력, 환경(상황) 및 과제와의 상호작용을 통해 결정되는 것이다.
Csikszent- mihalyi (2003)	제목	창의성의 즐거움
	정의	창의성은 특별한 사람들의 정신적 활동이 아니라 사람의 사고와 사회·문화적 맥락의 상호작용에서 나오는 새롭고 가치가 있는 아이디어나 행동이다.

Guilford는 창의성이란 개인에 따라 개인차가 있을 뿐 모든 사람들이 가지고 있는 능력이라고 주장하였다. 새로운 시각에서 창의성 연구에 관심을 가질 것을 촉구한 이후로 창의성에 관한 연구는 심리 측정적 접근을 통하여 구체적이고 과학적인 양상을 띄게 되었다. 그는 유창성, 융통성, 독창성을 기본 구성 요인으로

하며 정답이나 오답이 없고, 상상력을 발휘하여 주어진 문제에 대해 다양하고 보
다 많은 해결책을 산출해 내도록 하는 확산적 사고를 창의성의 핵심 요인으로
보았다(김윤미, 2013).

본 연구에서는 창의성이란 어떤 문제에 대한 반응이 다양하고 독특한 아이디
어를 탐색하고 고안할 수 있는 확산적 사고로 정의한다. 확산적 사고란 자발적,
독자적, 유동적 그리고 주도적인 것으로 이전에 가지고 있던 생각을 정교화하고
내재된 뜻을 끌어내어 새로운 자료로 일반화시키도록 하는 것이다. 주어진 정보
로부터 다른 정보를 생성하는 것으로 질적으로 우수하고 양적으로 많은 반응을
말한다.

2) 창의성의 구성 요인

창의성을 분석적으로 접근하기 위해, 이러한 창의성의 구성 요인들에 대한 관
련 연구를 조사하여 인지적 영역에서의 각 학자마다 주장하는 창의성의 구성 요
인에 대해 <표 II-2>에 정리하였다(김병수, 2014).

<표 II-2> 창의성 관련 연구에서의 창의성 구성 요인(김병수, 2014)

관련 연구 구성 요인	Guilford (1956)	Torrance (1974)	Gold (1981)	Feldhusen (1983)	허경철 (1991)	이영덕, 정원식 (1969)	전경원 (2000) K-CCTYC
유창성	○	○	○	○	○	○	○
융통성	○	○	○	○	○	○	○
독창성	○	○	○	○	○	○	○
정교성	○	○		○	○		
민감성		○			○		
재정의 및 재구성력		○					
조직성						○	
지각적 개방성						○	
성격적 요인						○	
상상력							○

관련 연구의 조사에서 볼 수 있듯이 유창성, 융통성, 독창성, 정교성은 창의성의 구성 요인으로 대부분 공통적으로 인정하고 있는 구성 요인이다.

이미정(2011)의 연구에 따르면 유창성, 융통성, 독창성은 다음과 같은 개념이라고 할 수 있다. 유창성이란 특정한 문제 상황에서 가능한 많은 양의 아이디어를 산출해 내는 능력이다. 흔히 정확하고 훌륭한 하나의 답을 얻기 위해 오랫동안 머리를 짜내며 고민한다. 이는 가장 훌륭한 아이디어를 얻으려는 강박관념 때문에 생기는 현상이나 유창성이란 질보다는 양이 중요한 개념이다.

융통성은 고정적인 사고방식이나 시각 자체를 전환 시켜 다양한 아이디어나 문제 해결 방법을 산출하는 능력이다. 즉 어떤 문제를 해결하거나 아이디어를 떠올릴 때, 한 가지 방법만을 고집하지 않고 다양한 측면에서 여러 가지 방법으로 접근하려고 하는 능력이다.

독창성이란 새롭고 참신함의 의미이며, 창의성의 가장 핵심적인 요소이다. 흔히 사람들이 얘기하는 창의성이 바로 독창성을 말하는 것이기도 하다. 독창성은 기존의 사고에서 벗어나 독특하면서도 참신한 아이디어를 생성하는 능력으로 창의적 사고의 궁극적인 목표라고도 할 수 있다.

본 연구에서는 창의성 검사 도구로 TTCT 도형 검사를 사용하였으며, 김영채(2010)의 연구에서 나타나 있는 ‘유창성’, ‘독창성’, ‘제목의 추상성’, ‘정교성’, ‘성급한 종결에 대한 저항’으로 다섯 가지 창의성 구성 요소로 나누어 살펴보았다.

3) 창의성 검사 도구

2001년에서부터 현재까지 국내 창의성 관련 연구에서 창의성 검사 도구로 TTCT(Torrance Tests of Creative Thinking)가 가장 많이 사용되었다. 이러한 통계 결과는 컴퓨터 과학 분야에서도 다르지 않다. <표 II-3>은 최근 컴퓨터 과학 분야에서 창의성과 관련된 논문들에서 사용되는 창의성 측정 검사 도구를 정리한 것이다. 소프트웨어 교육 연구 분야에서도 TTCT 검사가 주로 활용되고 있으며 대부분이 TTCT 도형 검사이다.

<표 II-3> 프로그래밍과 창의성 관련 연구

연구자	연구명(학위논문)	평가 도구
김종진 (2011)	EPL을 이용한 창의성 증진 교육 프로그램 개발 및 적용에 관한 연구-로고와 스크래치를 중심으로	TTCT (도형) 검사
김용민 (2018)	컴퓨팅 사고력과 창의성 향상을 위한 데이터 과학 교육 프로그램	TTCT (도형) 검사
최윤석 (2019)	언플러그드 기반 프로그래밍 언어 교육이 초등학생의 창의성에 미치는 영향	초등 도형 창의성 검사 (K-FCTES)
오정철 (2020)	컴퓨팅 사고력과 창의성 신장을 위한 퍼즐 기반 컴퓨터 교육 프로그램	TTCT (도형) 검사
김정아 (2021)	컴퓨팅 사고력과 창의성 신장을 위한 데이터 시각화 기반 컴퓨터 교육 프로그램	TTCT (도형) 검사

위 사전 연구 결과를 종합하여 본 연구에서는 창의성 검사 도구로 TTCT 검사 도형 A, B형을 사용하여 연구 결과를 검증하였다.

2. 디자인씽킹(Design Thinking)

1) 디자인씽킹의 정의

디자인씽킹이 대중들에게 알려지게 된 것은 1990년대 초반 IDEO라는 디자인 컨설팅 회사가 설립될 무렵이지만, 그 시작은 1960년대로 올라가게 된다. 1969년 Simon은 「인공과학의 이해」에서 현재 디자인씽킹의 특성과 일치된 개념을 처음으로 언급하였다. 1987년 Rowe가 호기심을 가지고 탐구하며 질문하는 디자인 프로세스를 디자인씽킹이라는 책에서 소개하였고, 1991년 디자인컨설팅회사 IDEO의 설립으로 디자인씽킹은 대중적으로 알려지기 시작하였으며, 1992년 Buchanan은 디자인씽킹을 지식의 한 전문적인 분야로 정립되는 계기를 만들게 되었고 이를 통해 디자인씽킹은 주목받기 시작하였다(신계옥, 2018). 2005년에는 SAP의 창업주 Platner 회장의 후원으로 스탠포드대학 디스쿨이 설립되면서 교육 분야에서 디자인씽킹이 확산되는 계기가 마련되었다.

디자인과 마찬가지로 디자인씽킹에 대해 하나로 수렴된 정의는 존재하지 않는다. 다만 디자인씽킹이 디자이너들이 문제를 해결해가는 방식과 그 과정 속에서의 독특한 사고 흐름에서 유래하였기에 디자이너들의 문제해결 방식과 사고방법을 공통적으로 의미한다는 것을 짐작할 수 있다(양경란 외, 2015). <표 II-4>는 여러 학자들이 디자인씽킹에 대하여 정의한 것을 정리한 것이다.

<표 II-4> 연구자별 디자인씽킹의 정의(홍정순, 장환영, 2018)

연구자	디자인씽킹의 정의
Simon (1969)	사회, 문화, 경제, 정치, 환경 등 인간생활의 모든 제반 문제를 학제적인 협동을 통하여 디자인의 통합적이고 종합적인 문제해결능력과 맞물려 해결하는 과정
Brown (2008)	직관적인 능력, 일정한 패턴을 인식할 수 있는 능력, 감성적인 의미를 전달할 뿐 아니라 기능적인 아이디어를 생각해낼 수 있는 능력. 더 나아가서는 인간의 언어나 기호가 아닌 다른 매개체를 통해 우리 자신을 표현할 수 있는 능력

연구자	디자인씽킹의 정의
Martin (2009)	생각의 가장 완벽한 방식은 분석적 사고에 기반을 둔 완벽한 숙련과 직관적 사고에 근거한 창조성이 역동적으로 상호작용하면서 균형을 이루는 것
Kolko (2010)	가추법적(Abductive) 의미짓기(Sensemaking)를 통한 디자인 종합 능력
IDEO & Riverdale (2012)	일종의 마음가짐(Mindset)이며, 우리가 변화할 수 있다고 믿는 것이고, 세상에 긍정적인 영향을 주는 새로운 솔루션을 도출하기 위한 체계적인 문제해결과정
Roth (2015)	인간의 욕구를 찾아내고, 과거엔 주로 전문적인 상품 디자이너들이 주로 사용하던 도구와 사고방식을 이용하여 새로운 해결책을 창조하는 하나의 방법. 디자인 과제(Design challenge)를 해결하는 데 효율적이며 어떤 종류의 제품이나 경험에도 적용할 수 있는 무정형의 개념

그렇다면, 디자이너들이 디자인씽킹 과정을 통해서 해결하고자 하는 문제는 무엇인지 살펴보고자 하는데, 이를 통해서 디자인씽킹의 또 다른 개념들을 파악할 수 있기 때문이다. Kangas, Seitamaa-Hakkarainen, & Hakkarainen(2013)은 디자인은 불분명한 것을 명확하게 인식하는 행위로 보고 있는데, 이렇게 아직 정의되지 않은 문제를 불명확한 문제(Wicked problem)이라고 한다. 더 구체적으로 설명하자면, 정보의 혼란으로 인한 애매모호함, 고객과 의사 결정자 사이에서 상충되는 가치들, 혼란스러운 상황이 존재하는 사회적 시스템에서 발생하는 난해한 문제들(Brown & Wyatt, 2010)이라고 볼 수 있는데, 이 불명확한 문제로 인하여 심미성을 추구하던 디자인의 개념이 문제 해결 방안을 찾기 위한 디자이너들의 방법에 중점을 두는 것으로 변천되었다(Buchanan, 1992). 그러므로 디자인씽킹에서는 불명확한 문제가 존재하는 상황을 인식하고 문제를 정의할 수 있는 능력과 이를 해결하고자 하는 도전적이고 긍정적인 태도, 복합적 사고가 요구된다.

이 불명확한 문제를 정의하고 해결하는데 있어 중요한 점은 인간 중심(Human-centered)의 관점을 지녀야 하는 것인데, 디자이너가 아닌 사용자, 불명확한 문제 상황에 있는 대상의 관점에서 생각하는 것으로 대상이 처한 상황의 특수성을 발견하고 반영하는 것을 의미한다. 즉 ‘공감(Empathy)’을 바탕으로 해

결안을 찾아야 하는 것이다(Benson & Dresdow, 2015, 2015; Carroll et al., 2010; Coleman, 2016; Dunne & Martin, 2006). Coleman(2016)은 디자인씽킹을 인간 중심적 디자인 방법으로 보고 사람들의 제품 활용 모습, 제품과의 상호작용, 제품이 삶에 미치는 영향 등을 세심하게 관찰하여 깊은 이해가 수반되어야 함을 주장한다. 이러한 사람 중심의 관점은 디자인씽킹의 핵심이자(Noel & Liu, 2017), 디자인 태도의 특성(Brown, 2008)으로 간주되고 있다.

이처럼 타인과의 소통, 타인에 대한 이해 및 공감을 전제로 하고 있는 인간 중심적 문제 해결은 디자인씽킹 개념의 사회적 측면과도 연결된다. Bucciarelli(1996)는 다양한 시각을 통한 사고와 수행이 이루어지는 디자인씽킹 과정은 때때로 갈등상황과 그것을 해결하고자 하는 협상도 포함되는 사회적 과정(Social process)으로 보고 있다. 이와 마찬가지로 Stewart(2011) 역시 디자인씽킹이 필요한 상황과 관련된 대화에 반복적으로 참여하는 상호작용에 초점을 두고 있다. 애매모호함을 분명하게 정의하고 이를 위한 해결안을 도출하기 위하여 다양한 배경, 관점 및 의견을 가진 사람들 사이에서 이루어지는 대화는 서로의 생각을 공유하면서 지속적으로 정교하게 다듬어 가장 적절한 안을 제시할 수 있게 하기 때문이다. 즉 팀원 간의 협업, 다른 가치 속에서 공통된 가치를 발견하게 되는 적극적이고 급진적인 협력 관계는 인간 중심적 접근 외에 디자인씽킹의 또 다른 핵심이다(Adams et al., 2011; Brown, 2008; Noel & Liu, 2017).

아이디어 교류, 인터뷰를 위해 타인의 이야기 듣는 행위, 서로 순서를 바꾸어가며 대화하는 행위는 친밀한 상호적 관계를 형성할 수 있게 한다. 특히 갈등을 해소하기 위한 협상은 권력에 의하여 이루어지는 것이 아니라 동등한 위치에서 이루어지는 것임을 볼 때(Campbell, 2015), 디자인씽킹의 사회적 상호작용은 타인의 상황을 깊게 이해하면서 ‘너’만의 문제가 아닌 ‘우리’의 문제로 바라보는 시각으로 전환하여 사회적 실천을 추구하게 한다. 이러한 측면에서 Renard(2014)가 제시하는 디자인씽킹 정의를 주목할 필요가 있다. 그는 디자인씽킹을 세상의 긍정적인 변화를 추구하는 행위가 이루어지게 하는 과정과 마음가짐으로 보면서 미래의 효과적인 실천가, 시민이 될 수 있는 기회를 제공한다고 정의하였다. 그의 이러한 관점은 디자인씽킹 개념에 관한 전문적인 토대를 제공하는 학자들인 Schön(1983), Lawson(1990), Rowe(1987)의 저서들을 근거로 하고 있는데, 그들

은 공통적으로 갈망하는 변화를 가져오기 위하여 세상을 보고 이해하는 방법으로 디자인씽킹을 정의하고 있다. 그들과 동일하게 Simon(1969) 역시 디자인은 '무엇인가(what is)'에 중점을 두는 과학과 다르게 '무엇이 되어야 하는가(What ought to be)'를 지향하는 것으로 현존하는 상황을 더 나은 것으로 변화시키기 위한 목적의 새로운 방안을 고안하는 과정으로 설명하고 있다(Kimbell & Street, 2009). 즉 디자인은 실제 삶이 개선되어야 할 방향을 찾고 다양한 사람들과의 협력적 관계 속에서 개인의 역할을 인식하고 참여하는 것이라고 할 수 있다(Adams et al., 2011). 이들의 정의를 종합하자면, 디자이너들, 즉 디자인씽킹 방법을 실천하게 되는 누구에게나 부여되는 사회적 역할을 강조하고 있는 것으로 볼 수 있다. 그들의 결과물이 타인과 사회의 요구에 대한 응답으로 어떻게 작용할 것인지 고려하며, 크고 작은 다양한 사회적 변화를 시도하는 실천가 태도 함양이 디자인씽킹이 지니는 사회적 측면의 개념이라고 할 수 있는 것이다.

마지막으로 몇몇 학자들은 학습 방법의 측면에서 디자인씽킹을 정의하고 있음을 볼 수 있다. Dunne & Martin(2006)은 디자인씽킹을 문제 정의, 질문 제시, 아이디어 및 해결안의 도출이 이루어지는 탐구기반학습으로 보면서, 학습자는 이를 통하여 문제를 바라보는 사고의 확장, 사용자에게 대한 이해의 심화, 타인 및 공동체를 위한 가치 인식이 가능해진다고 정의하고 있다. 이와 마찬가지로 Carroll 외(2010) 역시 학습 방법의 측면에서 학습자의 창의적 자신감을 향상시키며, 공감 형성, 수행에 대한 의욕 및 동기 유발, 아이디어 도출의 독려 등을 통해 능동적인 문제 해결력을 함양하게 되는 헨즈온(Hands-on)이라고 디자인씽킹을 정의하고 있다. 이는 디자인씽킹 과정 중 선택한 해결안을 실제적인 결과물로 구현하는 프로토타입(Prototype)이 포함되어 있는 것과 연결된다고 볼 수 있다(Benson & Dresdow, 2015; Brown, 2008; Owen, 2007). 이렇게 헨즈온 활동이 중점이 되는 학습으로서의 관점은 Bjögvinnsson, Ehn, & Hillgren(2012)은 머릿속의 아이디어가 헨즈온의 방식으로 탐구되는 즉 가시적으로 표현되어야 함을 강조한 것과 Benson & Dresdow(2015)가 디자이너로서의 태도는 기술을 활용하는데 있어 새로운 방안 도출을 가능하게 함을 언급한 것에서도 확인할 수 있다. 즉 디자인씽킹의 'Learning in making' 특성이 강조되고 있는데(Chu et al., 2015), 이 주장에 대한 구체적인 예시는 Renard(2014)의 연구에서 찾아볼 수 있다. 그는 디자인씽

킹 활용의 교육적 가치의 근거로 학습자의 구어화되지 않은 생각과 의사를 스케치, 모델링, 프로토타입 등 다양한 방법으로 가시적인 것으로 발전시킬 수 있다고 하였다. 실제로 Dowling(2012)은 핸드온 학습의 중요성을 강조하는 이론들에 따라 3D 드로잉, 스케치, 모델링 등의 도구를 활용하여 해결안을 발전시키는 수업을 설계하였는데, 이렇게 촉각을 자극하는 학습으로 인하여 학생들은 심화 학습과 학습에 대한 주인의식 함양이 가능했음을 확인할 수 있었다.

대화를 바탕으로 발생하는 상호작용, 협력은 디자인씽킹의 핵심적 요소이며, 학습으로서의 가능성을 제시해준다(Murphy & Hennessy, 2001; Kangas, Seitamaa-Hakkarainen, & Hakkarainen, 2013). Medway(1996)는 협력적인 디자인 활동을 통해서 학습자 간의 실제적이고 구체적인 인공물 제작이라는 과제의 공유는 점차 인지적 공유로 확장된다고 하였다. 이는 협력적 관계에서 상호작용을 통한 생각의 공유가 핵심이라는 학자들의 주장(Azmitia, 1988; Teasley & Roschelle, 1993; Kangas, Seitamaa-Hakkarainen, & Hakkarainen., 2013)과 연결되는데, 이 생각의 공유는 Saltmarsh 외(2009)의 협력적 지식 구성(Collaborative knowledge construction)을 이루는 과정으로 볼 수 있다. 여기서 협력은 문제의 정의, 이해, 해결안 계획, 실행, 평가를 위하여 학문적인 지식과 지역 커뮤니티의 지식이 융합되어 새로운 지식을 창출하는 것(Saltmarsh et al., 2009)으로 디자인 씽킹이 표방하는 사회적 실천의 의지가 강조되는 실천공동체를 형성한다. 이 관점은 지식을 일상적인 활동 중 이루어지는 사회적 성취(Kimbell & Street, 2009), 혹은 더 이상 개인적인 마음의 소유가 아닌 타인과의 상호작용 속에서 구성되는 것(Schatzki, 2001)으로 간주하는 것과 상통한다. 즉 디자인씽킹은 학습자 간의 상호작용, 협력을 통해 지식 공유가 가능하게 하는데 이 지식은 맥락적인 상황 속에서 우리가 추구해야 하는 공동의 가치를 지향하는 실천을 전제로 구성된다고 정리할 수 있다.

지금까지 논의된 디자인씽킹 개념에 관한 다양한 논지를 바탕으로 발견할 수 있는 디자인씽킹의 특성을 <표 II-5>와 같이 정리할 수 있다.

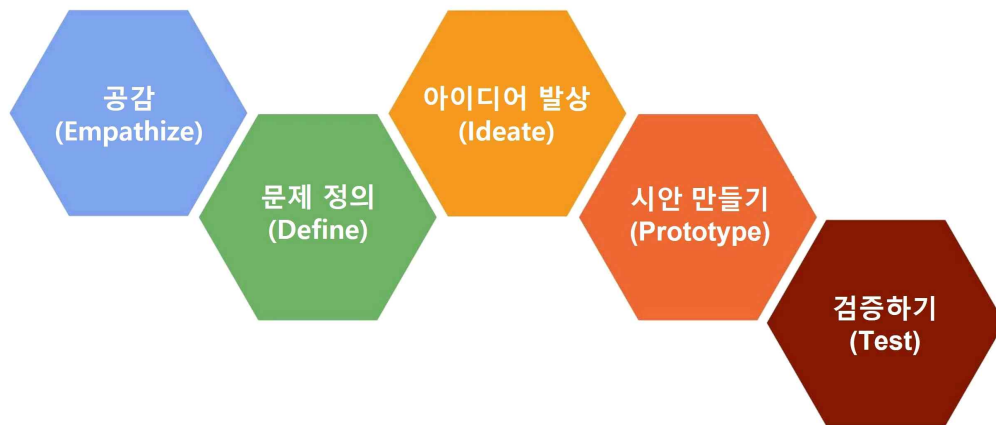
<표 II-5> 디자인씽킹의 특성(홍정순, 장환영, 2018) 수정

구분	디자인씽킹 특성	연구자
디자인적 방법	문제 및 이슈의 사회문화적 맥락성	Lawson & Dorst(2009), Suchman(1987), Stewart(2011), Suwa, Gero & Purcell(2000)
	다학문적, 융합적 문제 해결 과정	Adams et al.(2011), Brown & Wyatt(2010), Lande & Leifer(2010)
	불명확함에 대한 도전하는 태도	Brown & Wyatt(2010)
	애매모호한 문제의 정의	Brown & Wyatt(2010), Buchanan(1992), Kangas, Seitamaa-Hakkarainen, & Hakkarainen.(2013)
	인간 중심의 관점, 공감	Brown(2008), Brown & Wyatt(2010), Carroll et al.(2010), Coleman(2016), Lawson & Dorst(2009), Owen(2007)
	타인과의 의사소통, 상호작용, 협력	Adams et al.(2011), Bucciarelli(1996), Martin(2009), Noel & Liu(2017), Owen(2007), Stewart(2011)
사회적 측면	타인의 관점, 비판적 시각 형성	Bucciarelli(1996), Watson(2015)
	세상의 긍정적 변화 추구	Schön(1983), Simon(1969), Lawson(1990), Rowe(1987), Renard(2014)
	실천가로서의 면모	Renard(2014)
학습방법	제약 없는 탐구 활동	Carroll et al.(2010), Dunne & Martin(2006)
	시각화, 핸즈온 활동	Benson & Dresdow(2015), Carroll et al.(2010), Cross(2017), Hall & Nemirowsky(2011), Owen(2007)
	협력적 지식 구성	Azmitia(1998), Murphy & Hennessy(2001), Kangas, Seitamaa-Hakkarainen, & Hakkarainen(2013), Kimbell & Street(2009), Medway(1996), Saltmarsh et al.(2009), Schatzki(2001), Teasley & Roschelle(1993)
	학문적인 역량 함양	Carroll et al.(2010)
	실패의 극복 및 도전	Brown & Wyatt(2010), Martin(2009)
	다양한 인지적 사고 능력 함양	Cross(2006), Kangas, Seitamaa-Hakkarainen, & Hakkarainen(2013), Kolodner et al.(2009), Scheer et al.(2012)

따라서 본 연구에서 디자인씽킹은 불명확한 문제들을 소프트웨어 도구를 활용하여 해결하는 방법으로 타인과의 상호작용과 협력을 통해 아이디어 및 해결안을 구체화하는 과정이다. 즉 사회적 상황 속에서 야기되는 문제들을 소프트웨어 도구를 활용하여 인간 중심의 관점에서 문제를 해결하는 방법으로 이 과정 가운데 타인과의 친밀한 상호작용 기반의 협력과 지식의 공유와 헨즈온 활동을 통한 아이디어의 시각화와 다양한 사고력 함양의 깊은 학습의 효과를 얻을 수 있으며, 타인에 대한 이해를 바탕으로 세상의 긍정적 변화의 실천을 추구하는 총체적 활동이다.

2) 디자인씽킹의 절차

디자인씽킹에 대한 정의에서도 보여지듯이 디자인씽킹은 일종의 프로세스이기 때문에, 그것을 가르치고 사용하는 대학이나 기업 등에 따라 사용 모델이 조금씩 다를 수 있고 때에 따라서는 상황에 맞게 수정해서 사용하기도 한다(송동주, 박재호, 강상희, 2016). 본 연구에서는 디자인씽킹과 관련하여 가장 널리 사용되고 있는 Stanford d.school의 디자인씽킹 프로세스에 대해 살펴보고자 한다.



[그림 II - 1] Stanford d.school의 디자인씽킹 프로세스 모델(Stanford d.school, 2010)

Stanford d.school의 모델은 디자인씽킹 프로세스 모델 중 세계에서 가장 폭넓게 사용되는 것으로, 공감(Empathize), 문제 정의(Define), 아이디어 발상

(Ideate), 시안 만들기(Prototype), 검증하기(Test)의 다섯 단계로 구성되어 있다. 디스쿨에서 발행된 ‘An introduction to designthinking process guide’의 내용을 바탕으로 각각의 단계에 대하여 살펴보면 다음과 같다(Standford d.school, 2010).

(1) 공감 단계

Standford d.school의 디자인씽킹 프로세스는 ‘공감’이라는 단계를 특별히 하나의 단계로 따로 구성하였다는 점에서 ‘공감’을 보다 강조한 모델이라 할 수 있다. 공감은 인간중심 디자인 과정의 핵심으로, 공감 단계에서는 해결할 문제의 맥락 내에서 사람들을 이해하기 위한 작업이 이뤄진다. 즉 그들(디자인 문제의 대상이 되는 사람들)이 일하는 방식과 이유, 신체적 정신적으로 필요로 하는 것, 세상에 대한 그들의 생각과 그들에게 의미 있는 것을 이해하도록 노력하는 것이 이 단계에서 이루어진다.

디자인씽커(Design thinker)가 해결하려고 하는 문제는 대부분 자신의 것이 아닌 ‘그들’의 것이므로, 그들을 위해 디자인하기 위해서는 그들을 진정으로 이해하고 파악하기 위한 노력이 필요한데, 그 이유는 우리는 깨닫지도 못한 사이에 입력된 정보를 자동으로 필터링하기 때문이다. 따라서 잘 공감하기 위해서는 먼저 나 자신의 생각을 유보하고 새로운 시각으로 사물 또는 사람을 바라보아야 한다. 더불어 일종의 공감을 위한 전략을 구축하고 부단히 노력해야 한다. Standford d.school에서는 공감을 하기 위한 방법으로 특히 관찰(Observe), 대화 또는 인터뷰(Engage), 보고 듣기(Watch and listen)의 방법을 제안한다.

(2) 문제 정의 단계

“올바른 문제 설정이 올바른 솔루션을 만들어내는 유일한 방법이다(Standford d.school, 2010)”이라는 말은 문제가 제대로 정의되지 않았을 때, 즉 사용자(고객)가 진짜로 원하는 것을 발견하지 못했을 때, 아이디어 발상부터의 나머지 단계는 해결하려는 문제의 해결에 큰 도움을 줄 수 없다는 것을 의미한다. 문제 정의 단계에서 가장 중요한 것은 문제에 대한 초점과 명확성을 가져오는 것으로, 의미 있고 실행 가능한 문제 진술(Problem statement)이 목표가 된다. 공감 단계에서 수집한 자료를 통해 그들을 이해하고 진실로 원하는 것을 파악해가는 공감의 과

정 속에 ‘아하’하는 통찰의 순간을 경험하며 이에 착안하여 문제를 새롭게 발견하며 그 범위를 좁히는 과정을 하게 된다.

(3) 아이디어 발상 단계

아이디어 발상 단계는 아이디어를 만들어 내는데 집중하는 단계이다. 이 단계에서는 좋은 아이디어를 내는 것보다 더 넓은 범위의, 더 많은 아이디어를 내는데 중점을 두어야 한다. 아이디어는 곧 가능성을 의미하므로, 전 단계에서 만든 문제를 해결하기 위해 질이 높은 아이디어뿐만 아니라 조금은 엉뚱해 보이거나 실현 불가능해 보이는 아이디어도 환영받는 분위기가 마련되어야 한다. 이렇게 모아진 방대한 아이디어를 바탕으로 더 나은 것들을 선정하여 다음 단계로 넘어가게 된다.

(4) 시안 만들기 단계

시안 만들기는 아이디어를 구체화하는 단계이다. 시안은 사용자와 상호작용할 수 있는 그 어떤 것도 된다. 예를 들면, 포스트잇, 역할극 하기 활동, 스토리보드 등이 있다. Standford d.school(2010)은 다음과 같은 이유로 시안이 중요함을 설명하고 있다. 첫째, 아이디어를 발상하고 문제를 풀기 위해서, 둘째, 시안을 통해 고객과 의사소통을 하기 위해서, 셋째, 다른 사람들과 대화를 시작하기 위해서, 넷째, 빨리 값싸게 실패하기 위해서, 다섯째, 가능성을 검증하기 위해서, 마지막으로 해결안 만들기 과정을 운영하기 위해서 꼭 필요한 과정이다. 위의 여섯 가지에 비추어볼 때 시안의 완성도는 높지 않고, 빠르게 만들어 즉각적인 피드백을 받는 것이 중요하다. 송동주, 박상호와 강상희(2016)는 시안의 궁극적 목적에 대해 실패하기 위한 것이라고 이야기하며 실패를 통한 학습과도 관련지어 그 교육적 가치를 설명하였다.

(5) 검증하기 단계

검증하기 단계에서는 시안을 고객에게 제시해주고 피드백을 요청하게 된다. 고객의 피드백을 귀 기울여 듣고 직접 사용하는 것을 눈으로 관찰하며 고객에 대해 공감을 하게 된다. 이때의 공감은 문제가 정의되기 전의 공감과는 구별이 되

며, 검증하기 단계는 고객을 더 잘 이해할 기회를 가질 수도 있는 과정이다. 시안을 만들 때는 내가 알고 있는 것이 맞다는 것처럼 생각하면서 제작하고, 검증하기 단계에서는 그와 반대로 내가 알고 있는 것이 틀렸다는 것처럼 생각하고 검증하기를 진행해야 디자인씽킹 팀이 제시한 솔루션이 더 나아질 기회를 얻을 수 있다. 특히 검증하기 단계는 디자인씽킹 팀의 다음 스텝이 어디로 향해야 할지를 알려주는 역할을 한다. 만약 문제 정의가 잘못되었다면 문제 정의 단계로 돌아가고, 공감에 부족했다면 공감 단계부터 다시 시작하게 된다.

3) 디자인씽킹 활용 교육

디자인씽킹 활용 교육은 디자인씽킹을 교수·학습 방법론으로 활용한 교육이다. 디자인씽킹 활용 교육은 학습자들이 주도적으로 학습하는 과정에서 학습이 이루어지는 것으로, 학습자들은 디자이너처럼 사고하고 문제해결하는 과정에서 창의성과 문제해결능력, 그리고 해당 교과목에서 학습해야 할 지식을 습득하게 된다(Puntambekar & Kolodner, 2005; Retna, 2016). 디자인씽킹은 복잡하고 실제적인 문제를 다룬다는 점에서 문제중심학습이나 학습과 유사한데, ‘인간 중심 과정’을 강조한다는 독특성을 갖고 있다(Scheer, Noweski, & Meinel, 2012). Anderson(2012)은 ‘공감’을 강조하는 디자인씽킹은 미래사회의 혁신을 주도할 사람들이 갖추어야 할 중요한 기술이라고 강조하였다.

디자인씽킹을 활용하여 수업을 운영한 교수자들은 디자인씽킹을 학습자들이 문제를 해결하고, 창의적 아이디어를 생성하고, 인간의 요구에 부합한 해결안을 도출하게 하는 문제해결과정으로 인식하고 있다. 교수자들은 수업에서 디자인씽킹을 활용하기 전에 교사들이 먼저 디자인씽킹을 알고, 디자인씽킹 마인드를 갖는 것이 중요하며, 디자인씽킹과 교육과정을 통합할 수 있는 훈련이 선행되어야 한다고 강조하였다(Retna, 2016).

디자인씽킹은 실세계의 문제나 도전과제를 디자인 마인드로 접근하는 사고과정을 거친다(Koh et al., 2015). 디자인씽킹을 활용한 수업에 참여하는 학습자들 역시 이러한 과정을 거치는 것으로, 학습자들은 교수자의 지원과 도움을 원한다. Puntambekar & Kolodner(2005)는 학습자들이 특히 모델링과 스캐폴딩을 원한다

고 강조하면서, 학습자들이 디자인 과정을 충실히 경험하고, 이를 통해 체계적으로 지식을 습득할 수 있도록 ‘디자인 다이어리’ 개발하여 활용하였다. ‘디자인 다이어리’는 학습자들의 디자인 과정을 돕기 위해 질문과 안내를 제공하고 이를 학습자 스스로 기록하도록 한 일종의 워크시트이다. 디자인 다이어리는 4단계로 구성되어 있는데, 디자인 과제 또는 문제 이해하기, 정보 모으기, 해결안 생성하기, 평가하기 단계로 구성되어 있으며 각 단계별로 학습자들의 사고를 촉진하는 질문을 제시하였다. 학습자들은 디자인 다이어리에서 안내한 질문에 답하는 과정에서 좋은 디자인 결과물을 만드는데 중요한 내용을 기록하고, 이 과정을 통해 디자인과 학습내용을 연결할 수 있다.

학습자들의 사고를 촉진하는 질문을 제시하는 것과 함께 교수자가 수행해야 하는 또 다른 역할은 학습자들이 팀으로 활동할 수 있도록 조력하는 것이다. 디자인씽킹은 모르는 사람을 만나기도 하고, 다른 사람들과의 협업을 통해 결과물을 도출해야 한다. 따라서 학습자들의 팀활동을 안내하고 팀원들간의 관계가 원활하게 유지될 수 있도록 살펴봐야 한다(Retna, 2016).

이밖에도 디자인씽킹처럼 학습자가 주도권을 가지고 팀이 설정한 공동의 과제를 해결하는 수업에서 학습자들은 다양한 측면에서 많은 어려움을 겪게 된다(Barron et al., 1998). 구체적으로 살펴보면, 자기주도성이 낮은 학습자의 경우 디자인씽킹 수업에서처럼 학습자 중심으로 진행되는 수업에서 학습에 대한 부담감이 높은 것으로 나타났다(강명희, 홍경선, 1998). 결국 학습에 대한 높은 부담은 팀에서의 과제해결 활동을 미숙하게 수행하는데 영향을 미칠 가능성이 높다. 또한 디자인씽킹 수업에서 학습자들이 해결해야 할 과제처럼 실제성을 반영한 복잡한 문제를 해결하기 위해서는 학습자들의 높은 메타인지 수준이 필요하다. 실제로 학습자의 메타인지 수준이 낮은 경우 팀으로 과제를 해결하는데 어려움을 겪는 것으로 나타났다(박정환, 우옥희, 1999). 이밖에도 교수자 중심의 전통적인 수업에 익숙한 학습자의 경우, 디자인씽킹 수업에 적응하는데 어려움을 겪을 수도 있다. 따라서 교수자는 디자인씽킹 수업에서 학습자들이 겪을 수도 있는 어려움을 고려하고, 이를 해결하고 조력하기 위한 교수자의 역할, 즉 퍼실리테이터(Facillitator)의 역할을 수행해야 한다.

4) 국내 디자인씽킹 선행연구

창의성과 소프트웨어 교육 관련 국내 디자인씽킹 연구 흐름을 알아보기 위해, 최근 국내 학술지에 게재된 디자인씽킹 교육 프로그램을 <표 II-6>과 같이 분석해 보았다.

<표 II-6> 창의성 관련 디자인씽킹 선행연구

연구자	년도	연구 논문
변현정	2015	디자인 사고과정(Design thinking process) 경험이 대학생 창의성 계발에 미치는 영향
우영진, 윤지현, 강성주	2016	초등 과학영재의 집단 창의성 발현을 돕는 전략으로서 디자인적 사고의 가능성 탐색
서응교, 전은화, 정효정	2016	대학생 창의역량 개발을 위한 디자인씽킹 기반 강좌 개발
이도현, 윤지현, 강성주	2016	중등 과학 영재들의 집단 창의성을 돕기 위한 전략으로서 디자인적 사고 프로그램의 가능성 탐색
송영은, 이정현	2018	창의적 사고력 향상을 위한 디자인창의융합교육 모델 개발 및 검증
우영진, 윤지현, 강성주	2018	공감 중심 디자인적 사고 프로그램 적용을 통한 초등 과학 영재학생의 집단 창의성 함양 연구
이선영, 윤지현, 강성주	2018	초등학생들의 집단 창의성 개발 전략으로서 디자인씽킹 기반 문제 해결 프로그램의 가능성 탐색
김선연	2019	디자인 씽킹에 기반한 집단 창의성 사고과정 모형 개발
배예나, 박혜진	2019	디자인 사고 교육을 통한 성인학습자들의 창의적 사고 경험에 관한 연구
원연정, 배예나, 최운영, 안미리	2019	학생 대상의 디자인 사고 교육이 창의성역량에 미치는 영향
김경은	2020	유아의 창의성 증진을 위한 디자인씽킹 프로그램 개발 연구

연구자	년도	연구 논문
김선연	2020	디자인 씽킹에 기반한 집단 창의성 교육목표 분류체계 프로토타입 개발
송가영, 정은영	2020	디자인 씽킹 기반 융합 수업이 창의성에 미치는 영향
송미연, 황재원	2020	디자인사고 과정을 적용한 디자인 교육 프로그램이 유아의 창의성과 정서지능에 미치는 영향
류은진, 김민정	2021	디자인씽킹을 활용한 공학제품 설계수업에서의 창의적 문제 해결 경험 연구

창의성과 관련한 연구는 2015년부터 꾸준히 연구되고 있는 분야로 주로 창의적 사고력 신장, 집단 창의성 함양 등을 위한 디자인씽킹 기반 교육프로그램 또는 모델을 제시하고 있다.

다음으로 소프트웨어 교육과 관련된 디자인씽킹 연구는 <표 II-7>과 같다.

<표 II-7> 소프트웨어 교육 관련 디자인씽킹 선행연구

연구자	년도	연구 논문
이지선	2015	컴퓨터적 사고를 기반으로 한 컴퓨터 교육에 디자인적 사고 적용에 관한 연구
서영호, 김종훈	2017	디자인 사고를 적용한 디자인씽킹 기반교육이 초등학교 예비교사의 창의성에 미치는 효과
서응교	2017	플립러닝과 디자인씽킹에 기반을 둔 창의적사고 강화와 코딩교육을 위한 강좌개발
최형신, 김미송	2017	디자인 사고와 컴퓨팅 사고를 결합한 새로운 교사 교육 코스 설계
서영호, 김종훈	2018	CSCL기반 디자인 사고를 적용한 디자인씽킹 기반교육이 초등 학생의 창의성과 문제해결능력에 미치는 효과
신윤희, 정효정, 서응교	2019	비공학계열 학생을 위한 디자인 씽킹 기반 코딩교육 프로그램 효과분석 및 지원전략 연구
신윤희, 정효정, 송종숙	2019	디자인씽킹 기반 비전공 대학생을 위한 디자인씽킹 기반 코딩교육에서의 학습경험 분석

연구자	년도	연구 논문
박주연 외	2020	디자인 사고 과정에 기반한 초등 해커톤의 개발 및 적용
서영호, 김관민, 김종훈	2021	창의성과 컴퓨팅 사고력 향상을 위한 디자인 사고 적용 피지컬 컴퓨팅 프로그램 개발 및 적용 : 마이크로비트를 중심으로

소프트웨어 교육에 관한 연구는 2017년부터 새롭게 등장하고 있다. 이는 2015 개정교육과정에 따라 중학교에서 2018년부터 정보과목이 필수화 되고, 초등학교에서는 2019년부터 소프트웨어 기초소양교육을 운영하고 있어 이를 반영한 연구들이 2017년부터 시작된 것으로 보인다. 이 분야에는 디자인씽킹 프로세스나 방법에 기반을 둔 소프트웨어 교육이 창의성에 미치는 효과 분석 또는 디자인씽킹에 기반한 컴퓨터 교육방법론 제시 등의 연구가 있다.

3. CSCL

1) CSCL의 정의

CSCL(Computet Supported Collaborative Learning, 컴퓨터 지원 협력학습)은 컴퓨터 또는 인터넷을 통한 사회적 상호작용을 통해 학습이 이루어지는 교육학적 접근 방식이다. 이러한 종류의 학습은 기술을 주요 의사소통 수단 또는 공통 자원으로 사용하여 참가자 간의 지식 공유 및 구성이 특징이다(Stahl, Koschmann, & Suthers, 2006). 컴퓨터를 기반으로 한 협력학습 연구는 1970년대와 80년대 면대면 학습상황에서 한정된 컴퓨터를 다수의 학습자들이 동 시간에 활용하는 연구로 시작되었다. 오늘날의 CSCL 연구가 다루고 있는 웹이나 인공지능 컴퓨팅 기반의 협력학습 환경 구축은 극히 최근의 일이라고 할 수 있다.

CSCL 환경은 협력적 지식이나 지식 구축 과정을 다양한 형태로 표상화 할 수 있도록 도와주며, 다른 학습자들과 지식 및 정보를 공유, 교환, 논증, 조언, 수정하는 등의 상호작용을 지원한다. 또한 시공간의 구애를 받지 않고 언제, 어디서

나 다른 이들과 연결되어 학습할 수 있어 이를 지원하기 위한 프로그램이나 시스템 개발에 관한 연구의 필요성과 중요성이 제기되고 있다(김효숙, 김동식, 권성호, 2009; 정효정, 김동식, 2006).

특히 CSCL환경이 제공하는 상호작용은 학습자의 생각을 표출하고, 그 생각에 대한 다른 학습자의 견해를 받아들이는데 도움을 준다. 따라서 협력적 지식구성 과정을 촉진하여 개인이 독립적으로 학습할 때보다 이해의 수준을 뛰어넘을 수 있도록 한다. 또한 CSCL에서 획득한 협력적 지식은 다양한 표상화 도구에 의해 구조화 될 수 있으며 협력 과정과 결과물 역시 저장과 보관 및 재가공이 가능하다. 이는 유사한 협력 상황에서 활용할 수 있는 학습 자료를 제공할 뿐 아니라, 외재적인 지원이 학습자 인지구조에 내면화 되었을 경우 유사한 협력 상황에서 외재적 지원 없이 협력학습을 할 수 있도록 학습자를 돕는다(김효숙, 김동식, 권성호, 2009). 또한 CSCL이 제공하는 상호작용은 학습공동체 구성원 간 또는 다른 학습공동체간에 관계를 돈독히 해주고 선행 학습자의 협력 노하우나 해당 협력과제에 대한 사전지식과 경험을 공유 할 수 있게 해 주기 때문에 학습활동에서 매우 중요한 역할을 한다.

2) CSCL에서 상호작용의 개념과 특징

협력학습의 차별성과 성공요인이 학습자들 간의 상호작용에 달려있다. 협력학습의 효과는 협력학습이 일어나는 동안의 상호작용 강도(Intensity)와 질(Richness)에 따라 달라진다고 할 수 있으며 상호작용의 촉진은 협력학습의 성공과 직결된다(Dillenbourg et al., 2004).

(1) CSCL에서 상호작용의 개념

Oliver, Omari, & Herrington(1998)은 CSCL에서 학습자의 커뮤니케이션 내용에 따라 학습자간 상호작용 유형을 세 가지로 분류하였다. 학습자가 주어진 협력과제를 해결하거나 학습을 수행하기 위해 동료 간의 친밀감을 형성하거나 사교적인 커뮤니케이션이 오고가는 것을 사회적 상호작용이라 하였다. 이와 달리 학습하는데 필요한 소프트웨어의 사용 방법이나 기술적인 문제, 학습하는 순서 등 학습하는 과정에서 발생하는 문제를 해결하기 위한 커뮤니케이션 활동을 절차적

상호작용이라 하였다. 또한 내용에 대한 이해가 어렵거나 필요한 자료가 불충분할 때의 커뮤니케이션을 설명적 상호작용, 학습 내용의 평가나 분석을 통해 지식을 재구성하는 비판적 사고와 특정 상황을 접했을 때 기존의 지식에 비추어 추측하고 추론해보는 반추적 사고를 위한 커뮤니케이션을 인지적 상호작용으로 구분하였다.

또한 사회심리학자들은 집단 내에서 어떤 목표의 달성을 위해 구성원 간에 전개되는 상호작용을 '목표구조(Goal structure)'라는 개념을 통하여 구분한다(장은정, 2004). 목표구조란 집단 구성원들이 목표 달성을 추구하는 방식으로 Johnson 등(1984)은 이를 학습에 적용하여 협동구조, 경쟁구조, 개별구조와 같이 학습에서의 상호작용을 구분하였다.

협동구조는 각 개인이 소속되어 있는 집단 성원 모두에게 유익한 결과를 얻게 하고자 힘을 합하여 학습을 전개하는 목표구조이다. 경쟁 구조는 한 개인의 학습 활동이나 학습 결과가 자신에게는 유익하지만 동료에게는 경쟁적 관계 때문에 해를 끼치게 되는 목표구조이다. 개별 구조는 개인과 동료가 공동의 학습 목표를 가지고 있으나 개인의 학습 성취가 집단 성원의 학습 성취와 직접적으로 연결되지 않는 목표구조이다(장은정, 2004).

(2) CSCL에서 상호작용의 특징

CSCL에서 상호작용의 필요성이나 교육적 효과에 대해서는 여러 가지 측면에서 다양하게 논의되고 있다(Dillenbourg, Hong, & Brahm, 2009). 그 중 협력학습에서 상호작용의 가장 두드러진 특징적 역할은 학습 동기 유발이다. 상호작용은 학습자의 학습 정도에 대한 정보를 제공하거나, 적극적으로 학습하도록 학습자를 촉진하는 역할을 한다. 고차원적 상호작용은 학습에 대한 긍정적 태도와 만족감을 가져오고, 동기를 유발하여 학습 내용에 주의집중하게 함으로써 정보의 이해를 돕는다(Jonassen, 1988).

(3) CSCL에서 상호작용의 전략

기존 연구에 따르면 협력학습이 늘 성공적인 결과를 보장하는 것은 아니다(Dillenbourg & Hong, 2008). 학습자는 구체적으로 어떻게 협력해야 하는지 방법

을 모르는 경우가 많으며 이로 인한 학습자의 비적극적 참여, 낮은 토론수준, 낮은 학습전이 등은 협력학습의 대표적 실패 요인으로 언급되었다(Kollar, Fischer, & Hesse, 2006). 특히 학습자의 낮은 상호작용은 CSCL 환경에서 더욱 많이 발생한다(김효숙 등, 2009). 따라서 CSCL의 효과성을 높이기 위한 상호작용 촉진 전략은 CSCL의 가장 중요한 연구 주제가 되었다.

CSCL에서 학습자간 상호작용을 촉진해온 기존의 방법을 요약하면 다음과 같다(Dillenbourg & Hong, 2008). 첫째 커뮤니케이션 도구 설계를 통해 상호작용을 촉진한다. 이 때 커뮤니케이션 도구는 주로 버튼이나 문장을 클릭하면 사전에 정의된 명령이 화면에 출력되는 반구조화된 인터페이스를 가지고 있다. 둘째, 학습자들이 과제나 학습목표를 재구성 할 수 있도록 시각적 표상화를 지원한다. 개념도 작성 도구가 대표적인 예라 할 수 있다. C-map tool과 같은 개념도 작성 도구는 협력적 지식 구축 결과를 평면도로 표상화 할 수 있도록 돕는다. 셋째, 특정한 방법으로 팀을 편성한다. 연령, 성별, 사전지식, 성격, 인종, 관심사 등과 같은 학습자 특성요인, 팀 구성원 수, 팀 소속기간, 팀원 선발과정과 같은 팀의 구조요인, 자가 배치, 임의 표집, 능력별 배치와 같은 분류 방식요인에 따라 팀을 편성하는 것이다(이영민, 2005). 넷째, 학습자들의 상호작용의 수준을 표상화하여 학습자에게 제공한다. 상호작용의 횟수와 방향을 벡터 데이터로 표상화하는 SNA(Social Network Analysis)방법과 이를 지원하는 Netminder와 같은 프로그램이 이에 속한다. SNA 연구방법은 상호작용의 집중도, 파당의 형성여부, 상호작용의 횟수와 방향 등을 노드와 링크로 표현해준다. 김동식, 강인구(2004)는 CSCL에서 SN(Social Network)데이터를 학습자들에게 제시하면 학습자의 상호작용 과정과 활동성도에 긍정적 영향을 미친다고 밝힌 바 있다. 다섯째, 학습자에게 피드백을 제공을 통해 상호작용을 촉진한다. 피드백 관련 연구는 대상에 있어서는 교수자나 튜터의 피드백 여부와 제공 방법에 대한 효과성을 검증한 연구들이 주를 이루었으며, 방법에 있어서는 면대면 피드백, 채팅이나 게시물을 통한 피드백 등의 효과성 검증 연구가 있었다.

(4) CSCL에서 스크립트 지원과 상호작용 촉진

CSCL 연구에서는 스크립트(Script)나 모델(Model)을 스캐폴더로 제공하는 것에 대한 학습 효과가 입증되었다(Dillenbourg & Hong, 2008; Kollar, Fischer, & Hesse, 2006; 시지현, 2014; 김태우, 이영주, 2019). 그 중에서 협력 스크립트(Collaboration script) 제공은 학습자간 사회적 상호작용을 촉진하여 CSCL의 효과성을 높이기 위한 중요한 교수전략이다(Dillenbourg & Hong, 2008).

협력 스크립트는 교수자에 의해 학습자에게 외재적으로 주어지는 일종의 학습 시나리오이다(O'Donnel & Dansereau, 1992). 협력 스크립트는 학습자가 어떻게 협력해야 하는지에 대한 과제 분담, 책무분담, 협력순서, 상호작용 구조 등을 기술하고 있다(Dillenbourg et al., 2004). 특히 협력 스크립트는 협력학습 과정에 필요한 상호작용을 구조화하여 학습자에게 제시하는 특징이 있다.

Kollar, Fischer, & Hesse(2006)는 인지 심리학에서의 스크립트와 학습 심리학에서의 스크립트를 비교하면서 학습을 위한 스크립트의 공통적 속성 다섯 가지를 정리하였다. 첫째, 스크립트는 특정 목표를 수행한다. 스크립트는 개인 학습에서 과제를 이해하고 기억하는데 도움이 될 뿐 아니라 협력학습 상황에서 협력 과정에 참여하여 특정한 학습 목표에 도달하도록 학습자를 돕는다. 둘째, 스크립트는 특정 활동을 수반한다. 스크립트는 사전에 정의된 목표에 부합하는 하부 활동을 포함해야만 한다. 학습상황에서의 스크립트는 '질문하기', '설명하기', '토론하기', '요약하기' 등 학습자가 수행하기 기대하는 활동을 지시한다. 셋째, 스크립트는 시퀀스(Sequence)에 따라 구성된다. 스크립트는 해야 할 활동 외에도 언제 활동을 해야 하는지에 대한 지침을 준다. 학습 상황에서의 스크립트는 학습의 과정을 구분하는 역할을 한다. 넷째, 스크립트는 역할을 분배한다. 스크립트는 개인차원의 임무를 부여하거나 협력 상황에서의 책무를 분배한다. 스크립트의 역할 분배는 반드시 명확히 제시되지 않아도 일어난다. 예를 들어 협력학습 상황에서 타인의 의견을 비평한 학습자는 자동적으로 비평가의 역할을 수행하게 된다. 특히 역할의 분배는 누구와 함께 학습할 것인가에 대한 정보를 포함한다. 다섯째, 스크립트는 다양하게 표현된다. 스크립트는 텍스트, 그래픽 및 구두로 표현될 수 있으며 학습자 내부에 내재화 된 스크립트는 겉으로 표현되지 않을 수도 있다. 학습 상황에서 스크립트의 표현 방법이나 양식의 차이는 학습 효과의 차이를 가져온다.

3) 디자인씽킹 기반 교육에서의 CSCL

Medway(1996)는 협력적인 디자인 활동을 통해서 학습자 간의 실제적이고 구체적인 인공물 제작이라는 과제의 공유는 점차 인지적 공유로 확장된다고 하였다. 이는 협력적 관계에서 상호작용을 통한 생각의 공유가 핵심이라는 학자들의 주장(Azmitia, 1988; Teasley & Roschelle, 1993; Kangas, Seitamaa-Hakkarainen, & Hakkarainen, 2013)과 연결되는데, 이 생각의 공유는 Saltmarsh 외(2009)의 협력적 지식 구성(Collaborative knowledge construction)을 이루는 과정으로 볼 수 있다.

CSCL 환경은 협력적 지식이나 지식 구축 과정을 다양한 형태로 표상화 할 수 있도록 도와주며, 다른 학습자들과 지식 및 정보를 공유, 교환, 논증, 조언, 수정하는 등의 상호작용을 지원한다(김효숙, 김동식, 권성호, 2009; 정효정, 김동식, 2006).

본 연구에서는 디자인씽킹 기반 교육활동에서 CSCL 환경이 학습자들의 상호작용을 지원하게 된다. CSCL 환경으로 패들렛과 구글 프레젠테이션을 사용한다.

패들렛이란 하나의 작업공간에 많은 사람들이 동시에 들어와서 접착식 메모지를 붙여 놓는 작업이 가능한 웹 애플리케이션이다. 로그인하지 않아도 담벼락 생성, 공유, 출판이 모드 가능하여 회원가입과 로그인이 어려운 초등학생들에게 적합하다. 로그인을 하지 않을 경우 URL(Uniform Resource Locator)을 기억하거나 즐겨찾기(Bookmark) 해야하는 번거로움이 있을 수 있으나 QR코드(Quick Respose code)로 제시하면 쉽게 접근할 수 있다는 장점이 있다.

구글 프레젠테이션은 여러 명이 함께 문서를 작성할 수 있는 학생 참여형 문서 작성 도구이다. 모든 기기와 플랫폼에서 사용할 수 있다. 인터넷 연결과 웹 브라우저만 있으면 사용가능하다. 나머지는 구글이 처리하고 무거운 작업을 처리하는 동시에 클라우드에서 소프트웨어를 실행한다.

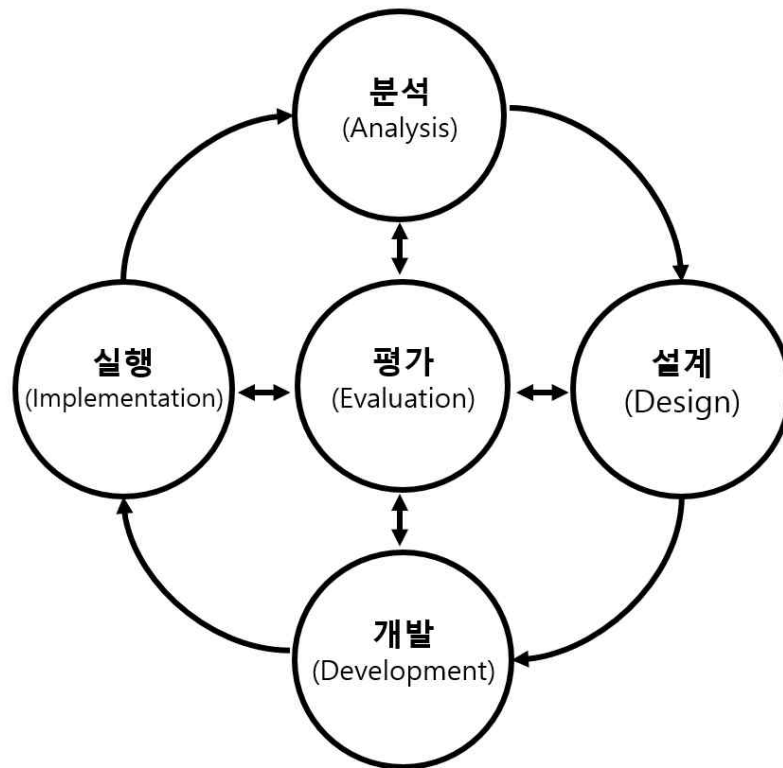
또한 프레젠테이션은 온라인 프레젠테이션 프로그램이므로 동일한 파일에서 여러 사람과 공유 및 공동 작업 할 수 있으며 수정 사항, 변경 사항 및 제안 사항을 모두 실시간으로 추적할 수 있다.

4. 프로그램 개발 모형

프로그램 개발 모형은 기본적으로 ADDIE 모형을 적용하였으며 ADDIE 모형의 분석 과정에서는 Rossett 요구 분석 모형을 적용하였다.

1) ADDIE 모형

ADDIE 모형은 과정에 초점을 맞추고 있는 교수설계의 대표 모형으로 교수설계의 과정을 크게 다섯 단계, 즉 분석, 설계, 개발, 실행, 평가로 나누고, 각 단계에서 어떠한 작업들이 어떤 방법으로 이루어져야 하는지를 보여준다(Booth, 2011).



[그림 II-2] ADDIE 모형의 요소 및 개념화

[그림 II-2]는 ADDIE 모형의 요소와 개념을 도식화한 것이다. ADDIE 모형의 각 단계는 교수설계 주기에서 한 부분씩을 차지하지만, 분석, 설계, 개발, 실행은

선형의 과정이 되는 반면, 평가는 교수설계의 전체 과정에 영향을 미치는 과정이 된다(Booth, 2011).

분석 단계는 수업 설계의 초기 단계로서 설계 상황에서 절대적인 중요성을 지닌 요소들을 분석하는 단계이다. 즉 현재의 학습자에게 어떤 문제가 있고, 어떤 점이 부족한지를 밝혀서 전체 과정을 수행하는 타당한 근거를 확보하고, 그 문제가 해결된 상태의 진술인 일반 수준의 목표를 도출하게 된다.

설계 단계는 분석 단계에서 나온 산출물을 창의적으로 종합하는 단계이며 수행목표의 명세화, 평가 도구의 설계, 프로그램의 구조화 및 계열화, 교수 전략과 매체의 선정이 포함될 뿐만 아니라 수업의 전체 모습이 그려진다.

개발 단계는 수업에 사용될 교수·학습 자료를 실제로 개발하고 제작하는 단계이며 교수 자료의 초안이 개발되고, 이 초안들은 몇 차례의 수정·보완의 과정을 거치게 된다.

실행 단계는 개발 단계에서 완성된 교수·학습 자료와 다양한 매체를 실제 현장에서 활용해 보는 단계이다.

평가 단계는 최종 산출물이 의도한 목적을 충실히 달성하였는지를 판단하는 과정이다.

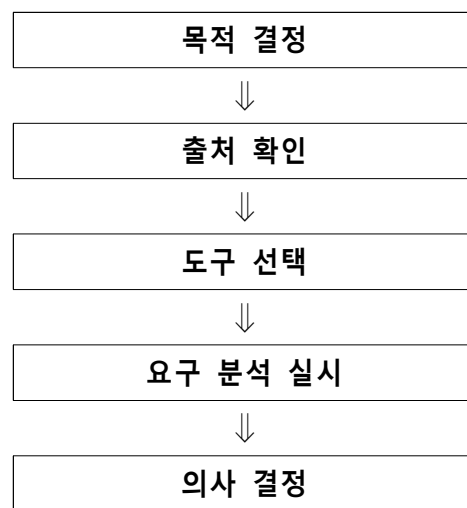
ADDIE 모형의 각 단계는 유기적으로 연관되어 있으며, 각 과정에 따른 역할, 세부 활동 및 그 산출물의 관계는 <표 II-8>과 같다(정재삼, 1998).

<표 II-8> ADDIE 모형의 과정과 산출물

과정	역할(기능)	산출물
분석	학습 내용 정의	요구, 목적, 제한점, 과제
설계	교수 방법 구체화	성취 행동 목표 교수 전략 등의 설계명세서
개발	교수 자료 제작	완성 프로그램
실행	교수 자료 실제 상황에 적용	실행 프로그램
평가	교수 자료의 효과성 효율성 결정	프로그램 평가 보고서

2) Rossett 요구 분석 모형

Rossett 요구 분석 모형(Rossett, 1987)은 기업 교육에서 널리 활용되고 있는 대표적인 교육 요구 분석 모형으로 요구 분석의 실행과정에 초점을 두으로써, 실제 요구 분석 실행자들이 적용하기 쉬운 안내를 제공하기 때문이다(이재무, 2014). Rossett 요구 분석 모형은 유발된 문제에 관한 요구 분석 목적 결정에서부터 문제 해결을 위한 의사결정까지 단계별로 제시하고 있으며 Rossett 모형의 절차는 [그림 II-3]과 같다(이재무, 2014).



[그림 II-3] Rossett 모형 절차

Rossett 모형은 유발된 문제에 관한 요구 분석 목적 결정에서부터 문제 해결을 위한 의사 결정까지 단계별로 제시하고 있다. Rossett 모형에 따른 본 연구의 요구 분석 과정(절차)은 다음과 같다.

첫째, 요구 분석 목적 선택으로, 본 연구의 요구 분석은 창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하기 위하여 학습자들의 흥미, 사전 배경지식 등을 알아보는 데 있다.

둘째, 요구 분석 출처 확인으로, 문제 유발 상황 확인 및 요구 분석 정보 획득 출처를 결정한다.

셋째, 요구 분석 도구 선택으로, 설문조사 방법 등으로 요구 분석을 실시한다.

넷째, 요구 분석 결과 분석을 통해 디자인씽킹 기반 소프트웨어 교육 프로그램 개발에 반영할 요소를 결정한다.

Ⅲ. 디자인씽킹 기반 소프트웨어 교육 프로그램 설계

1. 디자인씽킹 기반 소프트웨어 교육 프로그램을 위한 요구 분석

요구 분석은 교수·학습 목표를 추출하고, 교육 프로그램의 내용을 선정하는 데 있어서 반드시 필요한 활동이다. 교수·학습자의 요구 상황을 제대로 반영하게 되면 교수·학습자에게 유용하게 활용될 수 있기 때문에 본 연구에서는 학습자뿐만 아니라 소프트웨어 교육 전공 현직 초등교사들을 대상으로도 요구 분석을 실시하였다. 본 연구에서는 Rossett 요구 분석 모형을 사용하였다.

1) 요구 분석 목적 결정

본 연구의 요구 분석 과정을 통해 알아보려고 하는 정보를 결정하는데 본 연구에서는 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하기 위한 요구 분석을 실시한다.

2) 출처 확인

출처 확인은 정보의 출처를 확인하는 단계인데, 본 연구에서는 디자인씽킹 기반 소프트웨어 교육 프로그램의 교수·학습자들을 중심으로 요구 정보를 추출하였고, 정보 출처와 관련된 대상과 내용에 대해 <표 III-1>에 제시하였다.

<표 III-1> 정보원과 추출 가능한 정보

정보원	추출 가능한 정보
학습자	디자인씽킹 기반 소프트웨어 교육 프로그램 설계·개발에 따른 학습자의 실태 및 요구되는 정보
교수자	디자인씽킹 기반 소프트웨어 교육 프로그램 연구 설계 및 개발에 따른 학습 환경 및 교수 전략 등에 대한 지식·정보

3) 도구 선택

본 연구의 요구 분석 도구는 연구자가 직접 개발한 설문지를 사용하였고, 설문 내용은 디자인씽킹 기반 소프트웨어 교육 프로그램 연구 설계 및 개발을 위한 학습자의 실태에 대해 묻는 내용이 중심이다.

설문 형태는 폐쇄형 질문을 중심으로 학습자들의 생각을 효율적으로 알아볼 수 있도록 개방형 질문과 혼합되어 있다.

4) 요구 분석 실시

디자인씽킹 기반 소프트웨어 교육 프로그램에 대하여 다음과 같은 내용의 요구 분석을 실시하였다.

- 소프트웨어 교육 참여 경험
- 소프트웨어 교육 도구 선호도
- 소프트웨어 교육 방법 선호도
- 디자인씽킹 기반 소프트웨어 교육을 통한 능력 신장

(1) 요구 분석 대상

요구 분석을 위한 설문조사는 제주특별자치도내 초등학교 교사 27명, 초등학교 3~6학년 162명의 학생을 대상으로 실시하였다. 초등학교 교사들의 컴퓨터 관련 소지 학위는 <표 III-2>와 같다.

<표 III-2> 초등교사 컴퓨터 관련 소지 학위

학사	석사과정	석사	박사과정	박사수료	박사
7(25.9%)	10(37.1%)	5(18.5%)	2(7.4%)	1(3.7%)	2(7.4%)

초등학교 학생들의 성비의 구성은 <표 III-3>과 같다.

<표 III-3> 초등학교 학생들의 학년군별 성비

대상	남학생	여학생
초등학교 3~6학년 학생	84(51.9%)	78(48.1%)

(2) 소프트웨어 교육 참여 경험

소프트웨어 교육 참여 경험에 대하여 설문 실시 결과는 <표 III-4>에 제시하였다.

<표 III-4> 소프트웨어 교육 참여 경험

내용	구분	있다	없다
교내외에서 소프트웨어 교육에 참여해 본 경험이 있나요?	학생	123(75.9%)	39(24.1%)
	교사	27(100%)	0
('예' 라고 대답한 사람만 응답) 어떤 교육에 참여해 본 경험이 있나요?	학생 (중복 대답)	<ul style="list-style-type: none"> • 교육용 프로그래밍 언어 118(62.4%) • 앱인벤터 12(6.4%) • 피지컬 컴퓨팅 10(5.3%) • 언플러그드 활동 49(25.9%) 	
	교사 (중복 대답)	<ul style="list-style-type: none"> • 교육용 프로그래밍 언어 25(51.0%) • 앱인벤터 4(8.2%) • 피지컬 컴퓨팅 5(10.2%) • 언플러그드 활동 15(30.6%) 	

교사는 컴퓨터 관련 학위를 받은 대상이어서 소프트웨어 교육을 모두 경험하였고, 초등학생은 소프트웨어 교육 경험을 갖지 못한 경우도 있는 것으로 나타났다. 소프트웨어 교육을 경험한 응답자에 대해서는 교육용 프로그래밍 언어에 대한 경험이 많았다. 따라서, 적용하는 소프트웨어 교육 프로그램을 초보자들도 다루기 쉬운 교육용 프로그래밍 언어나 교육용 프로그래밍 언어처럼 쉽게 활용할 수 있는 도구를 활용할 필요가 있다.

(3) 소프트웨어 교육 도구 선호도

소프트웨어 교육을 위한 도구 선호도를 알아보기 위하여 설문을 실시하였으며, 설문 결과는 <표 III-5>와 같다.

<표 III-5> 소프트웨어 교육 도구 선호도

내용	구분	교육용 프로그래밍 언어	앱인벤터	피지컬 컴퓨팅	언플러그드
소프트웨어 교육에 참여할 기회가 주어진다면 어떤 교육 도구로 학습하고(가르치 고) 싶나요?	학생 (중복 대답)	102(42.1%)	58(24.0%)	42(17.4%)	40(16.5%)
	교사 (중복 대답)	25(67.6%)	5(13.5%)	6(16.2%)	1(2.7%)

초등학생과 교사 모두 교육용 프로그래밍 언어에 대한 관심도가 높게 나타났다. 소프트웨어 교육 경험에 대한 설문과 연관이 있는 것으로 파악된다. 학생들의 선호 조사에서 앱인벤터와 피지컬 컴퓨팅에 대한 선호도가 사용 경험에 비해 높게 나타났다. 따라서 디자인씽킹 기반 소프트웨어 교육 활동에서 교육용 프로그래밍 언어 뿐만아니라 앱인벤터와 피지컬 컴퓨팅 도구도 사용하는 것이 효과가 높을 것으로 생각된다.

(4) 소프트웨어 교육 방법 선호도

소프트웨어 교육 방법 선호도를 알아보기 위하여 설문을 실시하였으며, 설문 결과는 <표 III-6>과 같다.

<표 III-6> 소프트웨어 교육 학습 방법 선호도

내용	구분	강의/실습	개인 문제해결학습	협력 문제해결학습	프로젝트 학습
소프트웨어 교육에 참여할 기회가 주어진다면 어떤 방법으로 학습하고(가르치고) 싶나요?	학생 (중복 대답)	44(20.4%)	38(17.6%)	76(35.2%)	58(26.8%)
	교사 (중복 대답)	4(8.9%)	6(13.3%)	16(35.6%)	19(42.2%)

학생들과 교사들 모두 협력 문제해결학습과 프로젝트 학습을 선호하는 것으로 나타났다. 소프트웨어 교육 활동에서 문제를 협력해서 해결하거나 프로젝트 학습

처럼 결과물을 만들어내는 것을 선호하는 것으로 보아 디자인씽킹을 기반으로 한 소프트웨어 교육이 효과적일 것으로 기대된다.

(5) 디자인씽킹 기반 소프트웨어 교육을 통한 능력 신장

소프트웨어 교육을 통한 능력 신장에 관한 설문을 실시하였으며, 설문 결과는 <표 III-7>과 같다.

<표 III-7> 디자인씽킹 기반 소프트웨어 교육을 통한 능력 신장

내용	구분	창의성	컴퓨팅 사고력	문제해결력	논리적 사고력	정보활용 능력
디자인씽킹 기반 소프트웨어 교육에 참여한다면 어떤 능력 신장을 보일 것이라고 생각하나요?	학생	57(26.0%)	50(22.8%)	54(24.7%)	24(11.0%)	34(15.5%)
	교사	20(33.3%)	15(25.0%)	16(26.7%)	7(11.7%)	2(3.3%)

디자인씽킹 기반 소프트웨어 교육 프로그램을 통하여 기대되는 효과로는 창의성에 대한 응답이 가장 높게 나타났다. 따라서, 본 연구 주제가 창의성 신장에 영향을 주는지 알아보고자 한다.

5) 요구 분석 의사 결정

요구 분석 결과는 다음과 같다.

첫째, 초등학생과 교사 모두 교육용 프로그래밍 언어에 대한 관심도가 높게 나타났다. 따라서, 소프트웨어 교육 도구로 초보자들도 다루기 쉬운 교육용 프로그래밍 언어나 교육용 프로그래밍의 특징인 쉽게 사용할 수 있는 블록형 언어를 활용할 필요가 있다.

둘째, 초등학생과 교사 모두 소프트웨어 교육 방법으로 협력형 문제해결학습과 프로젝트 학습을 선호하였다. 따라서, 강의나 실습형 시간은 최대한 줄이고 문제를 협력해서 해결하거나 프로젝트 학습처럼 결과물을 만들어내는 디자인씽킹 기반의 소프트웨어 교육이 효과적일 것으로 기대된다.

셋째, 초등학생과 교사 모두 디자인씽킹 기반 소프트웨어 교육 프로그램을 통하여 창의성이 신장될 것으로 기대하고 있다. 따라서, 본 연구에서 개발된 교육 프로그램의 적용을 통해 창의성이 신장되었는지에 대한 검증이 필요하다.

2. 디자인씽킹 기반 소프트웨어 교육 프로그램의 설계

1) 성취 목표 명세화

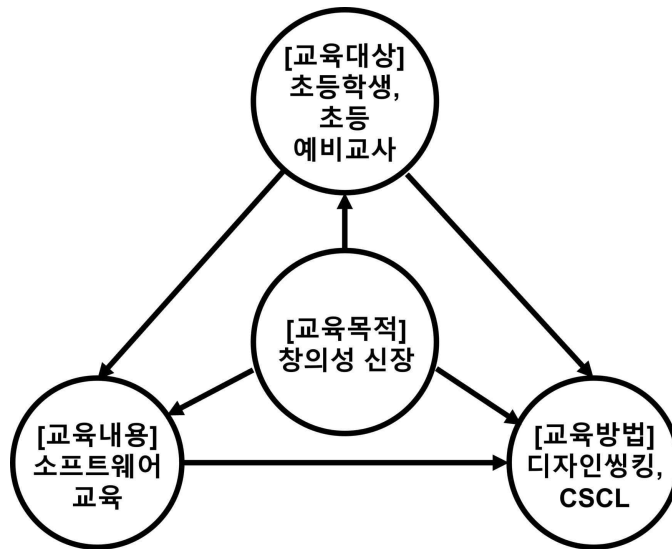
요구 분석 결과 및 디자인씽킹 기반 소프트웨어 교육 문제 해결 단계를 바탕으로 <표 III-8>과 같이 성취 목표를 선정하였다.

<표 III-8> 디자인씽킹 기반 소프트웨어 교육 성취 목표

연번	성취 목표
1	소프트웨어 교육 내용을 디자인씽킹 과정을 통해 학습할 수 있다.
2	디자인씽킹 기반 문제 해결 단계에 따라 문제를 창의적으로 해결할 수 있다.
3	디자인씽킹 과정을 통해 산출된 결과물을 공유하고 의견을 나눌 수 있다.

2) 프로그램 구조 설계

본 연구에서 개발하고자 하는 프로그램의 교육 요소와 요소 간의 관계를 도식화하면 [그림 III-1]과 같다. 본 연구의 교육 목적 달성을 위해 교수자, 학습자, 학습 내용이 어떻게 유기적으로 연결될 수 있는가에 대한 방향과 구조를 포함한 교육 프로그램 설계를 통해 교수자가 효과적이고 체계적인 교육 목표를 달성할 수 있도록 설정하였다.



[그림 III-1] 교육 요소 간의 관계

3) 학습 내용 설계

본 연구에서 디자인씽킹 기반 소프트웨어 교육 프로그램의 학습 내용과 활용 도구는 창의성 신장을 위해 해결해야 할 문제를 발견하는 단계에서부터 다양한 해결 방법을 도출하여 시안을 만들고 검증하는 일련의 과정을 경험할 수 있도록 선별적으로 구성하였다.

디자인씽킹 기반 소프트웨어 교육 문제 해결의 교육 단계별 활동 내용 및 활용 도구는 <표 III-9>와 같이 나타낼 수 있다.

<표 III-9> 디자인씽킹 기반 소프트웨어 교육 단계별 활동 내용 및 활용 도구

단계	활동 내용	활용 도구
공감	해결해야 할 문제를 발견하는 활동	CSCL 활용(패들렛)
문제 정의	명확하게 어떤 것이 문제인지 정의 내리는 활동	CSCL 활용(구글 프레젠테이션)
아이디어 발상	문제에 대한 다양한 해결 방법을 찾는 활동	CSCL 활용(구글 프레젠테이션), 스크래치, 애플벤터, 마이크로비트
시안 만들기	손을 움직여 아이디어에 등장했던 아이디어를 구체화하는 활동	스크래치, 애플벤터, 마이크로비트
검증하기	시안을 제시하고 피드백하는 활동	CSCL 활용(구글 프레젠테이션), 스크래치, 애플벤터, 마이크로비트

4) 교수 전략 개발

요구 분석 결과 및 디자인씽킹 기반 소프트웨어 교육의 대상, 소프트웨어 교육 도구의 특징, 학습 방법을 바탕으로 <표 III-10>과 같이 3가지의 교육 프로그램 교수 전략을 설계하였다. 교육의 대상으로 초등 예비교사와 초등학생 간의 차이가 있으나 초등 예비교사들은 미래의 초등학생들을 가르칠 것이고, 초등학교 교사들이 소프트웨어 교육 도구로 스크래치를 가장 선호하고 있어 초등 예비교사에게 스크래치를 적용하였다. 초등학생들은 교육 도구 사용 경험에 비해 앱인벤터와 마이크로비트에 대한 선호도가 높아 앱인벤터와 마이크로비트를 적용하였다.

스크래치는 교육용 프로그래밍 언어 중 전 세계적으로 가장 많이 사용되고 있으며, 앱인벤터는 앱을 개발하는 프로그래밍 언어 중 블록 기반의 언어로 많이 쓰이고 있다. 마이크로비트는 피지컬 컴퓨팅 교육 도구 중 기본 입력, 출력 장치가 내장되어 있고 쉽게 사용 가능하여 사용자가 많다.

<표 III-10> 교수 전략 설계

연번	대상	표본수 (명)	교육 도구	학습 방법	수업시수 (차시)
1	초등 예비교사 (○○대학교 초등교육과 1학년)	22	스크래치	강의, 실습 협력 문제해결학습, 프로젝트 학습	39
2	초등학교 4~6학년	28	앱인벤터	강의, 실습 협력 문제해결학습, 프로젝트 학습	42
3	초등학교 5~6학년	22	마이크로비트	강의, 실습 협력 문제해결학습, 프로젝트 학습	20

5) 교육 프로그램 개발 원리

본 연구의 디자인씽킹 기반 소프트웨어 교육 프로그램은 다음과 같은 원리에 따라 개발하였다.

첫째, 디자인씽킹 기반 소프트웨어 교육의 모든 프로세스를 완벽하게 지원해주는 도구를 찾을 수는 없지만, 소프트웨어 교육에서 널리 쓰이는 도구 중에서 초

등학생 및 초보자들을 대상으로 한 편리한 사용 환경과 사용법이 비교적 간단한 도구로 스크래치, 애플벤터, 마이크로비트를 활용하였다.

둘째, 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 실생활에서 찾을 수 있는 문제 중심으로 학습 주제를 선정하여 제작하였다.

셋째, 요구 분석 결과를 고려하여 학습자들의 요구에 알맞게 강의, 실습 과정은 최소화하고 협력 문제해결학습과 프로젝트 학습을 적절하게 활용할 수 있도록 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하였다.

넷째, 프로그래밍에 익숙하지 않은 학습자가 새로운 언어를 익힌다는 것은 큰 부담이 될 수 있으므로 문제 해결 과정에 중점을 두어 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하였다.

6) 교육 도구 선정

본 연구에서는 디자인씽킹 기반 소프트웨어 교육의 적절한 교수 매체로 사전 요구분석에 따라 초등학생 및 초등 예비교사들을 대상으로 편리한 사용 환경과 사용법이 간단하고 비교적 적은 프로그래밍 경험으로도 사용할 수 있는 도구들을 선정하였다. 디자인씽킹 기반으로 문제해결 과정에 집중해야 하는데 사용 환경이나 사용법이 복잡하면 기능을 익히는데 너무 많은 시간이 할애될 수 있어 스크래치, 애플벤터, 마이크로비트가 적합하다고 판단하였다.

(1) 스크래치

스크래치(Scratch)는 2007년 5월 MIT(Massachusetts Institute of Technology)의 미디어랩과 UCLA(University of California at Los Angeles)의 연구자가 인텔과 미국 과학재단의 지원 하에 공동으로 개발한 교육용 프로그램 언어이다 (Maloney, Burd, Kafai, Rusk, Silverman & Resnick, 2004).

기존의 프로그래밍 언어의 단점인 추가적인 코딩이 필요 없이 명령어가 적힌 블록들을 활용하여 프로그래밍이 가능하며, 다양한 멀티미디어를 활용할 수 있도록 만들어진 공개용 프로그램이다(송정범, 조성환 & 이태욱, 2008).

스크래치는 간단히 블록을 쌓는 것만으로도 프로그래밍이 가능하기 때문에 학생들이 복잡한 프로그래밍 언어를 외울 필요가 없으며, 블록에 쓰여 있는 단어만

으로도 직관적인 해석이 가능하여 프로그래밍을 처음 접하는 학생들도 쉽게 사용할 수 있다(안경미, 손원성 & 최윤철, 2010).

(2) 앱인벤터

구글에서 2010년 7월에 제안한 앱인벤터(App Inventor)는 전문적인 앱 개발자가 아닌 프로그래밍에 거의 경험이 없는 일반인들이 앱을 직접 개발할 수 있도록 한 웹 기반의 소프트웨어였다(Tyler, 2014). 하지만, 지금은 오픈소스화되어 미국의 MIT 대학에서 지속적인 관리를 맡고 있다. 앱인벤터는 사용자 인터페이스를 담당하는 디자이너 (Designer)와 프로그래밍을 위한 블록에디터 (Blocks Editor), 안드로이드 에뮬레이터로 구성된다(Wolber et al., 2015). 앱인벤터의 블록에디터는 교육용 프로그래밍 언어인 MIT의 스크래치(Scratch)와 유사한 사용자 인터페이스를 제공한다. 여러 개의 컴포넌트로 구성되고 각 컴포넌트는 고유 색상을 가진다. 컴포넌트 안에는 사용자가 정의한 객체 목록이 있으며, 정의된 한 객체를 선택하면 그 객체가 수행할 수 있는 행위를 별도로 관리하여 프로그래밍을 용이하게 한다.

(3) 마이크로비트

마이크로비트(Micro:bit)는 영국의 컴퓨터 교육용으로써 영국방송공사(BBC)에 의해 설계되었다. 초등학생의 소프트웨어 교육을 위해 만들어진 마이크로비트는 ‘아두이노(Arduino)’, ‘라즈베리파이(Raspberry Pi)’와 같은 오픈 소스 하드웨어로서 하드웨어의 기능과 소프트웨어의 기능을 동시에 갖춘 작은 컴퓨터와 같다.

마이크로비트는 가로 4cm, 세로 5cm의 크기로 총 25개의 빨간색 LED와 AB 버튼, 움직임 인식기와 나침반, 블루투스 등의 기능을 가지고 있다. 학생 대상의 소프트웨어 교육에 중점을 두고 개발되었기 때문에 수업 계획, 코드 샘플 등의 다양한 개발 도구를 지원한다.

파이썬, 자바스크립트, 블록 에디터 등이 지원되며 특별한 설치 없이도 인터넷이 가능한 환경이 갖춰진다면 어디서든지 코딩을 할 수 있다는 장점이 있다. 웹 브라우저를 이용하여 프로그래밍을 하기 때문에 학생 개개인이 프로그래밍을 한 결과를 다른 사람에게 공유하기 쉽다.

IV. 디자인씽킹 기반 소프트웨어 교육 프로그램 개발 및 적용

1. 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

1) 요구 분석

스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램에 대하여 다음과 같은 내용의 요구 분석을 실시하였다.

- 스크래치 사용 경험
- 디자인씽킹 교육 이해도
- 소프트웨어 교육 방법 선호도

요구 분석은 ○○대학교 초등교육과 1학년 22명의 초등 예비교사들을 대상으로 실시하였다.

<표 IV-1>과 같이 스크래치 사용 경험에 대한 조사 결과는 대다수의 학습자들이 경험이 없거나 적은 것으로 나타났다.

<표 IV-1> 스크래치 사용 경험

내용	사용한 적이 없다	사용 경험이 적다	보통이다	사용 경험이 많다.	사용 경험이 아주 많다.
스크래치 프로그램을 사용해 본 적이 있나요?	10(45.5%)	6(27.3%)	3(13.6%)	2(9.1%)	1(4.5%)

<표 IV-2>는 디자인씽킹 교육에 대한 이해도를 조사한 결과이다. 학습자들 대부분이 디자인씽킹 교육에 대해 들어본 적 없다는 의견을 제시하였다.

<표 IV-2> 스크래치 학습자 디자인씽킹 교육 이해도

내용	들어본 적 없다.	잘 모른다.	보통이다	조금 알고 있다.	아주 잘 알고 있다.
디자인씽킹 교육에 대하여 알고 있나요?	15(68.2%)	6(27.3%)	1(4.5%)	0	0

<표 IV-3>에 의하면 학습자들은 소프트웨어 교육 학습 방법에서 협력 문제해결학습과 을 선호하는 것으로 나타났다.

<표 IV-3> 스크래치 학습자 소프트웨어 교육 학습 방법 선호도

내용	강의/실습	개인 문제해결학습	협력 문제해결학습	프로젝트학습
소프트웨어 교육을 받을 때 어떤 방법으로 학습하고 싶나요? (중복)	5(11.6%)	8(18.6%)	18(41.9%)	12(27.9%)

요구 분석 결과 학습자들은 스크래치를 거의 사용해 보지 않았고, 디자인씽킹 교육에 대한 이해도가 낮은 것을 고려하여 스크래치 교육 방향을 설정하였다. 먼저 학생들이 스크래치 사용 방법을 학습할 때 디자인씽킹 교육 방법을 경험하도록 수업교재를 설계하였다. 또한, 학생들이 교육 학습 방법으로 협력하여 문제해결할 수 있도록 팀을 구성하여 교육을 진행하였다.

2) 연구 설계 및 개발

(1) 연구 가설

- ① 연구가설 : 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 있다.
- ② 귀무가설: 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 없다.

(2) 연구 대상

본 연구에서 적용한 프로그램의 교육적 효과를 확인하기 위해 ○○대학교 초등교육과 1학년 22명의 초등 예비교사들을 대상으로 연구를 실시하였다. 성별 구성은 <표 IV-4>와 같다.

<표 IV-4> 초등 예비교사의 성별 구성

	여학생	남학생	합계
실험 집단	16	6	22

(3) 연구 설계

연구 대상인 초등 예비교사들은 소프트웨어 교육에 대한 기본적인 이해가 부족하고 연구에 사용할 교육용 프로그래밍 언어인 스크래치 프로그램에 대한 기초적인 지식을 갖추지 못하였다. 따라서 18차시에 걸쳐 소프트웨어 교육에 대한 기본적인 이해 교육과 스크래치 프로그램의 기초 기능을 학습하는 시간을 가졌다. 소프트웨어에 대한 기본적인 이해 교육과 스크래치 프로그램의 기초 기능을 학습한 이후에 그룹을 조직하고 21시간에 걸쳐 디자인씽킹 협력 문제해결 과정을 진행하였다. 그룹 조직은 1그룹에 3-4명으로 구성하였으며 총 6개의 그룹을 조직하였다. 강의 시간만으로는 문제 해결시간이 부족하여 과제를 제시하였고 의견을 공유해야 하는 과제는 CSCL 도구로 패들렛, 구글 프레젠테이션을 이용하였다.

(4) 교육 내용의 선정

요구 분석 결과 및 디자인씽킹 기반 소프트웨어 교육의 효과성을 검증하기 위해 <표 IV-5>와 같이 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 내용을 선정하였다.

<표 IV-5> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제

차시	학습 주제	학습 과정
1-3	<ul style="list-style-type: none"> • 소프트웨어 교육의 이해 • 소프트웨어 교육의 필요성 • 블록 기반 EPL 알아보기 	
4-9	<ul style="list-style-type: none"> • 블록 기반 프로그램으로 문제 해결하기 • 스크래치 기본 블록 이해하기 • 디자인씽킹 이해하기 	
10-21	<ul style="list-style-type: none"> • 도형 그리기 • 창의적인 모양 그리기 • 계산 프로그램 만들기 • 기초 게임 만들기 • 간단한 게임 완성하기 • 계산 게임 만들기 • 동물 영어 퀴즈 프로그램 만들기 • 학생 추천 프로그램 만들기 	<p>- 디자인씽킹 기반 문제해결 과정으로 다음 학습 단계를 따름.</p> <div style="text-align: center;"> <pre> graph TD A[공감] --> B[문제 정의] B --> C[아이디어 발상] C --> D[시안 만들기] D --> E[검증] </pre> </div>
22-30	<ul style="list-style-type: none"> • 디자인씽킹 기반 그룹별 협력 프로젝트 과정(기본) 	<p>- 디자인씽킹 기반 그룹별 협력 프로젝트 과정 후 공유와 피드백</p>
31-39	<ul style="list-style-type: none"> • 디자인씽킹 기반 그룹별 협력 프로젝트 과정(심화) 	

교육 자료를 통하여 디자인씽킹의 개념을 익힌 후 디자인씽킹 절차에 따라 각각의 주제에 따른 문제를 해결해 봄으로써 소프트웨어 교육을 진행하였다.

(5) 검사 도구

본 연구에서 개발한 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램의 교육적 효과를 살펴보기 위해 창의성 검사 도구로 TTCT(도형) A, B형을 선정하여 투입하였다. 이 검사 도구는 그림 구성, 불완전 도형 완성, 선 더하기의 3문항으로 구성되어 있으며 각 문항당 10분씩 총 30분 동안 주어진 도형을 기초로 하여 그림을 그리고 제목을 붙이도록 하고 있다. 이 검사를 통하여 ‘유창성’, ‘독창성’, ‘제목의 추상성’, ‘정교성’, ‘성급한 종결에 대한 저항’을 측정하였다. 통계 분석 도구는 SPSS를 사용하였다.

<표 IV-6> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계

대상	사전 검사	처치	사후 검사
초등 예비교사 (○○대학교 초등교육과 1학년)	O ₁	X ₁	O ₂

X₁ : 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

O₁ : 사전 검사(창의성 검사)

⇒ 정규성 검정

O₁, O₂ : 사전·사후 검사(창의성 검사)

⇒ 정규성 검정 결과에 따른 대응표본 t 검정 또는 Wilcoxon 부호 순위 검정

3) 적용

(1) 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 학습하기

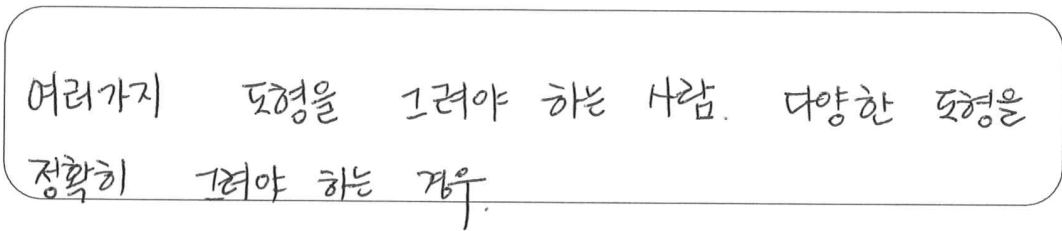
디자인씽킹 기반 소프트웨어 교육에서 각 단계를 직접 경험해 보는 것을 통해 디자인씽킹에 대한 이해를 높이면서 소프트웨어 교육을 진행하였다. 도형 그리기를 주제로 디자인씽킹 기반 소프트웨어 교육 교재를 바탕으로 학습한 내용을 살펴보고 한다.

① 공감 단계

공감 단계에서는 학습을 통해 제작할 프로그램이 어떤 문제를 갖은 사람들에게 도움이 될지 문제를 찾는 과정으로 [그림 IV-1]과 같이 활동하였다.

1. 공감 단계 [Empathize]

- 어떤 문제를 갖은 사람들에게 도움이 될지 적어보세요.



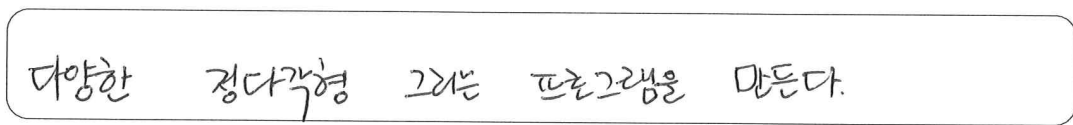
[그림 IV-1] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계

② 문제 정의 단계

문제 정의 단계에서는 공감 단계에서 생각한 문제를 명확히 하는 과정으로 [그림 IV-2]와 같이 활동하였다.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.



[그림 IV-2] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계

③ 아이디어 발상 단계

아이디어 발상 단계에서는 도형의 그리기 원리를 찾아내는 등 문제를 해결하기 위한 다양한 아이디어를 도출하는 과정으로 [그림 IV-3]과 같이 활동하였다.

3. 아이디어 발상 단계 [Ideate]

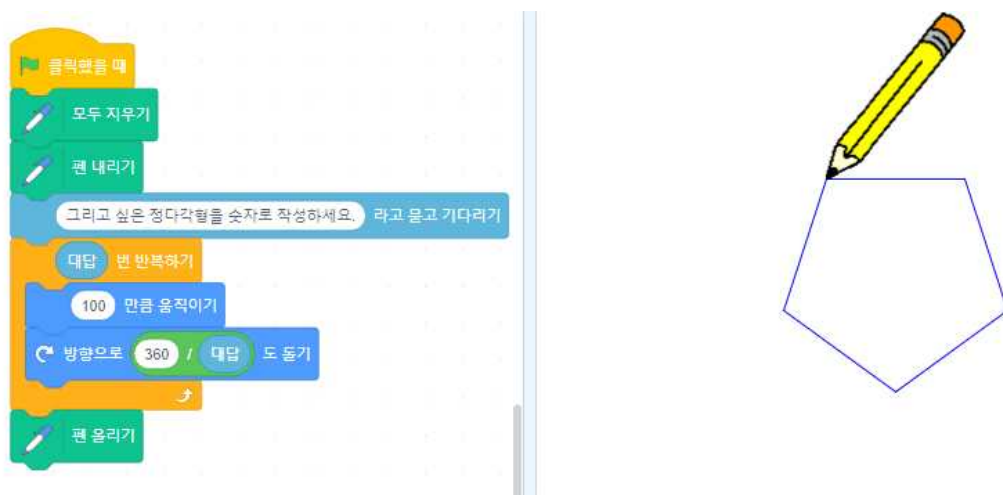
- 정삼각형, 정사각형, 정오각형... 등 그리기 원리를 아는 대로 적어보세요.

각변의 길이가 같다. 내각, 외각의 크기가 같다.
삼각형의 내각의 크기는 90° 이다.

[그림 IV-3] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계

④ 시안 만들기 단계

시안 만들기 단계에서는 직접 스크래치를 활용하여 도형을 그리는 프로그램을 작성하는 과정으로 [그림 IV-4]와 같이 활동하였다.



[그림 IV-4] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계

⑤ 검증하기 단계

제작한 프로그램의 오류 검증, 프로그램의 장점, 프로그램에 추가하면 좋을 점에 대하여 알아보는 과정으로 [그림 IV-5]와 같이 활동하였다.

5. 검증하기 단계 [Test]

1) 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

동작이 잘 이루어진다.

2) 만든 프로그램의 장점을 적어주세요.

다양한 정다각형 쉽게 그릴 수 있다.
정다각형들의 특징을 잘 파악할 수 있다.

3) 추가하면 좋을 내용을 적어주세요.

변의 길이나 그리는 위치를 정할 수 있으면 좋겠다.

[그림 IV-5] 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계

(2) 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안

앞서 소개했던 디자인씽킹 기반 소프트웨어 교육을 하기 전에 일반적인 수업에서 가장 많이 사용되는 ‘도입-전개-정리’의 구조를 사용하여 교사용 교수·학습 과정안을 구성하였다. 학생들이 실생활의 문제에 대해 스크래치를 활용하여 디자인씽킹 과정을 경험할 수 있도록 과정안을 구성하였다.

<표 IV-7> 스크래치 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안

주제	도형 그리기	차시	10차시
대상	초등 예비교사 22명 (○○대학교 초등교육과 1학년)	소요 시간	50분
학습 목표	순차, 반복을 이용하여 여러 가지 정다각형을 그릴 수 있다.		
학습 요소	교수·학습 활동		시간
도입	<ul style="list-style-type: none"> ▶ 동기유발 활동 <ul style="list-style-type: none"> - ‘도형 그리기’ 프로젝트 보여주기 ▶ 공감 단계 <ul style="list-style-type: none"> - ‘도형 그리기’ 프로젝트가 어떤 사람들에게 도움이 될지 떠올리기 		10
전개	<ul style="list-style-type: none"> ▶ 문제 정의 단계 <ul style="list-style-type: none"> - 해결할 문제에 대해 정의하기 		5
	<ul style="list-style-type: none"> ▶ 아이디어 발상 단계 <ul style="list-style-type: none"> - ‘도형 그리기’ 프로젝트에 필요한 스프라이트의 종류 확인하기 - 스프라이트의 특징 파악하기 - 그릴 도형의 원리 찾아보기 		10
	<ul style="list-style-type: none"> ▶ 시안 만들기 단계 <ul style="list-style-type: none"> - 도형 그리는 방법에 따라 프로그래밍 하기 - 변수를 활용한 방법으로 발전시키기 - 사용자의 입력 값을 바탕으로 동작하는 프로그램 만들기 		15
정리	<ul style="list-style-type: none"> ▶ 검증하기 단계 <ul style="list-style-type: none"> - 작성한 프로그램에 오류 확인하기 - 잘된 점과 개선할 점 생각하기 		10

(3) 스크래치 활용 디자인씽킹 기반 문제해결

6개의 그룹을 조직하여 2번의 문제해결 과정을 진행하였다. 그룹의 문제해결 주제는 <표 IV-8>과 같다.

<표 IV-8> 스크래치 활용 디자인씽킹 기반 문제해결 주제

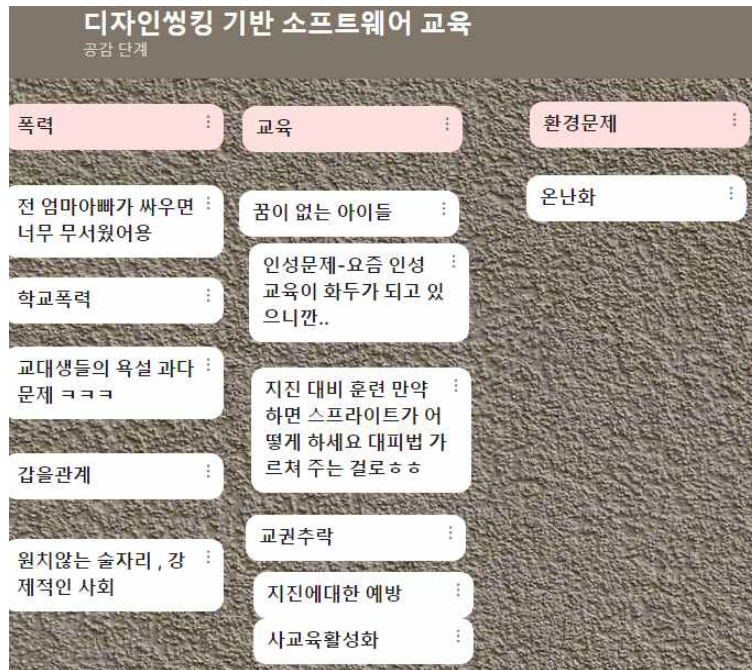
그룹	해결 주제1	해결 주제2
1 그룹	역사 교육	사회적 여성 차별
2 그룹	스마트폰 중독	잘못된 음주 문화
3 그룹	지진 대피 방법	사회적 약자의 피해
4 그룹	환경 문제	아프리카의 기아 문제
5 그룹	읽기 문제	소음 문제
6 그룹	비만 예방	다문화 학생 차별 문제

교사는 디자인씽킹 기반 문제해결 과정을 진행하는 동안 퍼실리테이터의 역할을 수행하였다. 디자인씽킹 각 단계에서 해결해야 할 일, 보이는 실수, 반복된 도전, 시행착오 등을 열린 마음으로 지켜보고, 도전 정신을 갖도록 격려하였다. 디자인씽킹 과정에서 협력이 필요한 상황에서는 CSCL 도구로 패들렛, 구글 프레젠테이션을 사용하였다.

다음 내용은 1그룹의 ‘사회적 여성 차별’이라는 주제로 디자인씽킹 기반의 문제해결 과정을 간략히 알아보려고 한다.

① 공감 단계

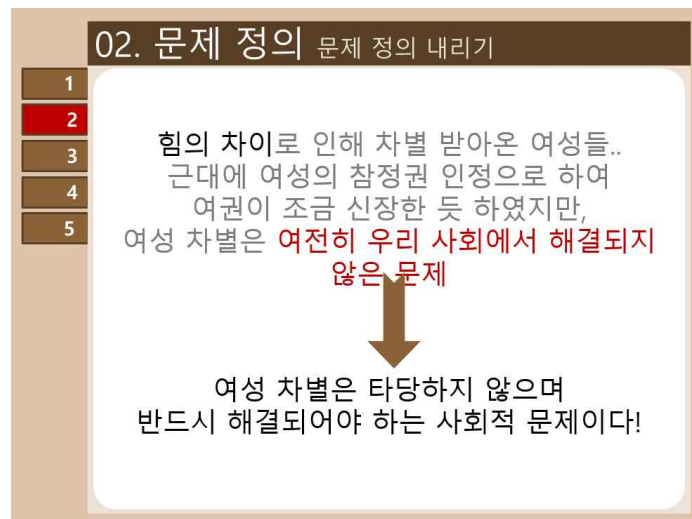
공감 단계에서는 패들렛을 이용하여 수업 시간이 아닌 과제로 제시하였다. 온라인 공간에서 다른 사람들과 인터뷰 또는 조사를 통해 해결해야 할 문제를 찾아내고 브레인스토밍(Brainstorming) 기법으로 최대한 다양한 의견을 제시하고 비판하지 않도록 하였다. 수업 시간에는 의견을 주제별로 목록화하는 작업을 실시하였다.



[그림 IV-6] 스크래치 활용 디자인씽킹 공감 단계 활동 장면

② 문제 정의 단계

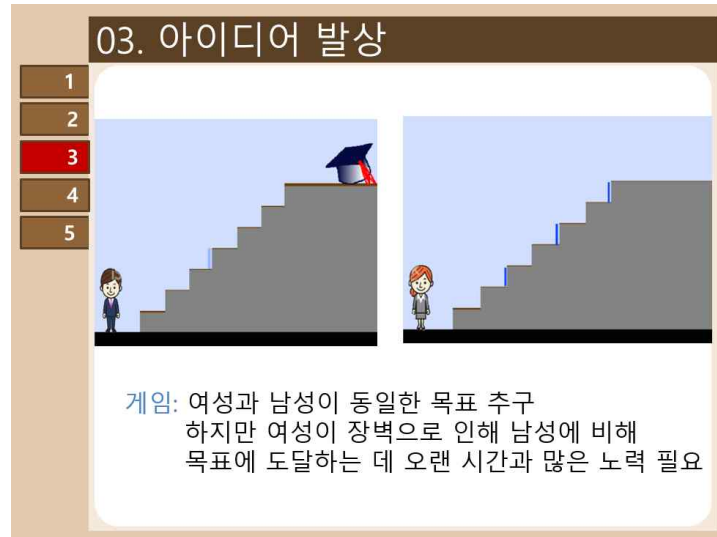
공감단계에서 찾은 문제들 중 의견 교환을 통해 해결하고자 하는 문제를 찾고 정의 내리는 단계이다. 학생들이 확산적 사고에서 수렴적 사고로 사고 전환을 하여 해결할 문제를 정하였다.



[그림 IV-7] 스크래치 활용 디자인씽킹 문제 정의 단계 활동 장면

③ 아이디어 발상 단계

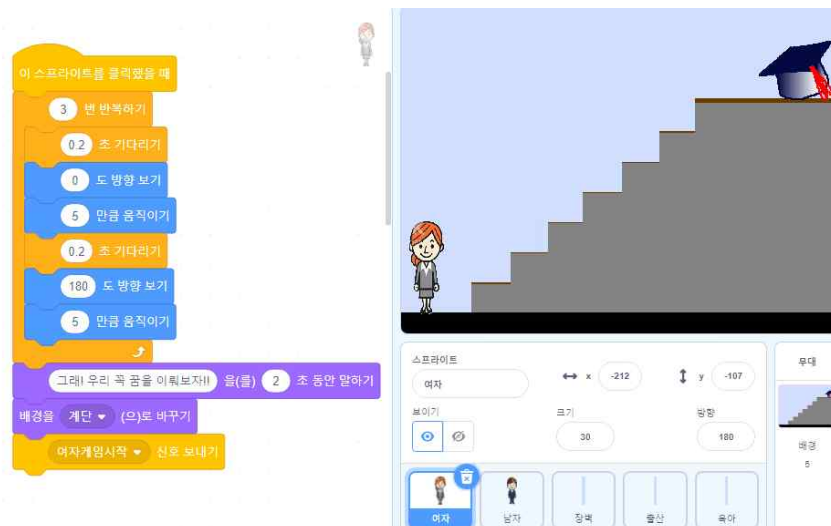
정의된 문제를 어떤 방법으로 해결할 것인지 아이디어를 떠올려 보는 단계이다. 결과물의 형태, 스프라이트의 종류, 스프라이트의 특징들에 대하여 구체화하였다.



[그림 IV-8] 스크래치 활용 디자인씽킹 아이디어 발상 단계 활동 장면

④ 시안 만들기 단계

문제 해결 아이디어를 바탕으로 시안을 만드는 단계이다. 직접 스크래치에서 프로그래밍을 제작하는 활동을 하였다.



[그림 IV-9] 스크래치 활용 디자인씽킹 시안 만들기 단계 활동 장면

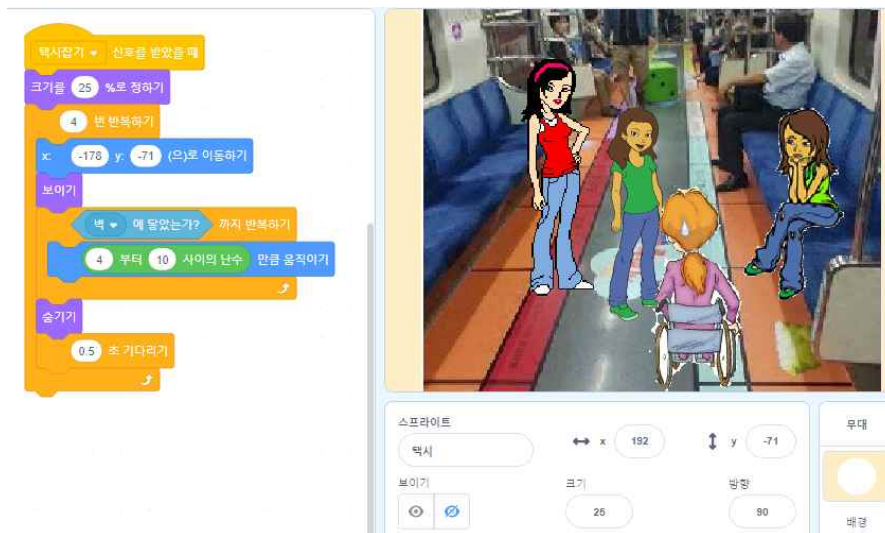
⑤ 검증하기 단계

제작한 시안을 발표하는 자리를 마련하고 직접 완성본을 검증해 보는 단계이다. 만든 시안이 잘 동작하는지 알아보고, 잘된 점과 추가하면 좋을 점에 대하여 공유하는 활동을 하였다.

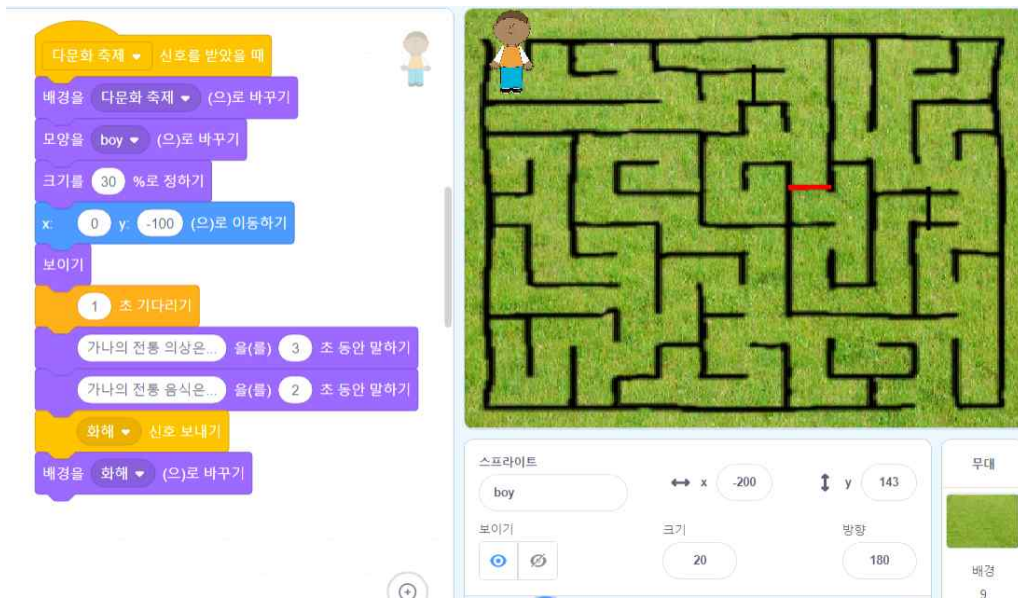


[그림 IV-10] 스크래치 활용 디자인씽킹 검증하기 단계 활동 장면

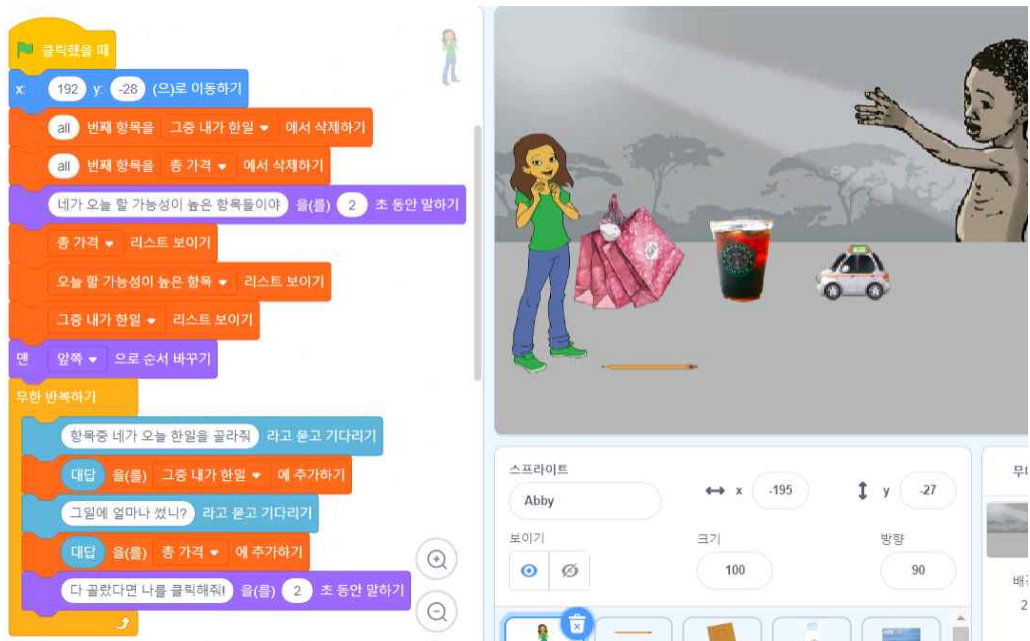
[그림 IV-11], [그림 IV-12], [그림 IV-13]은 그룹별로 제작한 시안들의 예시 작품들이다. 각각의 문제들을 스크래치를 이용해서 해결하였다.



[그림 IV-11] 디자인씽킹 기반 3그룹 ‘사회적 약자의 피해’ 활동 장면



[그림 IV-12] 디자인씽킹 기반 6그룹 ‘다문화 학생 차별 문제’ 활동 장면



[그림 IV-13] 디자인씽킹 기반 4그룹 ‘아프리카의 기아 문제’ 활동 장면

4) 연구 결과 분석

(1) 창의성 변화

① 정규성 검정

실험 집단이 정규성을 확보하고 있는지 확인하기 위하여 집단의 창의성 사전 검사 결과로 정규성 검정을 실시하였다. 창의성 사전 검사에 대한 비모수/모수 통계를 결정하기 위해 정규성 검정의 방법으로 Shapiro-Wilks 검정을 실시하였고 <표 IV-9>에 그 결과를 제시하였다.

<표 IV-9> 창의성 검사 정규성 검정

하위 요소	사례 수	평균	표준편차	최대값	최소값	통계량	<i>p</i>
창의성 지수	22	96.1636	10.87358	116.20	75.80	.956	.417
창의성 평균	22	93.8455	10.23438	111.20	73.80	.944	.240
유창성	22	98.0000	23.38090	150.00	58.00	.954	.374
독창성	22	93.0000	14.41890	128.00	73.00	.889	.018*
제목의 추상성	22	80.7273	26.36950	140.00	40.00	.916	.061
정교성	22	129.5000	13.14661	145.00	95.00	.793	.000**
성급한 종결에 대한 저항	22	68.0000	18.00529	110.00	40.00	.888	.017*

p*<.05, *p*<.01

실험집단의 정규성 검정 결과 ‘독창성’과 ‘정교성’, ‘성급한 종결에 대한 저항’에서 정규 분포를 이루지 않는 것으로 나타났다.

② 사전·사후 검사 집단 내 비교

실험집단 내 창의성 사전, 사후 검사 결과를 비교하기 위해, 창의성 사전 검사에 대한 정규성 검정 결과 정규분포를 이루는 하위요소들에 대해서는 대응표본 t 검정, 정규분포를 이루지 않는 하위요소에 대해서는 Wilcoxon 부호순위 검정을 실시하여 비교하였다.

실험집단의 창의성 사전·사후 검사를 비교·분석하여 <표 IV-10>과 <표 IV-11>에 제시하였다.

<표 IV-10> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)

하위 요소	사례수	사전 검사		사후 검사		t	p
		평균	표준편차	평균	표준편차		
창의성 지수	22	96.1636	10.87358	103.7545	14.13013	-4.415	.000**
창의성 평균	22	93.8455	10.23438	100.9818	12.83595	-5.024	.000**
유창성	22	98.0000	23.38090	106.1818	18.07560	-1.954	.064
제목의 추상성	22	80.7273	26.36950	80.4545	21.18564	.060	.952

* $p < .05$, ** $p < .01$

<표 IV-11> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)

하위 요소	사례수	사전 검사		사후 검사		z	p
		평균	표준편차	평균	표준편차		
독창성	22	93.0000	14.41890	108.0909	17.65519	-3.559	.000**
정교성	22	129.5000	13.14661	142.9091	14.13509	-3.169	.002**
성급한 종결에 대한 저항	22	68.0000	18.00529	67.2273	30.19923	-.552	.581

* $p < .05$, ** $p < .01$

실험집단의 창의성 사전·사후 검사 결과를 비교·분석한 결과 ‘창의성 지수’는 사전 검사 평균 96.1636에서 사후 검사 평균 103.7545로 7.5909만큼 증가하였으며, 검증통계량 t 값이 -4.415, 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘창의성 평균’에서는 사전 검사 평균 93.8455에서 사후 검사 평균 100.9818로 7.1363만큼 증가하였으며, 검증통계량 t 값이 -5.024, 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘독창성’에서는 사전 검사 평균 93.0000에서 사후 검사 평균 108.0909로 15.0909만큼 증가하였으며, 검증통계량 z 값이 -3.559, 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘정교성’에서는 사전 검사 평균 129.5000에서 사후 검사 평균 142.9091로 13.4091만큼 증가하였으며, 검증통계량 z 값이 -3.169, 유의확률 p 값은 .002로 99% 신뢰수준에서 유의미한 차이를 보였다.

‘유창성’에서는 사전 검사 평균 98.0000에서 사후 검사 평균 106.1818로 8.1818만큼 증가하였으나 검증통계량 t 값이 -1.954 , 유의확률 p 값은 $.064$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다. ‘제목의 추상성’에서는 사전 검사 평균 80.7273에서 사후 검사 평균 80.4545로 0.2728만큼 감소하였으나, 검증통계량 t 값이 $.060$, 유의확률 p 값은 $.952$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다. ‘성급한 종결에 대한 저항’에서는 사전 검사 평균 68.0000에서 사후 검사 평균 67.2273으로 0.7727만큼 감소하였으나, 검증통계량 z 값이 $-.552$, 유의확률 p 값은 $.581$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다.

(2) 창의성 검사 결과의 분석

스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등 예비교사의 창의성에 미치는 효과를 분석하고자 하였다. 이를 위하여 ○○대학교 초등교육과 1학년 22명의 초등 예비교사들을 대상으로 연구를 실시하였다. 연구 대상인 초등 예비교사들에게 디자인씽킹 기반 소프트웨어 교육을 진행하기 전, 후에 창의성 사후 검사를 실시하였다. 창의성 검사도구로는 Torrance의 TTCT 도형 검사지 A, B형을 사용하여 분석하였다.

검증 결과, 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등 예비교사의 창의성 신장에 효과적인 것으로 나타났다. 이는 CSCL를 활용한 디자인씽킹 기반의 협력학습을 통해 주변의 문제를 발견하고 다양한 해결 아이디어를 찾는 과정이 창의성 신장에 기여했다고 보여진다.

또한 창의성 하위 요소 중 ‘독창성’, ‘정교성’을 신장시키는데 효과적이다. 스크래치라는 교육 도구의 특성상 가상 공간에서 프로젝트를 제작하다 보니 실물 공간에 제약을 받지 않고 해결 방법을 제시하는 것이 ‘독창성’을 신장시키는 데 도움이 된 것으로 판단되며, 코딩이 복잡해 질 때가 많고 객체 지향적인 언어이기 때문에 정교한 작업이 필요한 것이 ‘정교성’에 영향을 준 것으로 분석된다. ‘유창성’에서는 95% 신뢰수준에서 유의미하지는 않았으나 신뢰수준에 근접한 유의확률을 보여서 평균의 상승에 영향을 미쳤다고 볼 수 있을 것이다.

2. 앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

1) 요구 분석

앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램에 대하여 다음과 같은 내용의 요구 분석을 실시하였다.

- 스크래치 사용 경험
- 디자인씽킹 교육 이해도
- 소프트웨어 교육 방법 선호도

요구 분석은 ○○대학교 겨울방학 교육기부를 신청한 4-6학년 학생 28명을 대상으로 실시하였다.

<표 IV-12>와 같이 앱인벤터 사용 경험에 대한 조사 결과는 대다수의 학습자들이 경험이 없거나 적은 것으로 나타났다.

<표 IV-12> 앱인벤터 사용 경험

내용	사용한 적이 없다	사용 경험이 적다	보통이다	사용 경험이 많다.	사용 경험이 아주 많다.
앱인벤터 프로그램을 사용해 본 적이 있나요?	16(57.1%)	8(28.6%)	2(7.1%)	1(3.6%)	1(3.6%)

<표 IV-13>은 디자인씽킹 교육에 대한 이해도를 조사한 결과이다. 대다수의 학습자들이 디자인씽킹 교육에 대해 들어본 적 없다는 의견을 제시하였다.

<표 IV-13> 앱인벤터 학습자 디자인씽킹 교육 이해도

내용	들어본 적 없다.	잘 모른다.	보통이다	조금 알고 있다.	아주 잘 알고 있다.
디자인씽킹 교육에 대하여 알고 있나요?	22(78.6%)	6(21.4%)	0	0	0

<표 IV-14>에 의하면 학습자들은 소프트웨어 교육 학습 방법에서 협력 문제 해결학습과 을 선호하는 것으로 나타났다.

<표 IV-14> 앱인벤터 소프트웨어 교육 학습 방법 선호도

내용	강의/실습	개인 문제해결학습	협력 문제해결학습	프로젝트학습
소프트웨어 교육을 받을 때 어떤 방법으로 학습하고 싶나요? (중복)	3(5.8%)	13(25.0%)	20(38.5%)	16(30.7%)

요구 분석 결과 학습자들은 앱인벤터를 거의 사용해 보지 않았고, 디자인씽킹 교육에 대한 이해도가 낮은 것을 고려하여 스크래치 교육 방향을 설정하였다. 먼저 학생들이 앱인벤터 사용 방법을 학습할 때 디자인씽킹 교육 방법을 경험할 수 있도록 수업교재를 설계하였다. 또한, 학생들이 교육 학습 방법으로 협력하여 문제해결할 수 있도록 팀을 구성하여 교육을 진행하였다.

2) 연구 설계 및 개발

(1) 연구 가설

- ① 연구가설 : 앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 있다.
- ② 귀무가설: 앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 없다.

(2) 연구 대상

○○대학교 겨울방학 교육기부를 신청한 4-6학년 학생을 대상으로 연구를 실시하였다. 성별 구성은 <표 IV-15>와 같다.

<표 IV-15> 실험 집단 성별 구성

	여학생	남학생	합계
실험 집단	14	14	28

성별에 따른 학년 구성은 <표 IV-16>과 같다.

<표 IV-16> 성별에 따른 학년 구성

	4학년	5학년	6학년	합계
여학생	7	3	4	14
남학생	3	8	3	14

(3) 연구 설계

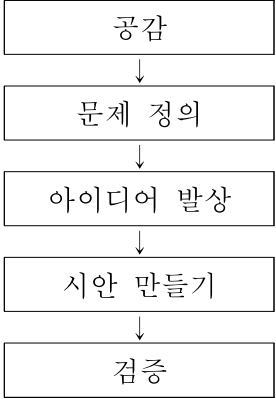
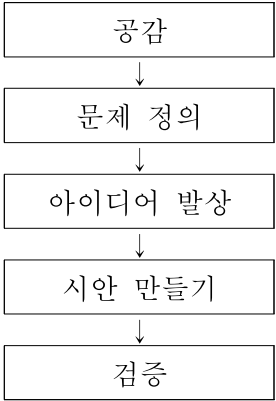
앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등학생의 창의성에 미치는 효과를 알아보기 위하여 실험연구를 실시하였다.

초등학생 28명을 대상으로 실험집단을 구성하였으며, 앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육 실시 사전, 사후에 창의성 검사를 실시하여 그 효과를 검증하고자 하였다.

(4) 교육 내용의 선정

요구 분석 결과 및 디자인씽킹 기반 소프트웨어 교육 도구의 특징에 따라 <표 IV-17>과 같이 앱인벤티를 활용한 교육 내용을 선정하였다.

<표 IV-17> 앱인벤티 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제

차시	학습 주제	학습 과정
1-2	<ul style="list-style-type: none"> • 앱인벤티 기본 기능 익히기 • 디자인씽킹 이해하기 	
3-18	• 청각 장애 도우미 앱	- 디자인씽킹 기반 문제해결 과정으로 다음 학습 단계를 따름. 
	• 받아쓰기 앱	
	• 그림판 앱	
	• 두더지 잡기 앱	
	• 사칙연산 앱	
	• 권장칼로리 계산 앱	
	• 비만도 계산 앱	
• 영어 단어 맞히기 앱		
19-24	• 디자인씽킹 기반 그룹별 협력 프로젝트 과정(기본)	- 디자인씽킹 기반 그룹별 협력 프로젝트 과정 후 공유와 피드백
25-36	• 만보계 앱	- 디자인씽킹 기반 문제해결 과정으로 다음 학습 단계를 따름. 
	• SOS 구조 요청 앱	
	• 운전 중 문자금지 앱	
	• 주문 앱	
	• 악기연주 앱	
• 초시계 앱		
37-42	• 디자인씽킹 기반 그룹별 협력 프로젝트 과정(심화)	- 디자인씽킹 기반 그룹별 협력 프로젝트 과정 후 공유와 피드백

교육 자료를 통하여 디자인씽킹 과정과 소프트웨어 도구의 사용법을 익힌 후 각각의 문제 해결을 통해 디자인씽킹 단계에 따라 앱인벤토를 활용하여 문제를 해결하였다.

(5) 검사 도구

창의성 검사 도구로 TTCT(도형) A, B형을 선정하여 투입하였다. 이 검사를 통하여 '유창성', '독창성', '제목의 추상성', '정교성', '성급한 종결에 대한 저항'을 측정하였다. 통계 분석 도구는 SPSS를 사용하였다.

<표 IV-18> 앱인벤토 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계

대상	사전 검사	처치	사후 검사
초등학교 4~6학년	O ₁	X ₁	O ₂

X₁ : 앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

O₁ : 사전 검사(창의성 검사)

⇒ 정규성 검정

O₁, O₂ : 사전·사후 검사(창의성 검사)

⇒ 정규성 검정 결과에 따른 대응표본 t 검정 또는 Wilcoxon 부호 순위 검정

3) 적용

(1) 앱인벤토를 활용한 디자인씽킹 기반 소프트웨어 교육 학습하기

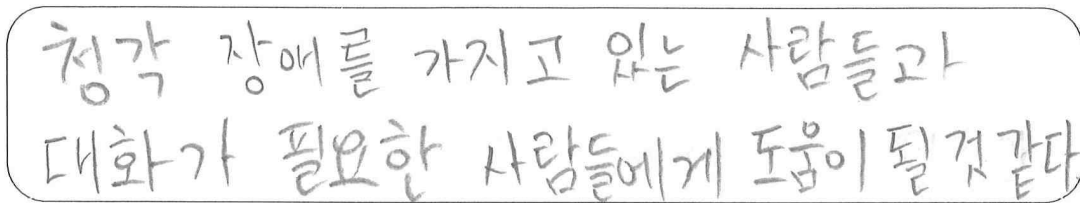
디자인씽킹 기반 소프트웨어 교육에서 각 단계를 직접 경험해 보는 것을 통해 디자인씽킹에 대한 이해를 높이면서 소프트웨어 교육을 진행하였다. 청각 장애 도우미 앱 만들기를 주제로 디자인씽킹 기반 소프트웨어 교육 교재를 바탕으로 학습한 내용을 살펴보고 한다.

① 공감 단계

공감 단계에서는 학습을 통해 제작할 앱이 어떤 문제를 갖은 사람들에게 도움이 될지 문제를 찾는 과정으로 [그림 IV-14]와 같이 활동하였다

1. 공감 단계 [Empathize]

- 어떤 문제를 갖은 사람들에게 도움이 될지 적어보세요.



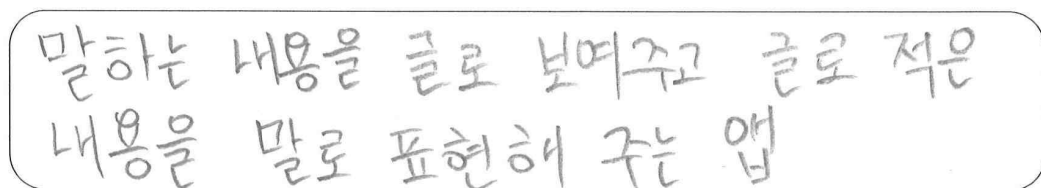
[그림 IV-14] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계

② 문제 정의 단계

문제 정의 단계에서는 공감 단계에서 생각한 문제를 명확히 하는 과정으로 [그림 IV-15]와 같이 활동하였다.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.



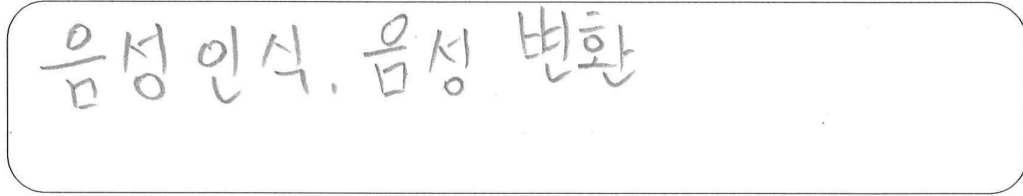
[그림 IV-15] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계

③ 아이디어 발상 단계

아이디어 발상 단계에서는 청각 장애인이 대화를 편하기 위해 필요한 기능을 알아보는 등 문제를 해결하기 위한 다양한 아이디어를 도출하는 과정으로 [그림 IV-16]과 같이 활동하였다.

3. 아이디어 발상 단계 [Ideate]

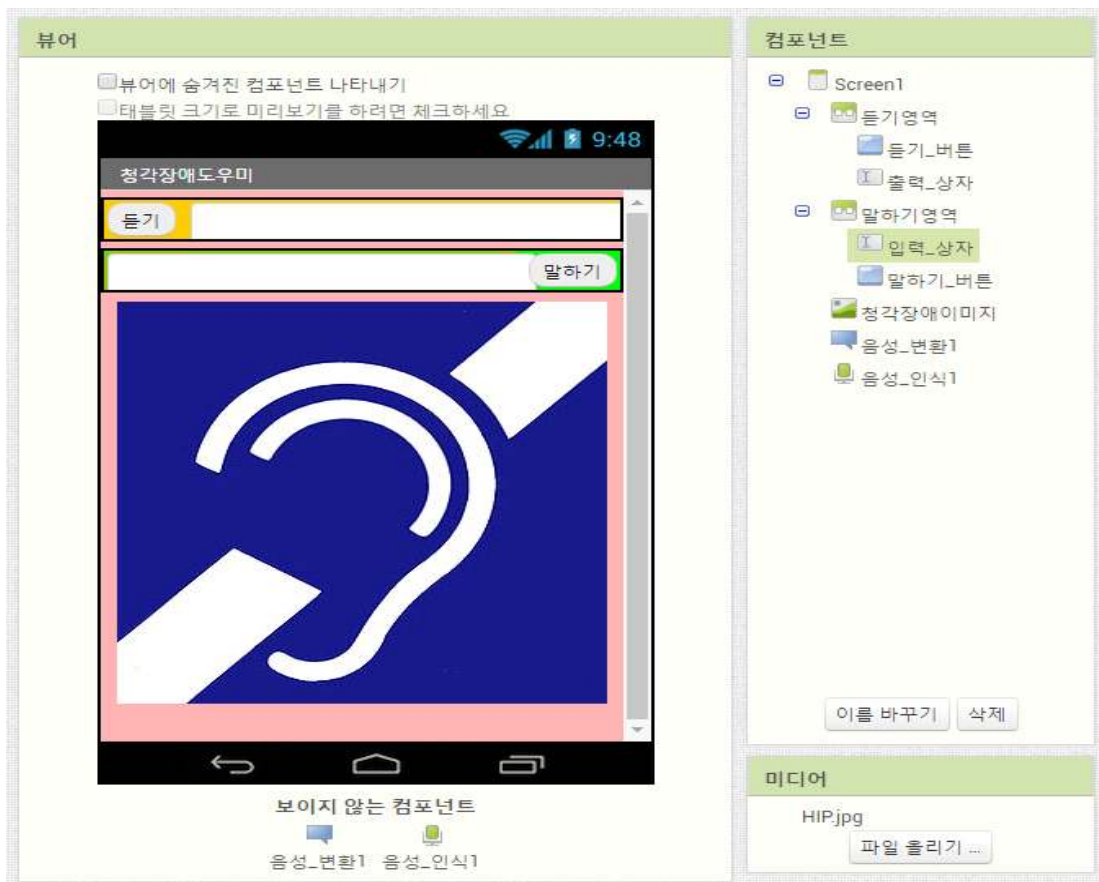
- 청각 장애인이 대화를 편하게 위해 어떤 기능이 필요한지 적어보세요.



[그림 IV-16] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계

④ 시안 만들기 단계

시안 만들기 단계에서는 직접 앱인벤터를 활용하여 청각 장애인과 대화를 도와주는 앱을 작성하는 과정으로 [그림 IV-17]과 같이 활동하였다.



[그림 IV-17] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계

⑤ 검증하기 단계

제작한 앱의 오류 검증, 앱의 장점, 앱에 추가하면 좋을 점에 대하여 알아보는 과정으로 [그림 IV-18]과 같이 활동하였다.

5. 검증하기 단계 [Test]

1) 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

키보드가 나타나지 않아 키보드가 보이는 블록을 추가했다.

2) 만든 앱의 장점을 적어주세요.

스마트폰을 이용해서 청각 장애인과 대화할 수 있다.

3) 추가하면 좋을 내용을 적어주세요.

한국어 뿐만 아니라 영어, 중국어 같이 다른 나라 말도 사용하면 좋겠다.

[그림 IV-18] 앱인벤터 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계

(2) 애플리케이션 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안
 앞서 소개했던 디자인씽킹 기반 소프트웨어 교육을 하기 전에 일반적인 수업
 에서 가장 많이 사용되는 ‘도입-전개-정리’의 구조를 사용하여 교사용 교수·학습
 과정안을 구성하였다. 학생들이 실생활의 문제에 대해 애플리케이션을 활용하여 디자
 인씽킹 과정을 경험할 수 있도록 과정안을 구성하였다.

<표 IV-19> 애플리케이션 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안

주제	청각 장애 도우미 앱	차시	3-4차시
대상	초등학교 4~6학년 28명	소요 시간	80분
학습 목표	미디어 기능을 이용하여 청각 장애 도우미 앱을 만들 수 있다.		
학습 요소	교수·학습 활동		시간
도입	<ul style="list-style-type: none"> ▶ 동기유발 활동 <ul style="list-style-type: none"> - ‘청각 장애 도우미’ 앱 보여주기 ▶ 공감 단계 <ul style="list-style-type: none"> - ‘청각 장애 도우미’ 앱이 어떤 사람들에게 도움이 될지 떠올리기 		10
전개	<ul style="list-style-type: none"> ▶ 문제 정의 단계 <ul style="list-style-type: none"> - 해결할 문제에 대해 정의하기 		10
	<ul style="list-style-type: none"> ▶ 아이디어 발상 단계 <ul style="list-style-type: none"> - 컴포넌트의 종류 확인하기 - 컴포넌트의 특징 및 기능 파악하기 		20
	<ul style="list-style-type: none"> ▶ 시안 만들기 단계 <ul style="list-style-type: none"> - 디자이너에서 컴포넌트 구성하기 - 블록으로 동작 설정하기 		30
정리	<ul style="list-style-type: none"> ▶ 검증하기 단계 <ul style="list-style-type: none"> - 작성한 프로그램에 오류 확인하기 - 잘된 점과 개선할 점 생각하기 		10

(3) 앱인벤터 활용 디자인씽킹 기반 문제해결

7개의 그룹을 조직하여 2번의 문제해결 과정을 진행하였다. 그룹의 문제해결 주제는 <표 IV-20>과 같다.

<표 IV-20> 앱인벤터 활용 디자인씽킹 기반 문제해결 주제

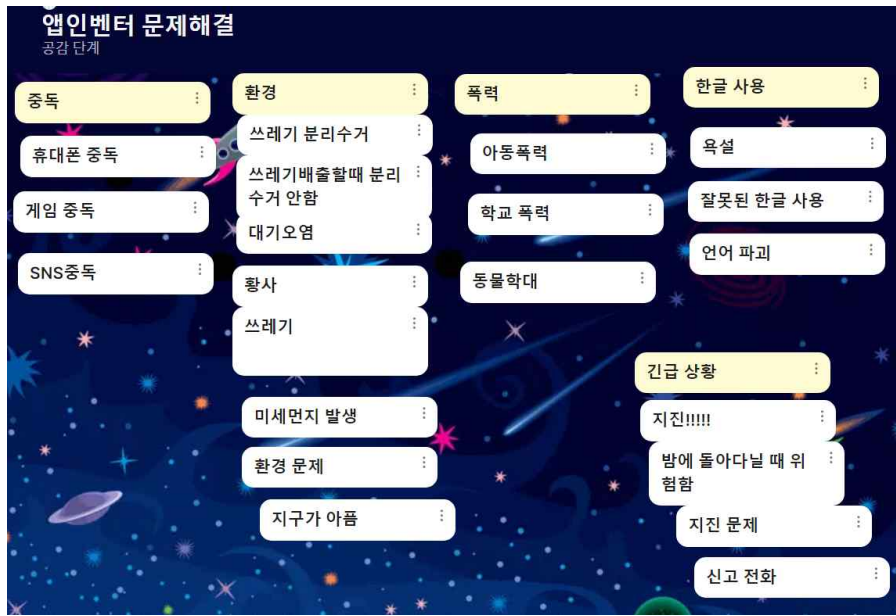
그룹	해결 주제1	해결 주제2
1 그룹	게임 중독 해결	학교 폭력 방지
2 그룹	올바른 외래어 표기	비만 줄이기
3 그룹	영유아의 편식 문제	독도 문제 해결
4 그룹	쓰레기 무단 투기	맞춤법 맞히기
5 그룹	학생들의 과도한 사교육	쉬운 피아노 연습
6 그룹	요일별 쓰레기 분리 배출	한국 역사 이해
7 그룹	TV 중독 해결	예쁜 글씨 쓰기

교사는 디자인씽킹 기반 문제해결 과정을 진행하는 동안 퍼실리테이터의 역할을 수행하였다. 디자인씽킹 각 단계에서 해결해야 할 일, 보이는 실수, 반복된 도전, 시행착오 등을 열린 마음으로 지켜보고, 도전 정신을 갖도록 격려했다. 디자인씽킹 과정에서 협력이 필요한 상황에서는 CSCL 도구로 패들렛, 구글 프레젠테이션을 사용하였다.

다음 내용은 6그룹의 ‘요일별 쓰레기 분리 배출’이라는 주제로 디자인씽킹 기반의 문제해결 과정을 간략히 알아보려고 한다.

① 공감 단계

공감 단계에서는 패들렛을 이용하여 수업 시간이 아닌 과제로 제시하였다. 온라인 공간에서 다른 사람들과 인터뷰 또는 조사를 통해 해결해야 할 문제를 찾아내고 브레인스토밍 기법으로 최대한 다양한 의견을 제시하고 비판하지 않도록 하였다. 수업 시간에는 의견을 주제별로 목록화하는 작업을 실시하였다.



[그림 IV-19] 앱인벤터 활용 디자인씽킹 공감 단계 활동 장면

② 문제 정의 단계

공감단계에서 찾은 문제들 중 의견 교환을 통해 해결하고자 하는 문제를 찾고 정의 내리는 단계이다.

2. 문제 정의

문제 정의 내리기

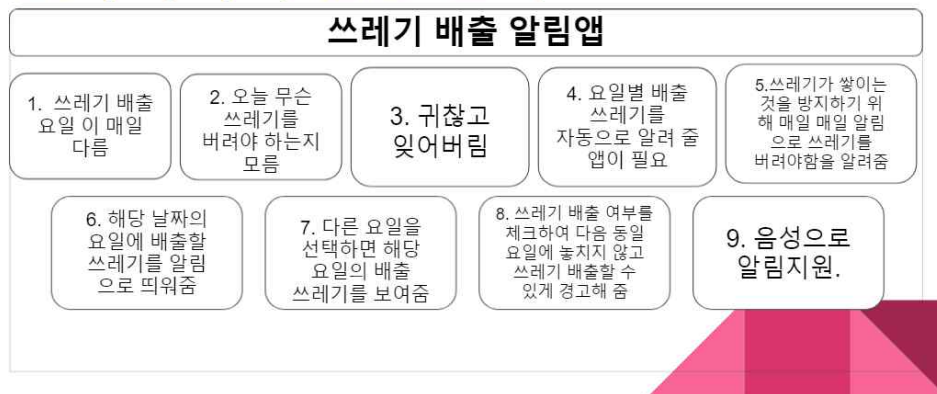
사람들이 요일별 배출하는
쓰레기가 뭔지 알게 하는
방법이 필요하다.

[그림 IV-20] 앱인벤터 활용 디자인씽킹 문제 정의 단계 활동 장면

③ 아이디어 발상 단계

정의된 문제를 어떤 방법으로 해결할 것인지 아이디어를 떠올려 보는 단계로 앱의 형태, 컴포넌트의 종류, 컴포넌트의 기능들에 대하여 구체화하는 활동을 하였다.

3.아이디어 발상 단계



[그림 IV-21] 앱인벤터 활용 디자인씽킹 아이디어 발상 단계 활동 장면

④ 시안 만들기 단계

문제 해결 아이디어를 바탕으로 시안을 만드는 단계이다. 직접 스크래치에서 프로그래밍을 제작하는 활동을 하였다.



[그림 IV-22] 앱인벤터 활용 디자인씽킹 시안 만들기 단계 활동 장면

⑤ 검증하기 단계

제작한 시안을 발표하는 자리를 마련하고 직접 완성본을 검증해 보는 단계이다. 제작한 앱을 QR코드로 제공하여 다운로드 받을 수 있게 하였다. 학생들이 직접 자신의 모바일 기기에서 동작시켜 보고 오류가 있는지 검증해 보았다. 또한 잘된 점과 추가하면 좋을 점에 대하여 공유하는 활동을 하였다.

5.테스트



[그림 IV-23] 앱인벤터 활용 디자인씽킹 검증하기 단계 활동 장면

[그림 IV-24], [그림 IV-25], [그림 IV-26]은 그룹별로 제작한 시안들의 예시 작품들이다. 각각의 문제들을 앱인벤터를 이용해서 해결하였다.



[그림 IV-24] 디자인씽킹 기반 1그룹 '게임 중독' 활동 장면



[그림 IV-25] 디자인씽킹 기반 2그룹 ‘비만 줄이기’ 활동 장면



[그림 IV-26] 디자인씽킹 기반 5그룹 ‘쉬운 피아노 연습’ 활동 장면

4) 연구 결과 분석

(1) 창의성 변화

① 정규성 검정

실험집단이 정규성을 확보하고 있는지 확인하기 위하여 집단의 창의성 사전 검사 결과로 정규성 검정을 실시하였다. 창의성 사전 검사에 대한 비모수/모수 통계를 결정하기 위해 정규성 검정의 방법으로 Shapiro-Wilks 검정을 실시하였고 <표 IV-21>에 그 결과를 제시하였다.

<표 IV-21> 창의성 검사 정규성 검정

하위 요소	사례 수	평균	표준편차	최대값	최소값	통계량	p
창의성 지수	28	85.3714	17.41023	118.60	46.00	.979	.834
창의성 평균	28	84.1571	16.32643	113.60	46.00	.972	.625
유창성	28	114.6071	25.55363	147.00	53.00	.923	.041*
독창성	28	95.3214	18.64692	123.00	40.00	.939	.103
제목의 추상성	28	46.1429	39.98492	148.00	0.00	.890	.007**
정교성	28	118.4286	22.18680	147.00	43.00	.869	.002**
성급한 종결에 대한 저항	28	43.7857	35.80747	106.00	0.00	.884	.005**

* $p < .05$, ** $p < .01$

실험집단의 정규성 검정 결과 ‘유창성’과 ‘제목의 추상성’, ‘정교성’, ‘성급한 종결에 대한 저항’에서 정규 분포를 이루지 않는 것으로 나타났다.

② 사전·사후 검사 집단 내 비교

실험집단 내 창의성 사전, 사후 검사 결과를 비교하기 위해, 창의성 사전 검사에 대한 정규성 검정 결과 정규분포를 이루는 하위요소들에 대해서는 대응표본 t 검정, 정규분포를 이루지 않는 하위요소에 대해서는 Wilcoxon 부호순위 검정을 실시하여 비교하였다.

실험집단의 창의성 사전·사후 검사를 비교·분석하여 <표 IV-22>와 <표 IV-23>에 제시하였다.

<표 IV-22> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)

하위 요소	사례수	사전 검사		사후 검사		t	p
		평균	표준편차	평균	표준편차		
창의성 지수	28	85.3714	17.41023	101.1143	18.07453	-4.709	.000**
창의성 평균	28	84.1571	16.32643	99.7214	16.96553	-4.787	.000**
독창성	28	95.3214	18.64692	107.2143	17.54224	-3.045	.005**

* $p < .05$, ** $p < .01$

<표 IV-23> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)

하위 요소	사례수	사전 검사		사후 검사		z	p
		평균	표준편차	평균	표준편차		
유창성	28	114.6071	25.55363	125.1786	27.38079	-2.733	.006**
제목의 추상성	28	46.1429	39.98492	60.6071	39.46807	-1.660	.097
정교성	28	118.4286	22.18680	130.1429	13.77253	-2.677	.007**
성급한 종결에 대한 저항	28	43.7857	35.80747	75.4643	36.24861	-3.710	.000**

* $p < .05$, ** $p < .01$

실험집단의 창의성 사전·사후 검사 결과를 비교·분석한 결과 ‘창의성 지수’는 사전 검사 평균 85.8714에서 사후 검사 평균 101.1143으로 15.7429만큼 증가하였으며, 검증통계량 t 값이 -4.709 , 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 것으로 나타났다. ‘창의성 평균’에서는 사전 검사 평균 84.1571에서 사후 검사 평균 99.7214로 15.5643만큼 증가하였으며, 검증통계량 t 값이 -4.787 , 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘독창성’에서는 사전 검사 평균 95.3214에서 사후 검사 평균 107.2143으로 11.8929만큼 증가하였으며, 검증통계량 t 값이 -3.045 , 유의확률 p 값은 $.005$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘유창성’에서는 사전 검사 평균 114.6071에서 사후 검사 평균 125.1786으로 10.5715만큼 증가하였으며, 검증통계량 z 값이 -2.733 , 유의확률 p 값은 $.006$ 으로 99% 신뢰수준에서 유의미한 것으로 나타났다. ‘정교성’에서는 사전 검사 평균 118.4286에서 사후 검사 평균 130.1429로 11.7143만큼 증가하였으며, 검증통계량 z 값이 -2.677 , 유의확률 p 값은 $.007$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘성급한 종결에 대한 저항’에서는 사전 검사 평균 43.7857에서 사후 검사 평균 75.4643으로 31.6786만큼 증가하였으며, 검증통계량 z 값이 -3.710 , 유의확률 p 값은 $< .001$ 로 99% 신뢰수준에서 유의미한 차이를 보였다.

‘제목의 추상성’에서는 사전 검사 평균 46.1429에서 사후 검사 평균 60.6071로 14.4642만큼 증가하였으나, 검증통계량 z 값이 -1.660 , 유의확률 p 값은 $.097$ 로 95% 신뢰수준에서 유의미하지 않은 것으로 나타났다.

(2) 창의성 검사 결과 분석

앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등학생의 창의성에 미치는 효과를 분석하고자 하였다. 이를 위하여 ○○대학교 겨울방학 교육기부를 신청한 4-6학년 학생 28명을 대상으로 연구를 실시하였다. 실험집단은 앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하여 진행하였다. 창의성 사전 검사를 실시하고 42시간에 걸쳐 디자인씽킹 기반 소프트웨어 교육을 진행한 후 창의성 사후 검사를 실시하였다. 창의성 검사도구로는 Torrance의

TTCT 도형 검사지 A, B형을 사용하여 분석하였다.

검증 결과, 앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등학생의 창의성 신장에 효과적인 것으로 나타났다. 이는 CSCL를 활용한 디자인씽킹 기반 협력학습을 통해 주변의 문제를 찾고 다양한 아이디어를 도출하며 이를 동료 학습자들과 협업하여 문제를 해결하는 과정이 창의성에 영향을 준 것으로 사료된다.

또한 창의성 하위 요소 중 ‘유창성’, ‘독창성’, ‘정교성’, ‘성급한 종결에 대한 저항’을 신장시키는데 효과적이었다. 이는 앱인벤터라는 교육 도구의 특성상 학생들이 스마트폰에 대한 관심도 많고 직접 사용해 보면서 필요한 앱에 대한 생각을 많이 해 본 것 같다. 그런 만큼 주변의 문제 해결을 위해 창의적인 생각을 많이 한 것으로 판단된다.

3. 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

1) 요구 분석

마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램에 대하여 다음과 같은 내용의 요구 분석을 실시하였다.

- 마이크로비트 사용 경험
- 디자인씽킹 교육 이해도
- 소프트웨어 교육 방법 선호도

요구 분석은 ○○대학교 SWEET 지원 사업 프로그램에 참여한 제주특별자치도 내 J, B 2개 초등학교 5-6학년 재학생 22명을 대상으로 실시하였다.

<표 IV-24>와 같이 마이크로비트 사용 경험에 대한 조사 결과는 사용 경험이 적은 학생들이 많은 것으로 나타났다.

<표 IV-24> 마이크로비트 사용 경험

내용	사용한 적이 없다	사용 경험이 적다	보통이다	사용 경험이 많다.	사용 경험이 아주 많다.
마이크로비트 프로그램을 사용해 본 적이 있나요?	5(22.7%)	8(36.4%)	4(18.2%)	3(13.6%)	2(9.1%)

<표 IV-25>는 디자인씽킹 교육에 대한 이해도를 조사한 결과이다. 학습자들 대부분이 디자인씽킹 교육에 대해 들어본 적이 없다는 의견을 제시하였다.

<표 IV-25> 마이크로비트 학습자 디자인씽킹 교육 이해도

내용	들어본 적이 없다.	잘 모른다.	보통이다	조금 알고 있다.	아주 잘 알고 있다.
디자인씽킹 교육에 대하여 알고 있나요?	20(90.9%)	2(9.1%)	0	0	0

<표 IV-26>에 의하면 학습자들은 소프트웨어 교육 학습 방법에서 협력 문제 해결학습과 을 선호하는 것으로 나타났다.

<표 IV-26> 마이크로비트 학습자 소프트웨어 교육 학습 방법 선호도

내용	강의/실습	개인 문제해결학습	협력 문제해결학습	프로젝트학습
소프트웨어 교육을 받을 때 어떤 방법으로 학습하고 싶나요? (중복)	3(8.1%)	6(16.2%)	16(43.3%)	12(32.4%)

요구 분석 결과 마이크로비트를 사용해 보지 않은 학습자들이 많았고, 디자인씽킹 교육에 대한 이해도가 낮은 것을 고려하여 마이크로비트 교육 방향을 설정하였다. 먼저 학생들이 마이크로비트 사용 방법을 학습할 때 디자인씽킹 교육 방법을 경험하도록 수업교재를 설계하였다. 또한, 학생들이 교육 학습 방법으로 협력하여 문제해결할 수 있도록 팀을 구성하여 진행하였다.

2) 연구 설계 및 개발

(1) 연구 가설

- ① 연구가설 : 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 있다.
- ② 귀무가설: 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 학습자의 창의성을 신장시킬 수 없다.

(2) 연구 대상

본 연구에서 적용한 프로그램의 교육적 효과를 확인하기 위해 ○○대학교 SWEET 지원 사업 프로그램에 참여한 제주특별자치도내 J, B 2개 초등학교 5-6학년 재학생 22명을 대상으로 진행하였다. 교육에 참여한 학생들은 블록 기반의 소프트웨어 교육을 받은 경험이 있지만 피지컬 컴퓨팅을 활용한 교육경험은 없었다.

(3) 연구 설계

본 연구에서 개발한 교육 프로그램은 주말을 이용하여 <표 IV-29>와 같이 5일 동안 1일 4차시씩 총 20차시의 집합 강의 및 실습으로 이루어졌다. 첫날에는 오리엔테이션, 사전 검사, 마이크로비트 기초, 디자인씽킹 이해를 중심으로 진행하였으며, 마지막 날에는 사후 검사와 학습자들이 만든 프로젝트를 공유하고 피드백하는 시간으로 이루어졌다.

(4) 교육 내용의 선정

요구 분석 결과 및 소프트웨어 교육 도구의 특징에 따라 <표 IV-27>과 같이 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 내용을 선정하였다.

<표 IV-27> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 프로그램 주제

차시	학습 주제	학습 과정
1-2	<ul style="list-style-type: none"> • 마이크로비트 기초 기능 이해하기 • 디자인씽킹 이해하기 	
3-12	<ul style="list-style-type: none"> • 자동 전등 프로그램 • 적정 온도 알림 프로그램 • 나침반 프로그램 • 타이머 프로그램 • 통신하기 프로그램 • 도난 방지 프로그램 • 식물 습도 알림 프로그램 • 전자 피아노 프로그램 • 움직임 감지 프로그램 • 사회적 거리두기 프로그램 	<p>- 디자인씽킹 기반 문제해결 과정으로 다음 학습 단계를 따름.</p> <div style="text-align: center;"> <pre> graph TD A[공감] --> B[문제 정의] B --> C[아이디어 발상] C --> D[시안 만들기] D --> E[검증] </pre> </div>
13-20	<ul style="list-style-type: none"> • 디자인씽킹 기반 그룹별 협력 프로젝트 과정(기본) 	<p>- 디자인씽킹 기반 그룹별 협력 프로젝트 과정 후 공유와 피드백</p>

(5) 검사 도구

창의성 검사 도구로 TTCT(도형) A, B형을 선정하여 투입하였다. 이 검사를 통하여 '유창성', '독창성', '제목의 추상성', '정교성', '성급한 종결에 대한 저항'을 측정하였다. 통계 분석 도구는 SPSS를 사용하였다.

<표 IV-28> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 실험 설계

대상	사전 검사	처치	사후 검사
초등학교 5-6학년	O ₁	X ₁	O ₂

X₁ : 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램

O₁ : 사전 검사(창의성 검사)

⇒ 정규성 검정

O₁, O₂ : 사전·사후 검사(창의성 검사)

⇒ 정규성 검정 결과에 따른 대응표본 t 검정 또는 Wilcoxon 부호 순위 검정

3) 적용

(1) 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 학습하기

디자인씽킹 기반 소프트웨어 교육에서 각 단계를 직접 경험해 보는 것을 통해 디자인씽킹에 대한 이해를 높이면서 소프트웨어 교육을 진행하였다. 사회적 거리 두기 프로그램 만들기를 주제로 디자인씽킹 기반 소프트웨어 교육 교재를 바탕으로 학습한 내용을 살펴보고 한다.

① 공감 단계

공감 단계에서는 학습을 통해 제작할 프로그램이 어떤 문제를 갖은 사람들에게 도움이 될지 문제를 찾는 과정으로 [그림 IV-27]과 같이 활동하였다.

1. 공감 단계 [Empathize]

- 어떤 문제를 갖은 사람들에게 도움이 될지 적어보세요.

사회적 거리두기를 지켜야 하는 사람

[그림 IV-27] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 공감 단계

② 문제 정의 단계

문제 정의 단계에서는 공감 단계에서 생각한 문제를 명확히 하는 과정으로 [그림 IV-28]과 같이 활동하였다.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

가깝게 접근하면 정보를 알려 줌

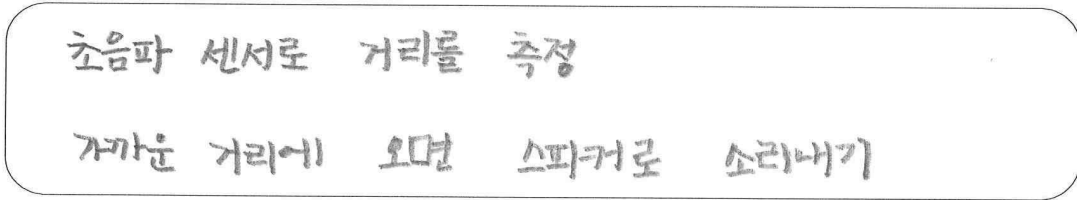
[그림 IV-28] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 문제 정의 단계

③ 아이디어 발상 단계

아이디어 발상 단계에서는 사회적 거리두기를 위해 어떤 센서와 액추에이터를 사용할지 등 문제를 해결하기 위한 다양한 아이디어를 도출하는 과정으로 [그림 IV-29]와 같이 활동하였다.

3. 아이디어 발상 단계 [Ideate]

- 어떤 센서와 액추에이터를 사용하면 좋을지 생각해 보세요.



[그림 IV-29] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 아이디어 발상 단계

④ 시안 만들기 단계

시안 만들기 단계에서는 직접 마이크로비트를 활용하여 사회적 거리두기를 도와주는 프로그램을 작성하는 과정으로 [그림 IV-30]과 같이 활동하였다.



[그림 IV-30] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 시안 만들기 단계

⑤ 검증하기 단계

제작한 앱의 오류 검증, 앱의 장점, 앱에 추가하면 좋을 점에 대하여 알아보는 과정으로 [그림 IV-31]과 같이 활동하였다.

5. 검증하기 단계 [Test]

- 1) 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

핀 번호를 잘못 입력하여 거리가 측정되지 않음
핀 번호를 바꿈.

- 2) 만든 프로그램의 장점을 적어주세요.

가까운 거리에 오면 경보를 알려 거리를
유지할 수 있다.

- 3) 추가하면 좋을 내용을 적어주세요.

좀 더 정밀하게 측정되면 좋을 것 같다.
다양한 액추에이터를 사용하면 좋을 것 같다.

[그림 IV-31] 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 검증하기 단계

(2) 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안

앞서 소개했던 디자인씽킹 기반 소프트웨어 교육을 하기 전에 일반적인 수업에서 가장 많이 사용되는 ‘도입-전개-정리’의 구조를 사용하여 교사용 교수·학습 과정안을 구성하였다. 학생들이 실생활의 문제에 대해 마이크로비트를 활용하여 디자인씽킹 과정을 경험할 수 있도록 과정안을 구성하였다.

<표 IV-29> 마이크로비트 활용 디자인씽킹 기반 소프트웨어 교육 교수·학습 과정안

주제	사회적 거리두기 프로그램	차시	12차시
대상	초등 5-6학년 22명	소요 시간	40분
학습 목표	센서와 액추에이터를 활용하여 사회적 거리두기 프로그램을 만들 수 있다.		
학습 요소	교수·학습 활동		시간
도입	<ul style="list-style-type: none"> ▶ 동기유발 활동 <ul style="list-style-type: none"> - ‘사회적 거리두기 프로그램’ 작품 보여주기 ▶ 공감 단계 <ul style="list-style-type: none"> - 어떤 사람들에게 도움이 될지 떠올리기 		5
전개	<ul style="list-style-type: none"> ▶ 문제 정의 단계 <ul style="list-style-type: none"> - 해결할 문제에 대해 정의하기 		5
	<ul style="list-style-type: none"> ▶ 아이디어 발상 단계 <ul style="list-style-type: none"> - 센서와 액추에이터의 종류 확인하기 - 센서와 액추에이터의 특징 파악하기 - 센서와 액추에이터 동작 원리 찾아보기 		10
	<ul style="list-style-type: none"> ▶ 시안 만들기 단계 <ul style="list-style-type: none"> - 센서값 확인하기 - 조건문을 이용하여 센서값에 따라 동작하는 프로그램 만들기 		15
정리	<ul style="list-style-type: none"> ▶ 검증하기 단계 <ul style="list-style-type: none"> - 작성한 프로그램에 오류 확인하기 - 잘된 점과 개선할 점 생각하기 		5

(3) 마이크로비트 활용 디자인씽킹 기반 문제해결

6개의 그룹을 조직하여 문제해결 과정을 진행하였다. 그룹의 문제해결 주제는 <표 IV-30>과 같다.

<표 IV-30> 마이크로비트 활용 디자인씽킹 기반 문제해결 주제

그룹	해결 주제
1 그룹	만능 분무기
2 그룹	휴대폰 쉽게 찾기
3 그룹	쓰레기통 비우기 알림
4 그룹	움직임을 감지 신호등
5 그룹	시간을 알려주는 IoT
6 그룹	친구와 메시지 주고 받기

교사는 디자인씽킹 기반 문제해결 과정을 진행하는 동안 퍼실리테이터의 역할을 수행하였다. 디자인씽킹 각 단계에서 해결해야 할 일, 보이는 실수, 반복된 도전, 시행착오 등을 열린 마음으로 지켜보고, 도전 정신을 갖도록 격려했다. 디자인씽킹 과정에서 협력이 필요한 상황에서는 CSCL 도구로 패들렛, 구글 프레젠테이션을 사용하였다.

다음 내용은 1그룹의 ‘만능 분무기’ 라는 주제로 디자인씽킹 기반의 문제해결 과정을 간략히 알아보려고 한다.

① 공감 단계

공감 단계에서는 패들렛을 이용하여 수업 시간이 아닌 과제로 제시하였다. 온라인 공간에서 다른 사람들과 인터뷰 또는 조사를 통해 해결해야 할 문제를 찾아내고 브레인스토밍 기법으로 최대한 다양한 의견을 제시하고 비판하지 않도록 하였다.



[그림 IV-32] 마이크로비트 활용 디자인씽킹 공감 단계 활동 장면

② 문제 정의 단계

공감 단계에서 찾은 문제들 중 의견 교환을 통해 해결하고자 하는 문제를 찾고 정의 내리는 단계이다.

문제 정의 단계

저희는 많은 의견중 0000학생의 의견인 식물을 키우기만 하면 계속 식물이 죽어서 속상하다.

(물조절, 햇빛 양 조절, 토양(흙)종류 실패 원인)를 골랐습니다.

왜냐하면 저희 의견중에 많이 발생하는 이유이고 식물을 키우는데 힘들어하는 사람이 많아서이다.

[그림 IV-33] 마이크로비트 활용 디자인씽킹 문제 정의 단계 활동 장면

③ 아이디어 발상 단계

정의된 문제를 어떤 방법으로 해결할 것인지 아이디어를 떠올려 보는 단계이다. 결과물의 형태, 스프라이트의 종류, 스프라이트의 특징들에 대하여 구체화하는 활동을 한다.

아이디어 발상 단계

식물은 빛센서를이용해서 햇빛조절을 한다

습도센서를 이용해서 물을 조절한다

고서연 습도센서를 이용하여 물조절.(서보모터 이용,분무기 뿌리기)

빛센서를 이용하여 햇빛 양 조절

[그림 IV-34] 마이크로비트 활용 디자인씽킹 아이디어 발상 단계 활동 장면

④ 시안 만들기 단계

문제 해결 아이디어를 바탕으로 시안을 만드는 단계이다. 직접 마이크로비트에서 프로그래밍을 제작하는 활동을 한다.

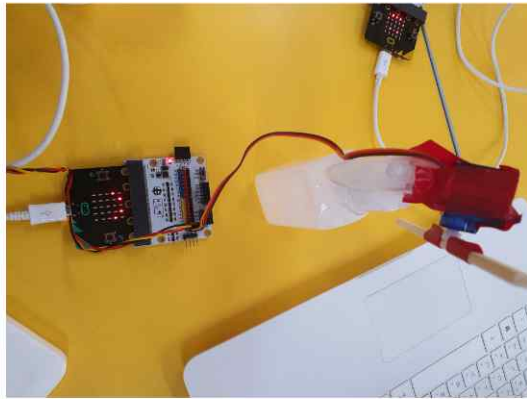


[그림 IV-35] 마이크로비트 활용 디자인씽킹 시안 만들기 단계 활동 장면

⑤ 검증하기 단계

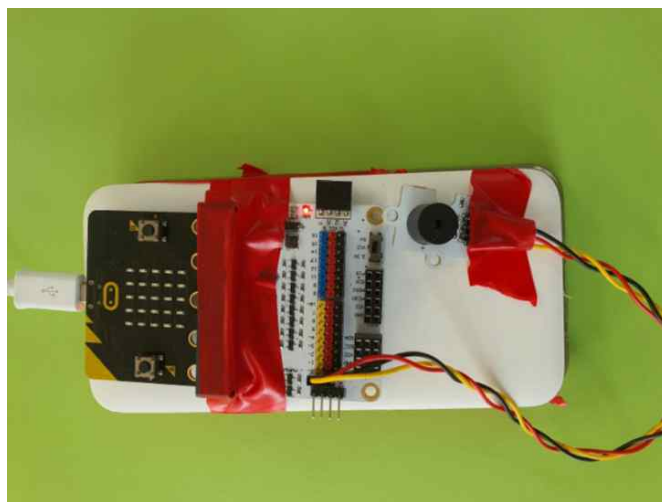
제작한 시안을 발표하는 자리를 마련하고 직접 완성분을 검증해 보는 단계이다. 만든 시안이 잘 동작하는지 알아보고, 잘된 점과 추가하면 좋을 점에 대하여 공유하는 활동을 한다.

검증하기 단계



[그림 IV-36] 마이크로비트 활용 디자인씽킹 검증하기 단계 활동 장면

[그림 IV-37]과 [그림 IV-38]은 그룹별로 제작한 시안들의 예시 작품들이다. 각각의 문제들을 마이크로비트를 이용해서 해결하였다.



[그림 IV-37] 디자인씽킹 기반 2그룹 '휴대폰 쉽게 찾기' 활동 장면

검증하기



[그림 IV-38] 디자인씽킹 기반 3그룹 ‘쓰레기통 비우기 알림’ 활동 장면

4) 연구 결과 분석

(1) 창의성 변화

① 정규성 검정

실험집단의 정규성을 확보하고 있는지 확인하기 위하여 집단의 창의성 사전 검사 결과로 정규성 검정을 실시하였다. 사전 검사 데이터를 바탕으로 비모수/모수 통계를 결정하기 위해 정규성 검정의 방법으로 Shapiro-Wilks 검정을 실시하였고, <표 IV-31>에 그 결과를 제시하였다.

<표 IV-31> 창의성 검사 정규성 검정

하위 요소	사례 수	평균	표준편차	최댓값	최솟값	통계량	p
창의성 지수	22	81.5273	18.91379	114.60	38.20	.968	.570
창의성 평균	22	79.5727	17.38180	107.60	38.20	.962	.323
유창성	22	103.1818	22.44281	145.00	57.00	.962	.888
독창성	22	89.9545	19.42134	125.00	54.00	.886	.406
제목의 추상성	22	51.6364	44.69425	117.00	0.00	.888	.004*
정교성	22	110.4545	24.30898	152.00	79.00	.861	.031*
성급한 종결에 대한 저항	22	42.6364	35.31834	106.00	0.00	.960	.016*

* $p < .05$, ** $p < .01$

창의성 사전 검사에 대한 정규성 검정 결과 ‘제목의 추상성’, ‘정교성’과 ‘성급한 종결에 대한 저항’은 유의확률 p 값이 각각 .004, .031, .016로 95% 신뢰수준 보다 낮게 나타나 정규성이 만족되지 않았다. 나머지 영역에서는 유의확률 p 값이 95% 신뢰수준인 .05보다 높게 나타나 정규분포임이 확인되었다.

② 사전·사후 검사 집단 내 비교

사전·사후 검사 결과 창의성의 변화를 알아보기 위하여 <표 IV-32>와 같이 정규성을 확보한 항목은 모수통계인 대응표본 t 검정을 실시하였고, 정규성을 확보 하지 못한 항목은 <표 IV-33>과 같이 비모수 통계인 Wilcoxon 부호 순위 검정을 실시하였다.

<표 IV-32> 사전·사후 검사 결과 집단 내 비교(대응표본 t 검정)

하위 요소	사례수	사전 검사		사후 검사		t	p
		평균	표준편차	평균	표준편차		
창의성 지수	22	81.5273	18.91379	94.2000	20.65130	-3.499	.002**
창의성 평균	22	79.5727	17.38180	91.5636	18.17519	-3.563	.002**
유창성	22	103.1818	22.44281	114.4545	20.69674	-2.701	.013*
독창성	22	89.9545	19.42134	100.8182	20.32975	-2.035	.055

* $p < .05$, ** $p < .01$

<표 IV-33> 사전·사후 검사 결과 집단 내 비교(Wilcoxon 부호 순위 검정)

하위 요소	사례수	사전 검사		사후 검사		z	p
		평균	표준편차	평균	표준편차		
제목의 추상성	22	51.6364	44.69425	73.5909	39.13671	-2.505	.012*
정교성	22	110.4545	24.30898	116.4545	18.09427	-1.924	.054
성급한 종결에 대한 저항	22	42.6364	35.31834	52.5000	36.16595	-1.854	.064

* $p < .05$, ** $p < .01$

실험집단의 창의성 사전·사후 검사 결과를 비교·분석한 결과 ‘창의성 지수’는 사전 검사 평균 81.5273에서 사후 검사 평균 94.2000으로 12.6727만큼 증가하였으며, 검증통계량 t 값이 -3.499 , 유의확률 p 값이 $.002$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘창의성 평균’에서는 사전 검사 평균 79.5727에서 사후 검사 평균 91.5636으로 11.9909만큼 증가하였으며, 검증통계량 t 값이 -3.563 , 유의확률 p 값이 $.002$ 로 99% 신뢰수준에서 유의미한 차이를 보였다. ‘유창성’에서는 사전 검사 평균 103.1818에서 사후 검사 평균 114.4545로 11.2727만큼 증가하였으며, 검증통계량 t 값이 -2.701 , 유의확률 p 값이 $.013$ 으로 95% 신뢰수준에서 유의미한 차이를 보였다. ‘제목의 추상성’에서는 사전 검사 평균 51.6364에서 사후 검사 평균 73.5909로 21.9545만큼 증가하였으며, 검증통계량 z 값이 -2.505 , 유의확률 p 값이 $.012$ 로 95% 신뢰수준에서 유의미한 차이를 보였다.

‘독창성’에서는 사전 검사 평균 89.9545에서 사후 검사 평균 100.8182로 10.8637만큼 증가하였으나, 검증통계량 t 값이 -2.035 , 유의확률 p 값이 $.055$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다. ‘정교성’에서는 사전 검사 평균 110.4545에서 사후 검사 평균 116.4545로 6.0000만큼 증가하였으나, 검증통계량 z 값이 -1.924 , 유의확률 p 값이 $.054$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다. ‘성급한 종결에 대한 저항’에서는 사전 검사 평균 42.6364에서 사후 검사 평균 52.5000으로 9.8636만큼 증가하였으나, 검증통계량 z 값이 -1.854 , 유의확률 p 값이 $.064$ 로 95% 신뢰수준에서 유의미한 차이를 보이지 않았다.

(2) 창의성 검사 결과의 분석

먼저, 프로그램 적용 전 실험 집단이 정규분포를 갖추었는지 확인하기 위하여 창의성 사전 검사에 대해 실시한 Shapiro-Wilks 정규성 검정 결과 창의성 하위 요소 중 ‘제목의 추상성’, ‘정교성’, ‘성급한 종결에 대한 저항’을 제외한 ‘창의성 지수’, ‘창의성 평균’, ‘유창성’, ‘독창성’은 정규분포를 갖춘 표본임이 검증되었다.

창의성 부분은 정규성을 확보한 창의성 요소들은 사전·사후 집단 내 대응표본 t 검정을 실시하였고, 정규성을 확보하지 못한 창의성 하위 요소에 대해서는 비모수 통계방법인 Wilcoxon 부호 순위 검정을 실시하였다.

검증 결과, 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육이 초등학

생의 창의성 신장에 효과적인 것으로 나타났다. 이는 CSCL를 활용한 디자인씽킹 기반 협력학습을 통해 주변의 문제를 찾고 다양한 아이디어를 도출하며 이를 동료 학습자들과 협업하여 문제를 해결하는 과정이 창의성에 영향을 준 것으로 판단된다.

특히, 창의성의 하위 요소 중 ‘유창성’, ‘제목의 추상성’을 신장시키는데 효과적이었다. 마이크로비트라는 교육 도구의 특성상 가상 공간에서 해결을 하는 것이 아니라 실물 공간에서 직접 만들어 보는 활동이 ‘유창성’ 신장에 도움을 준 것으로 판단되고, 구체적인 산출물을 도출하기 위하여 핵심 정보, 즉 센서 및 액추에이터의 역할이나 기능들을 정확하게 파악하여 구성하는 과정이 ‘제목의 추상성’ 신장에 도움을 준 것으로 생각된다.

‘독창성’, ‘정교성’, ‘성급한 종결에 대한 저항’에서도 95% 신뢰수준에서 유의미하지는 않았으나 신뢰수준에 근접한 유의확률을 보여서 평균의 상승에 영향을 미쳤다고 볼 수 있을 것이다.

4. 디자인씽킹 기반 소프트웨어 교육 프로그램 창의성 결과 비교 분석

적용 단계에서 디자인씽킹 기반 소프트웨어 교육 도구에 따라 개발된 3가지 디자인씽킹 기반 소프트웨어 교육 프로그램을 초등 예비교사, 초등학교 4-6학년, 5-6학년을 대상으로 적용하였다. 창의성 검증 결과를 종합하여 비교하면 <표 IV-34>와 같다.

<표 IV-34> 디자인씽킹 기반 소프트웨어 교육 프로그램 도구별 창의성 결과 비교

하위 요소	디자인씽킹 기반 소프트웨어 교육 도구	검정 도구	대상	평균 변화량	t(z)	p
창의성 지수	스크래치	대응표본 t 검정	초등 예비교사 (대학교 1학년)	+ 7.5909	-4.415	.000**
	앱인벤터		초등학교 4~6학년	+15.7429	-4.709	.000**
	마이크로비트		초등학교 5~6학년	+12.6727	-3.499	.002**
창의성 평균	스크래치	대응표본 t 검정	초등 예비교사 (대학교 1학년)	+ 7.1363	-5.024	.000**
	앱인벤터		초등학교 4~6학년	+15.5643	-4.787	.000**
	마이크로비트		초등학교 5~6학년	+11.9909	-3.563	.002**
유창성	스크래치	대응표본 t 검정	초등 예비교사 (대학교 1학년)	+ 8.1818	-1.954	.064
	앱인벤터	Wilcoxon 부호 순위 검정	초등학교 4~6학년	+10.5715	-2.733	.006**
	마이크로비트	대응표본 t 검정	초등학교 5~6학년	+11.2727	-2.701	.013*
독창성	스크래치	Wilcoxon 부호 순위 검정	초등 예비교사 (대학교 1학년)	+15.0909	-3.559	.000**
	앱인벤터	대응표본 t 검정	초등학교 4~6학년	+11.8929	-3.045	.005**
	마이크로비트		초등학교 5~6학년	+10.8637	-2.035	.055

제목의 추상성	스크래치	대응표본 t 검정	초등 예비교사 (대학교 1학년)	- 0.2728	.060	.952
	앱인벤터	Wilcoxon 부호 순위 검정	초등학교 4~6학년	+14.4642	-1.660	.097
	마이크로비트		초등학교 5~6학년	+21.9545	-2.505	.012*
정교성	스크래치	Wilcoxon 부호 순위 검정	초등 예비교사 (대학교 1학년)	+13.4091	-3.169	.002**
	앱인벤터		초등학교 4~6학년	+11.7143	-2.677	.007**
	마이크로비트		초등학교 5~6학년	+ 6.0000	-1.924	.054
성급한 종결에 대한 저항	스크래치	Wilcoxon 부호 순위 검정	초등 예비교사 (대학교 1학년)	- 0.7727	- .552	.581
	앱인벤터		초등학교 4~6학년	+31.6786	-3.710	.000**
	마이크로비트		초등학교 5~6학년	+ 9.8636	-1.854	.064

* $p < .05$, ** $p < .01$

사전·사후 검사 결과 창의성의 변화를 각 도구별로 비교해 보면 ‘창의성 지수’와 ‘창의성 평균’은 스크래치, 앱인벤터, 마이크로비트 모든 프로그램에서 99% 신뢰수준에서 유의미한 변화를 보였다. 특히 앱인벤터를 활용했을 때 평균 변화량은 15정도로 가장 많이 상승한 것으로 나타났다. 이는 앱인벤터를 활용한 디자인씽킹 소프트웨어 교육 프로그램을 적용했을 때 창의성 하위 요소 대부분이 상승한 것으로 나타난 영향으로 분석된다. 스마트폰에 대한 관심도 많고 직접 사용해 보면서 필요한 앱에 대한 생각을 많이 해 본 것 같다. 그런 만큼 주변의 문제 해결을 위해 창의적인 생각을 많이 한 것으로 판단된다.

‘유창성’은 앱인벤터와 마이크로비트를 적용했을 때 95% 신뢰수준에서 유의미한 변화를 보였고, 마이크로비트를 활용했을 때 평균 변화량은 11.2727로 가장 상승한 것으로 나타났다. 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램을 적용했을 때 가상 공간에서 해결을 하는 것이 아니라 실물 공간에서 직접 만들어 보는 활동이 다양한 아이디어를 창출하게 하였고 ‘유창성’ 신

장에 도움을 준 것으로 판단된다.

‘독창성’은 스크래치와 애플벤티를 적용했을 때 99% 신뢰수준에서 유의미한 변화를 보였고, 스크래치를 활용했을 때 평균 변화량은 15.0909로 가장 크게 상승한 것으로 나타났다. 가상 공간에서 프로젝트를 제작하다 보니 실물 공간에 제약 없이 해결 방법을 제시하였다. 예를 들면 애플벤티나 마이크로비트는 앱 또는 실물의 완성품을 만드는 활동이라면 스크래치에서는 스토리텔링을 만들고 대사를 넣는 등 일련의 과정을 표현하는 활동이었다. 이런 활동이 독특한 아이디어를 산출하는 능력인 ‘독창성’ 신장에 도움이 된 것으로 생각된다.

‘제목의 추상성’은 마이크로비트를 적용했을 때 95% 신뢰수준에서 유의미한 변화를 보였고, 마이크로비트를 활용했을 때 평균 변화량은 21.9545로 가장 크게 상승한 것으로 나타났다. ‘제목의 추상성’은 문제를 종합하고 조직화할 줄 아는 사고 과정과 관련이 있다. 가장 높은 수준에서 관련 정보의 핵심을 포착해 내고, 무엇이 중요한 것인지를 알 수 있는 능력이라 볼 수 있다. 마이크로비트 활동은 실세계에서 구체적인 산출물을 도출하기 위하여 핵심 정보, 즉 센서 및 액추에이터의 역할이나 기능들을 정확하게 파악하고 구성하여 해결해야 하기 때문에 ‘제목의 추상성’에 영향을 준 것으로 판단된다.

‘정교성’은 스크래치와 애플벤티를 적용했을 때 99% 신뢰수준에서 유의미한 변화를 보였고, 그 중 스크래치의 평균 변화량이 13.4091로 가장 높았다. 스크래치를 활용한 디자인씽킹 문제 해결에서는 코딩이 복잡해 질 때가 많았다. 또한 스크래치는 객체 지향적인 언어이기 때문에 객체들마다의 정교한 작업이 필요하고 정교한 작업이 이루어지지 않을 경우 오류가 많이 발생할 수 있다. 이러한 영향으로 ‘정교성’ 신장에 도움이 된 것으로 생각된다.

‘성급한 종결에 대한 저항’은 애플벤티를 적용했을 때 99% 신뢰수준에서 유의미한 변화를 보였고, 평균 변화량은 31.6786으로 가장 높은 변화를 가져왔다. ‘성급한 종결에 대한 저항’은 독창적인 아이디어를 가능하게 하기 위해 성급하게 반응을 폐쇄하고 종결시키는 것을 지연시킬 줄 아는 능력이다. 문제 해결 산출물에 대해 스크래치와 마이크로비트는 프로그램에 접근했을 때만 확인이 가능하였다. 하지만 애플벤티는 설치한 앱이 항상 스마트폰에 저장되어 있어 사용해 보고 앱의 불편한 점들을 계속 확인하는 과정을 겪게 되었다. 이런 과정은 산출물에 대

하여 끈임없이 발전시키려는 노력으로 나타났다. 이것이 ‘성급한 종결에 대한 저항’에 영향을 준 것으로 판단된다.

이와 같은 연구 결과를 종합하면 다음과 같은 결과를 알 수 있다.

첫째, 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 창의성 하위 요소 중 ‘독창성’, ‘정교성’을 신장시키는데 효과적이다. 가상 공간에서 프로젝트를 제작하다 보니 실물 공간에 제약을 받지 않고 해결 방법을 제시하는 것이 ‘독창성’을 신장시키는 데 도움이 된 것으로 판단되며, 코딩의 길이가 길고 객체 지향적인 언어이기 때문에 정교한 작업이 필요한 것이 ‘정교성’에 영향을 준 것으로 분석된다.

둘째, 앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 창의성 하위 요소 중 ‘유창성’, ‘독창성’, ‘정교성’, ‘성급한 종결에 대한 저항’을 신장시키는데 효과적이다. 창의성의 요소별 신장 정도를 비교하면 앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램의 효과가 가장 높았다. 이는 학생들이 스마트폰에 대한 관심도 많고 직접 사용해 보면서 필요한 앱에 대한 생각을 많이 해 본 것 같다. 그런 만큼 주변의 문제 해결을 위해 창의적인 생각을 많이 한 것으로 판단된다.

셋째, 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램은 ‘유창성’, ‘제목의 추상성’을 신장시키는데 효과적이다. 가상 공간에서 해결을 하는 것이 아니라 실물 공간에서 직접 만들어 보는 활동이 ‘유창성’ 신장에 도움을 준 것으로 판단되고, 구체적인 산출물을 도출하기 위하여 핵심 정보, 즉 센서 및 액추에이터의 역할이나 기능들을 정확하게 파악하여 구성하는 과정이 ‘제목의 추상성’ 신장에 도움을 준 것으로 생각된다.

본 연구를 통해 창의성 각 요소들을 신장시키는데 가장 효과적인 도구는 4~6 학년을 대상으로 적용한 앱인벤터로 나타났다. 그러나, 다른 도구들도 적용한 학년에서 ‘창의성의 지수’ 및 ‘창의성 평균’이 유의미하게 신장되었고, 각 부분에서 도구별 특성에 맞게 창의성 요소가 신장된 것으로 나타났다.

V. 결론 및 제언

본 연구에서는 초등학생을 대상으로 창의성 신장을 위해 ADDIE 모형에 따라 디자인씽킹 기반 소프트웨어 교육 프로그램을 설계·개발하고 이를 현장에 적용하여 교육적 효과를 검증하였다.

분석 단계에서는 창의성, 디자인씽킹 기반 소프트웨어 교육, CSCL 등에 대한 선행연구 분석과 Rossett 요구 분석 모형에 맞춰 초등학생과 교사를 대상으로 학습자 요구 분석을 진행하였다.

설계 단계에서는 성취 목표 명세화, 프로그램 구조 설계, 학습 내용 설계, 교수 전략 설계, 교육 프로그램 개발 원리, 교육 도구 선정 과정을 통해 교육 프로그램을 설계하였다.

개발 단계에서는 설계 단계의 내용을 바탕으로 3가지의 교육 도구에 따라 디자인씽킹 기반 소프트웨어 교육 프로그램을 개발하였다.

적용 단계에서는 디자인씽킹 기반 소프트웨어 교육 도구에 따라 개발된 3가지의 디자인씽킹 기반 소프트웨어 교육 프로그램을 총 3회에 걸쳐 초등 예비교사와 초등학생에게 적용하여 프로그램의 교육적 효과를 검증하였다. 디자인씽킹 과정에서 협력이 필요한 상황에서는 CSCL 도구로 페들렛, 구글 프레젠테이션을 사용하였다. 3가지의 교육 프로그램은 ‘스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’으로 구성하였다.

평가 단계에서는 창의성 검사 도구로 TTCT 검사지 도형 A, B형을 사용하였다. 평가 단계를 통해 분석된 본 연구의 ‘스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘앱인벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’, ‘마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 프로그램’의 학습 효과는 모두 초등학생의 창의성 신장에 효과적인 것으로 나타났다.

본 연구의 시사점은 다음과 같다.

첫째, 디자인씽킹 기반 소프트웨어 교육 프로그램은 창의성 신장에 효과적이

다. 공감 단계, 문제 정의 단계, 아이디어 발상 단계, 시안 만들기 단계, 검증하기 단계로 5가지 단계에 따라 학생 활동 중심의 디자인씽킹 기반 소프트웨어 교육 문제 해결 과정을 통해 창의성이 신장되었다.

둘째, 디자인씽킹 기반 소프트웨어 교육 프로그램의 도구로 활용된 스크래치, 애플벤터, 마이크로비트는 창의성의 하위 요소의 특징에 따라 구분하여 사용할 수 있음을 확인하였다. 스크래치는 ‘독창성’과 ‘정교성’, 애플벤터는 ‘유창성’, ‘독창성’, ‘정교성’, ‘성급한 종결에 대한 저항’, 마이크로비트는 ‘유창성’, ‘제목의 추상성’ 신장에 효과적이었다.

셋째, 디자인씽킹 기반 소프트웨어 교육 프로그램을 적용할 수 있는 교육 교재와 교수·학습 과정안 개발을 통해 초등학생, 예비 초등교사를 대상으로 한 디자인씽킹 기반 소프트웨어 교육을 일반화할 수 있는 가능성을 제시하였다.

본 연구의 결과 다음과 같이 제언하고자 한다.

첫째, 창의성 신장을 위한 디자인씽킹 기반 소프트웨어 교육 프로그램이 다양한 도구에 따른 교육 콘텐츠 개발이 필요하다. 디자인씽킹 기반 소프트웨어 교육 프로그램은 문제를 해결하는 과정을 통해 창의성 신장에 효과적이므로 본 연구에서 제시한 스크래치, 애플벤터, 마이크로비트 뿐만 아니라 다양한 도구들을 활용한 콘텐츠 개발이 요구된다.

둘째, 디자인씽킹 기반 소프트웨어 교육 프로그램이 다양한 대상에 대한 교육 콘텐츠 개발이 필요하다. 본 연구에서는 초등학생과 예비 초등교사들을 대상으로 디자인씽킹 기반 소프트웨어 교육 프로그램을 적용하여 창의성 신장에 대한 효과성을 검증하였다. 따라서 중학생, 고등학생 등 다양한 대상에 대한 교육 콘텐츠 개발이 요구된다.

셋째, 디자인씽킹 기반 소프트웨어 교육 프로그램을 학교 현장에서 쉽게 이해할 수 있고 적용 가능한 안내 자료가 개발되어야 한다. 현재 소프트웨어 교육의 중요성이 강조되고 있는 시대적 상황에 맞추어 디자인씽킹 기반 소프트웨어 교육 프로그램을 일반 교사들도 쉽게 가르칠 수 있는 안내 자료가 필요하다.

향후 후속 연구를 통해 본 연구에서 개발한 디자인씽킹 기반 소프트웨어 교육 프로그램을 지속적으로 개선하여 대상을 확대 적용해 간다면 창의성 신장을 위한 일반화된 또 하나의 소프트웨어 교육 프로그램이 될 것이다.

참 고 문 헌

- 강명희, 홍경선. (1998). 온라인 교육에서 문제중심학습 원리를 적용할 때에 나타나는 수행과정의 특성. *교육정보미디어연구*, 4, 18-40.
- 교육부. (2015). *소프트웨어 지향적 사회를 위한 인재 육성 방안*. 교육부.
- 김경은. (2020). 유아의 창의성 증진을 위한 디자인씽킹 프로그램 개발 연구. *학습자중심교과교육연구*, 20(13), 1023-1052.
- 김동식, 강인구. (2004). CSCL에서 SN(Social Network)데이터의 제시가 상호작용 과정과 활동성도에 미치는 영향. *교육공학연구*, 20(1), 89-115.
- 김병수. (2014). *계산적 사고력 신장을 위한 PPS기반 프로그래밍 교육 프로그램*. 박사학위논문, 제주대학교 대학원.
- 김선연. (2019). 디자인 씽킹에 기반한 집단 창의성 사고과정 모형 개발. *교육공학연구*, 35(3), 621-653.
- 김선연. (2020). 디자인 씽킹에 기반한 집단 창의성 교육목표 분류체계 프로토타입 개발. *학습자중심교과교육연구*, 20(13), 277-304.
- 김영채. (1999). *창의적 문제해결: 창의력의 이론 개발과 수업*. 교육과학사.
- 김영채. (2010). *Torrance TTCT (도형) 검사 A&B형*. 창의력 한국 FPSP.
- 김용민. (2018). *컴퓨팅 사고력과 창의성 향상을 위한 데이터 과학 교육 프로그램*. 박사학위논문, 제주대학교 대학원.
- 김윤미. (2013). *CPS 활용 미술프로그램이 유아의 창의성과 조형미술 표현력에 미치는 효과*. 석사학위논문, 숭실대학교 대학원.
- 김정아. (2021). *컴퓨팅 사고력과 창의성 신장을 위한 데이터 시각화 기반 컴퓨터 교육 프로그램*. 박사학위논문, 제주대학교 대학원.
- 김종진. (2011). *EPL을 이용한 창의성 증진 교육 프로그램 개발 및 적용에 관한 연구:로고와 스크래치를 중심으로*. 박사학위논문, 홍익대학교 대학원.
- 김태우, 이영주. (2019). CSCL 환경에서 초등학생을 위한 통합 지원 CSCL 스크립트 개발 및 활용 효과. *교육정보미디어연구*, 25(3), 499-524.
- 김혜숙. (1999). 창의성 진단 측정도구의 개발 및 타당화. *교육심리연구*, 13(4),

269-303.

- 김효숙, 김동식, 권성호. (2009). CSCL에서 내재적 논증 수준과 외현적 스크립트 제시 전략이 논증 능력에 미치는 영향. **교육공학연구**, 25(2), 247-270.
- 류은진, 김민정. (2021). 디자인씽킹을 활용한 공학제품 설계수업에서의 창의적 문제해결 경험 연구. **The Journal of the Convergence on Culture Technology (JCCT)**, 7(1), 223-233.
- 박정환, 우옥희. (1999). PBL(Problem-Based Learning)이 학습자의 메타인지 수준에 따라 문제해결 과정에 미치는 효과. **교육공학연구**, 15(3), 55-81.
- 박주연, 정혜영, 김성희, 이영미, 이유경, 윤혜선, 표지은, 이재호, 조수빈, 이원경, 이정아, 김은비. (2020). 디자인 사고 과정에 기반한 초등 해커톤의 개발 및 적용. **교육과학연구**, 51(1), 159-185.
- 배성환. (2017). **처음부터 다시 배우는 서비스 디자인 씽킹**. 한빛미디어.
- 배예나, 박혜진. (2019). 디자인 사고 교육을 통한 성인학습자들의 창의적 사고 경험에 관한 연구. **교육정보미디어연구**, 25(2), 299-324.
- 변현정. (2015). 디자인 사고과정(Design Thinking Process) 경험이 대학생 창의성 개발에 미치는 영향. **창의력교육연구**, 15(3), 149-167.
- 서영호, 김관민, 김종훈. (2021). 창의성과 컴퓨팅 사고력 향상을 위한 디자인 사고 적용 피지컬 컴퓨팅 프로그램 개발 및 적용 : 마이크로비트를 중심으로. **정보교육학회논문지**, 25(2), 377-385.
- 서영호, 김종훈. (2017). 디자인 사고를 적용한 SW교육이 초등학교 예비교사의 창의성에 미치는 효과. **정보교육학회논문지**, 21(3), 351-360.
- 서영호, 김종훈. (2018). CSCL기반 디자인 사고를 적용한 SW교육이 초등 학생의 창의성과 문제해결능력에 미치는 효과. **정보교육학회논문지**, 22(4), 427-438.
- 서응교. (2017). 플립러닝과 디자인씽킹에 기반을 둔 창의적사고 강화와 코딩교육을 위한 강좌개발. **학습자중심교과교육연구**, 17(16), 173-199.
- 서응교, 전은화, 정효정. (2016). 대학생 창의역량 개발을 위한 디자인씽킹 기반 강좌 개발. **학습자중심교과교육연구**, 16(4), 693-718.
- 송가영, 정은영. (2020). 디자인 씽킹 기반 융합 수업이 창의성에 미치는 영향. **융**

합교육연구, 6(2), 71-95.

- 송동주, 박재호, 강상희. (2016). 디자인씽킹. 페가수스.
- 송미연, 황재원. (2020). 디자인사고 과정을 적용한 디자인 교육 프로그램이 유아의 창의성과 정서지능에 미치는 영향. **유아교육연구**, 40(1), 241-265.
- 송영은, 이정현. (2018). 창의적 사고력 향상을 위한 디자인창의융합교육 모델 개발 및 검증. **브랜드디자인학연구**, 16(4), 241-254.
- 송정범, 조성환, 이태욱. (2008). 스크래치 프로그래밍 학습이 학습자의 동기과 문제해결력에 미치는 영향. **한국정보교육학회지**, 12(3), 323-332.
- 시지현. (2014). 집단작동기억효과를 위한 CSCL 스크립트가 협력학습효율성에 미치는 영향. **학습과학연구**, 8(3), 110-133.
- 신계옥. (2018). 디자인 씽킹에 의한 자율 학습 가이드 개발 연구-창의적 실기 프로젝트 중심으로. **커뮤니케이션디자인학연구**, 62, 44-57.
- 신윤희, 정효정, 서응교. (2019). 비공학계열 학생을 위한 디자인 씽킹 기반 코딩교육 프로그램 효과분석 및 지원전략 연구. **학습자중심교과교육연구**, 19(10), 351-373.
- 신윤희, 정효정, 송종숙. (2019). SW 비전공 대학생을 위한 디자인씽킹 기반 코딩교육에서의 학습경험 분석. **디지털콘텐츠학회논문지**, 20(4), 759-768.
- 안경미, 손원성, 최윤철. (2010). 스크래치 프로그래밍 교육이 초등학생의 학습 몰입과 프로그래밍 능력에 미치는 효과. **정보교육학회논문지**, 15(1), 1-10.
- 양경란, 문석찬, 한규하, 박소현. (2015). 디자인 씽킹, 새로운 패러다임의 전환. **Entrue Journal of Information Technology**, 14(3), 7-24.
- 오정철. (2020). **컴퓨팅 사고력과 창의성 신장을 위한 퍼즐 기반 컴퓨터 교육 프로그램**. 박사학위논문, 제주대학교 대학원.
- 우영진, 윤지현, 강성주. (2016). 초등 과학영재의 집단 창의성 발현을 돕는 전략으로써 디자인적 사고의 가능성 탐색. **학습자중심교과교육연구**, 16(7), 433-460.
- 우영진, 윤지현, 강성주. (2018). 공감 중심 디자인적 사고 프로그램 적용을 통한 초등 과학영재학생의 집단 창의성 함양 연구. **영재교육연구**, 28(2), 259-283.

- 원연정, 배예나, 최윤영, 안미리. (2019). 대학생 대상의 디자인 사고 교육이 창의성역량에 미치는 영향. **교육공학연구**, 35(2), 193-226.
- 이도현, 윤지현, 강성주. (2016). 중등 과학 영재들의 집단 창의성을 돕기 위한 전략으로서 디자인적 사고 프로그램의 가능성 탐색. **현장과학교육**, 10(2), 151-171.
- 이미정. (2011). **창의성 향상을 위한 글쓰기 교육 연구: 동화 바꾸어 쓰기를 중심으로**. 석사학위논문, 한남대학교.
- 이선영, 윤지현, 강성주. (2018). 초등학생들의 집단 창의성 개발 전략으로서 디자인씽킹 기반 문제 해결 프로그램의 가능성 탐색. **학습자중심교과교육연구**, 18(1), 525-554.
- 이영덕, 정원식. (1969). **표준화 창의성 검사**. 코리안테스팅센터.
- 이영민. (2005). 웹기반 팀학습 환경에서 공유정신모형이 과제수행에 미치는 영향. **교육공학연구**, 20(2), 53-69.
- 이재무. (2014). Rossett 모형을 적용한 적응형 이러닝 시스템을 위한 요구 분석. **한국콘텐츠학회논문지**, 14(6), 529-538.
- 이재용. (2012). **디자인 사고(Design Thinking)란?**. <http://story.pxd.co.kr/585>
- 이지선. (2015). 컴퓨터적 사고를 기반으로 한 컴퓨터 교육에 디자인적 사고 적용에 관한 연구. **한국디자인문화학회지**, 21(1), 455-467.
- 장은정. (2004). 웹기반 프로젝트 중심 학습에서 목표구조 방식과 팀 효과성이 학습결과에 미치는 영향. **교육공학연구**, 20(2), 53-81.
- 전경원. (2000). **유아 종합 창의성 검사**. 학지사.
- 정재삼. (1998). 교육 프로그램 개발 모형의 분석. **교과교육학연구**, 2(2), 80-97.
- 정효정, 김동식. (2006). CSCL에서 WOE의 유형이 지식 공유 과정에 미치는 영향. **교육공학연구**, 22(3), 23-56.
- 최윤석. (2019). **언플러그드 기반 프로그래밍 언어 교육이 초등학생의 창의성에 미치는 영향**. 석사학위논문, 경인교육대학교 대학원.
- 최형신, 김미송. (2017). 디자인 사고와 컴퓨팅 사고를 결합한 새로운 교사 교육 코스 설계. **정보교육학회논문지**, 21(3), 343-350.
- 허경철. (1991). **사고력 신장을 위한 프로그램 개발 연구(V)**. 한국교육개발원.

- 홍정순. (2020). 초등학생을 위한 디자인씽킹기반 학습프로그램 개발 및 적용에 관한 실험연구. 박사학위논문, 동국대학교 대학원.
- 홍정순, 장환영. (2018). 초등학교 ‘디자인씽킹(Design Thinking)’ 기반 학습프로그램의 개발 및 가능성 탐색. *학습자중심교과교육연구*, 18(22), 1309-1337.
- Adams, R., Daly, S., Mann, L., & Dall’Alba, G. (2011). Being a professional: Three lenses into design thinking, acting, and being. *Design Studies*, 32(6), 588-607.
- Amabile, T. (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, 45(2), 357 - 376.
- Anderson, N. (2012). Design thinking: Employing an effective multidisciplinary pedagogical framework to foster creativity and innovation in rural and remote education. *Australian and International Journal of Rural Education*, 22(2), 43-52.
- Azmitia, M. (1988). Peer interaction and problem solving. When are two heads better than one?. *Child Development*, 59, 87 - 96.
- Barron, B., Schwartz, D., Vye, N., Moore, A., Petrosino, A., Zech, L., Bransford, J., & The Cognition and Technology Group at Vanderbilt (1998). Doing with understanding: lessons from research on problem- and project-based learning. *The Journal of the Learning Sciences*, 7(3), 271-311.
- Benson, J., & Dresdow, S. (2015). Design for thinking: Engagement in an innovation project. *Decision Sciences Journal of Innovative Education*, 13(3), 377-410.
- Björgvinsson, E., Ehn, P., & Hillgren, P. (2012). Design things and design thinking: Contemporary participatory design challenges. *Design Issues*, 28(3), 101-116.
- Booth, C. (2011). *Reflective teaching, effective learning: Instructional literacy for library educators*. American Library Association.

- Brown, T. (2008). Design thinking. *Harvard Business Review*, 86(6), 84 - 92.
- Brown, T., & Wyatt, J. (2010). Design thinking for social innovation. *Development Outreach*, 12(1), 29-43.
- Bucciarelli, L. (1996). *Designing engineers*. Cambridge: MIT Press.
- Buchanan, R. (1992). Wicked problems in design thinking. *Design Issues*, 8(2), 5 - 21.
- Campbell, K. (2015). Design thinking and the deanly conversation: Reflections on conversation, community, and agency. *Journal of Learning Design*, 8(3), 42-45.
- Carroll, M., Goldman, S., Britos, L., Koh, J., Royalty, A., & Hornstein, M. (2010). Destination, imagination and the fires within: Design thinking in a middle school classroom. *International Journal of Art & Design Education*, 29(1), 37-53.
- Chu, S., Quek, F., Bhangaonkar, S., Ging, A., & Sridharamurthy, K. (2015). Making the Maker: A Means-to-an-Ends approach to nurturing the Maker mindset in elementary-aged children. *International Journal of Child-Computer Interaction*, 5, 11-19.
- Coleman, M. (2016). Design thinking and the school library. *Knowledge Quest*, 44(5), 62-68.
- Cross, A. (2017). *Tinkering in k-12: an exploratory mixed methods study of makerspaces in schools as an application of constructivist learning*. Pepperdine University.
- Cross, N. (2006). *Designerly ways of knowing*. Springer.
- Csikszentmihalyi, M. (2003). *Creativity: Flow and the psychology of discovery and invention*. Book Rord.
- Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macroscripts in Computer-Supported Collaborative Learning. *International Journal of Computer-Supported Collaborative Learning*, 3(1). 5-23.
- Dillenbourg, P., Hong, F. & Brahm, T. (2009). *The Many scripts Pedagogical*

- Handbook: How to build script for collaborative learning?*. Swiss Centre for Innovations in learning.
- Dillenbourg, P., Jermann, P., Weinberger, A., Stegman, K., & Fischer, F. (2004). WP23-Mobile Support for Integrated Learning. A framework for integrated learning scripts. *Kaleidoscope - JEIRP MOSIL-Deliverable, 23(4-1)*, 1 - 31.
- Dobbs, R., Manyika, J., & Woetzel, J. (2016). *No Ordinary Disruption: The Four Forces Breaking all the Trends*. PublicAffairs.
- Dowling, C. (2012). The hand: Kinesthetic creation and the contemporary classroom. *International Journal of Learning, 18(8)*, 51-66.
- Dunne, D., & Martin, R. (2006). Design thinking and how it will change management education: An interview and discussion. *Academy of Management Learning & Education, 5(4)*, 512-523.
- Feldhusen, J. (1983). *The purdue creative thinking program*. In I. S. Sato (Ed.), *Creativity research and educational planing*. National State Leadership Training Institute for the Gifted and Talented.
- Gold, J. (1981). *Developing the creative problem solving skills of intermediate age educable mentally retarded student*. Doctoral dissertation, Fordham University, Retrieved September 13, 2021, from <https://research.library.fordham.edu/dissertations/AAI8109076>
- Guilford, J. (1956). Structure of intellect. *Psychological Bulletin, 53(4)*, 267-293, from <https://doi.org/10.1037/h0040755>
- Guilford, J. (1967). *The nature of human intelligence*. McGraw-Hill.
- Hall, R., & Nemirovsky, R. (2011). Introduction to the special issue: Modalities of body engagement in mathematical activity and learning. *Journal of the Learning Sciences, 21(2)*, 207 - 215.
- IDEO., & Riverdale. (2012). *Design Thinking for Educators*. Retrieved September 13, 2021, from <https://www.ideo.com/post/design-thinking-for-educators>

- Johnson, D., Johnson, R., Holubec, E., & Roy, P. (1984). *Circles of learning. Alexandria*. Association for Supervision and Curriculum Development.
- Jonassen, D. (1988). Designing structured hypertext and structuring access to hypertext. *Educational Technology*, 28(11), 13-16.
- Kangas, K. Seitamaa-Hakkarainen, P., & Hakkarainen, K. (2013). Design thinking in elementary students' collaborative lamp designing process. *Design and Technology: An International Journal*, 18(1), 30 - 43.
- Kimbell, L., & Street, P. (2009). *Beyond design thinking: Design-as-practice and designs-in-practice*. CRESC Conference, 1-15, Manchester.
- Koh, J., Chai, C., Wong, B., & Hong, H. (2015). *Design thinking for education: Conceptions and applications in teaching and learning*. Springer.
- Kolko, J. (2010) Abductive Thinking and Sensemaking: Drivers of Design Synthesis. *Design Issues*, 26, 15 - 28.
- Kollar, I., Fischer, F., & Hesse, F. (2006). Collaboration Scripts - A Conceptual Analysis. *Educational Psychology Review*, 18(2), 159-185.
- Kolodner, J., Camp, P., Crismond, D., Fasse, B., Gray, J., Holbrook, J., & Ryan, M. (2009). Problem-based learning meets case-based reasoning in the middle school science classroom: Putting Learning by Design TM into practice. *Journal of the Learning Sciences*, 12(4), 495 - 547.
- Lande, M., & Leifer, L. (2010). *Incubating engineers, hatching design thinkers: Mechanical engineering students learning design through ambidextrous ways of thinking*. Proceedings of the Annual American Society for Engineering Education Conference, Louisville.
- Lawson, B., & Dorst, K. (2009). *Design Expertise*. Architectural Press.
- Lawson, B. (1990). *How designers think: the design process demystified*. Routledge.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: a sneak preview. *Connecting and Collaborating through Computing*, 104-109.

- Martin, R. (2009). *The Design of Business: Why Design Thinking is the Next Competitive Advantage*. Harvard Business Press.
- Medway, P. (1996). Virtual and material buildings: Construction and constructivism in architecture and writing. *Written Communication*, 13(4), 473 - 514.
- Murphy, P., & Hennessy, S. (2001). Realising the potential - and lost opportunities - for peer collaboration in a D&T setting. *International Journal of Technology and Design Education*, 11, 203 - 237.
- Noel, L., & Liub, T. (2017). Using design thinking to create a new education paradigm for elementary level children for higher student engagement and success. *Design and Technology Education: An International Journal*, 22(1), 501 - 512.
- O'Donnell, A., & Dansereau, D. (1992). *Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance*. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interactions in cooperative groups: The theoretical anatomy of group learning* (pp. 120 - 141). Cambridge University Press.
- Oliver, R., Omari, A., & Herrington, J. (1998). Exploring student interactions in collaborative World Wide Web computer based learning environments. *Journal of Educational Multimedia and Hypermedia*, 7(2-3): 263 - 287.
- Owen, C. (2007). Design thinking: Notes on its nature and use. *Design Research Quarterly*, 21(1), 16-27.
- Partnership for 21st Century Skills. (2009). *P21 framework definitions*. Retrieved September 13, 2021, from <https://files.eric.ed.gov/fulltext/ED519462.pdf>
- Puntambekar, S., & Kolodner, J. (2005). Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 42(2), 185-217.

- Renard, H. (2014). Cultivating Design Thinking in Students through Material Inquiry. *International Journal of Teaching and Learning in Higher Education*, 26(3), 414-424.
- Retna, K. (2016). Thinking about “design thinking”: A study of teacher experiences. *Asia Pacific Journal of Education*, 36(1), 5-19.
- Rossett, A. (1987). *Training needs assessment*. Englewood Cliffs. Educational Technology Publications.
- Roth, B. (2015). *Design thinking in Stanford*. In C. Meinel, U. Weinberg & T. Krohn (Eds.), *Design thinking live*. 64 - 71; 250 - 251. Murmann.
- Rowe, G. (1987). *Design Thinking*. The MIT Press.
- Runco, M. (1989). The creativity of children’s art. *Child Study Journal*, 19, 177-190.
- Runco, M. (2001). Introduction to the special issue: Commemorating Guilford’s 1950 presidential address. *Creativity Research Journal*, 13(3-4), 245.
- Saltmarsh, J., Hartley, M., & Clayton, P. (2009). *Democratic engagement white paper*. New England Resource Center for Higher Education, Retrieved September 13, 2021, from https://repository.upenn.edu/cgi/viewcontent.cgi?article=1252&context=gse_pubs
- Schatzki, T. (2001). *Practice theory*. In T.R. Schatzki, K. Knorr Cetina, and E. von Savigny (Eds.), *The practice turn in contemporary theory*. Routledge.
- Scheer, A., Noweski, C., & Meinel, C. (2012). Transforming constructivist learning into action: Design thinking in education. *Design and Technology Education: An International Journal*, 17(3), 8-19.
- Schön, D. (1983). *The reflective practitioner: How professionals think in action*. RoMedwayutledge.
- Schwab, K. (2016). *The Fourth Industrial Revolution: what it means, how to respond*. Retrieved September 13, 2021, from

<https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>

- Simon, H. (1969). *The Sciences of the Artificial*. MIT Press.
- Stahl, G., Koschmann, T., & Suthers, D. (2006). *Computer-supported collaborative learning: An historical perspective*. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences*. 409-426. Cambridge University Press.
- Standford d.school. (2010). *An introduction to design thinking process guide*. Stanford d.school.
- Stewart, S. (2011). Editorial. Interpreting design thinking. *Design Studies*, 32(6), 515 - 520.
- Suchman, L. (1987). *Plans and situated actions*. Cambridge University Press.
- Suwa, M., Gero, J., & Purcell, T. (2000). Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. *Design studies*, 21(6), 539-567.
- Teasley, S., & Roschelle, J. (1993). *Constructing a joint problem space: The computer as a tool for sharing knowledge*. In S. P. Lajoie & S. J. Derry (Eds.). Erlbaum.
- Torrance, E. (1966). *Torrance tests of creativity thinking: Norms, technical manual*. Princeton, Personnel Press.
- Torrance, E. (1974). *Norms-technical manual: Torrance tests of creative thinking*. Lexington, Ginn and Company.
- Treffinger, D., Isaksen, S., & Firestien, R. (1983). Theoretical perspectives on creative learning and its facilitation: An overview. *Journal of Creative Behavior*, 17(1), 9-17.
- Tyler, J. (2014). *App Inventor for Android*. Wiley.
- Watson, A. (2015). Design thinking for life. *Art Education*, 68(3), 12-18.
- Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2015). *App Inventor 2: Create Your Own Android Apps*. O'Reilly Media.

- World Economic Forum. (2015). *New vision for education: Unlocking the potential of technology*. Retrieved September 13, 2021, from http://www3.weforum.org/docs/WEFUSA_NewVisionforEducation_Report2015.pdf
- World Economic Forum. (2016). *The future of Jobs: Employment, skills and workforce strategy for the fourth industrial revolution*. Retrieved September 13, 2021, from http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf
- Zabelina, D., & Robinson, M. (2010). Creativity as flexible cognitive control. *Psychology of Aesthetics, Creativity, and the Arts*, 4(3), 136 - 143.

<ABSTRACT>

Design Thinking-based Software Education Program for Creativity Enhancement

Youngho Seo

Major in Computer Education, Faculty of Science Education
Graduate School, Jeju National University

Supervised by professor Jonghoon Kim

This study develops a design thinking-based software education program for enhancing creativity, and verifies the educational effect by applying it to the field.

In the age of the Fourth Industrial Revolution, the speed of transformation in human life, including science and technology, will become unpredictably faster, and the linkage and convergence between various fields of society will accelerate. Simultaneously, the rates of extinction, differentiation, and creation of jobs will lead to expedited yet uncertain situations. In this rapidly changing society, it has become critical for students to acquire capacities such as creative thinking skills, knowledge processing skills, communication, self-management, community collaboration, and aesthetic sensitivity. To help strengthen these competencies, project-like classes in which students can

work together closely are necessary.

This study developed a design thinking-based software education program according to the Analysis-Design-Development-Implementation-Evaluation (ADDIE) model to enhance creativity for elementary school students and prospective elementary school teachers, and the educational effect was verified through field application.

In the analysis stage, the subjects included elementary school students and teachers, and the survey was conducted to analyze the learners' needs following the analysis of previous studies on creativity, design thinking-based software education, and Computer Supported Collaborative Learning(CSCL), as well as Rossett's needs-analysis model.

Concerning the design stage, the education program was designed through the process of specifying achievement goals, designing program structure, designing learning content, designing teaching strategies, developing principles for educational programs, and selecting educational tools.

For the development stage, based on the content of the design stage, developed three design thinking-based software education programs.

In the application stage, three design thinking-based software education programs were developed according to the design thinking-based software education tools and were applied to prospective elementary school teachers and elementary school students a total of three times to verify their educational effect. In a situation where cooperation is required in the design thinking process, Padlet and Google Slides were used as CSCL tools. The three education programs that consisted of design thinking-based software education programs include Scratch, App Inventor, and micro:bit.

Upon evaluation, both types A and B of Torrance Tests of Creative Thinking (TTCT) were employed as creativity test tools. The study analyzed programs using Scratch, App Inventor, and micro:bit in the evaluation stage, and all were proven to be effective in enhancing the creativity of elementary

school students.

This study has the following implications.

First, the design thinking-based software education programs are effective in enhancing creativity. The students have undergone the five steps: empathy, problem definition, idea generation, drafting, and verification, and such educational programs and the problem-solving skills have proven to increase their creativity.

Second, it suggested that educational programming languages such as Scratch, App Inventor, and micro:bit can be used as practical tools for design thinking-based software education programs that explore experiential activities.

Third, the possibility of generalizing design thinking-based software education for prospective elementary school teachers and elementary school students was suggested, by developing educational materials and teacher's editions to which such programs can be applied.

In the future, if the design thinking-based software education program developed herein can continue to expand and target larger groups of students through follow-up studies, it will become another popular software education program for strengthening creativity.

부 록

<부록 1> 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 교재 117

<부록 2> 애플벤터를 활용한 디자인씽킹 기반 소프트웨어 교육 교재 162

<부록 3> 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 교재 213

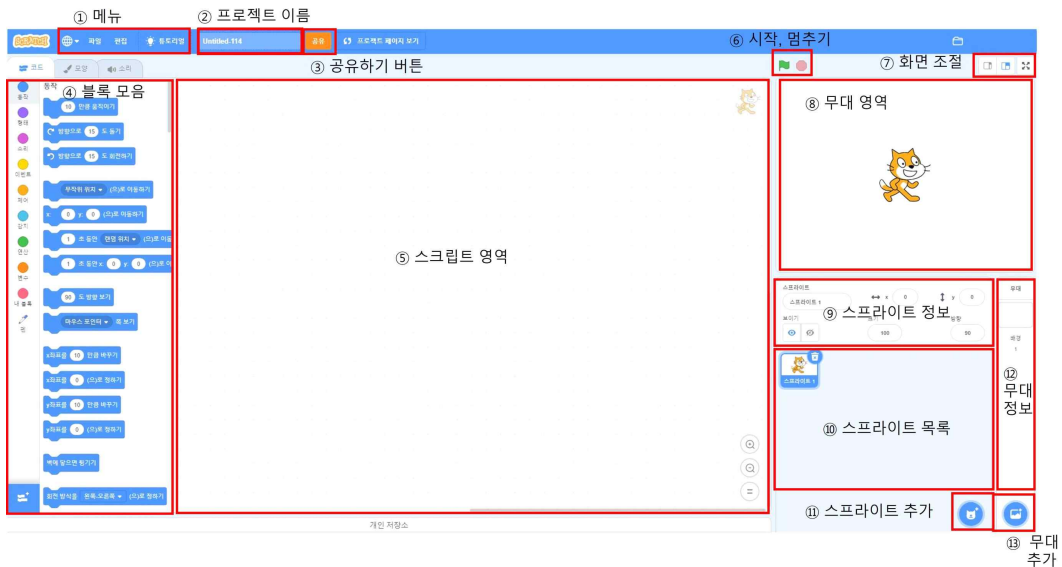
<부록 1> 스크래치를 활용한 디자인씽킹 기반 소프트웨어 교육 교재

스크래치를 활용한
디자인씽킹 기반 소프트웨어 교육

1. 스크래치 기본 블록 이해하기

학습목표: 스크래치 기본 블록 사용 방법을 이해할 수 있다.

1. 스크래치 화면 구성



① 메뉴

- 언어 선택: 한국어, 영어 등 사용 언어를 선택
- 파일: 새로 만들기, 저장하기, 불러오기 기능
- 편집: 스프라이트 되돌리기, 터보 모드 켜기
- 튜토리얼: 제공 프로젝트 학습

② 프로젝트 이름: 현재 작업 중인 프로젝트 이름을 저장

③ 공유: 다른 사람들과 프로젝트를 공유

④ 블록 모음

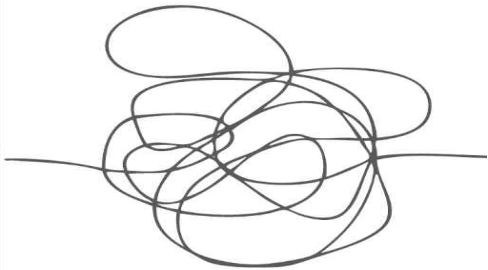
- 동작: 스프라이트의 움직임
- 형태: 스프라이트의 모양 및 말하기 표현
- 소리: 스프라이트의 소리 효과
- 이벤트: 프로젝트 시작 및 메시지 주고 받을 때 사용
- 제어: 반복, 조건 등을 사용
- 감지: 묻고 답하기, 상황 감지

- 변수: 변수, 리스트 관련
 - 내 블록: 새로운 블록 만들기
 - 확장 기능 추가하기: 음악, 펜, 비디오 감지 등 블록을 확장
- ⑤ 스크립트 영역: 블록을 이용해서 스크립트를 작성하는 공간
 - ⑥ 시작, 멈추기: 프로젝트의 시작, 멈추기 버튼
 - ⑦ 화면 조절: 무대와 스크립트 영역의 크기를 조절
 - ⑧ 무대 영역: 스프라이트가 동작하는 공간
 - ⑨ 스프라이트 정보: 선택 스프라이트의 정보 제공 및 수정
 - ⑩ 스프라이트 목록: 생성한 스프라이트 목록 제공 및 수정
 - ⑪ 스프라이트 추가
 - 스프라이트 고르기: 제공하는 스프라이트 추가
 - 그리기: 사용자가 직접 스프라이트 그려서 추가
 - 서프라이즈: 제공하는 스프라이트 중 임의로 골라서 추가
 - 스프라이트 업로드하기: 사용자 컴퓨터에서 그림 파일 추가
 - ⑫ 무대 정보: 무대의 정보 제공 및 수정
 - ⑬ 무대 추가
 - 무대 고르기: 제공하는 무대 추가
 - 그리기: 사용자가 직접 무대 그려서 추가
 - 서프라이즈: 제공하는 무대 중 임의로 골라서 추가
 - 무대 업로드하기: 사용자 컴퓨터에서 그림 파일 추가

2. 디자인씽킹 이해하기

학습목표: 디자인씽킹을 이용한 문제 해결 절차를 알 수 있다.

1. 디자인씽킹과 혁신

 <p>- 애플 창조서비스 팀 브레년의 디자인</p>	<p>혁신(Innovation)</p> <p>'혁신은 사회과학(인간적 가치)과 비즈니스, 기술의 교집합에서 나온다'</p> <p>↓</p> <p><u>디자인 씽킹: 미 디자인회사 아이데오의 CEO팀 브라운이 제안하고 적용해 널리 알려진 혁신을 위한 사고방식</u></p>
--	--

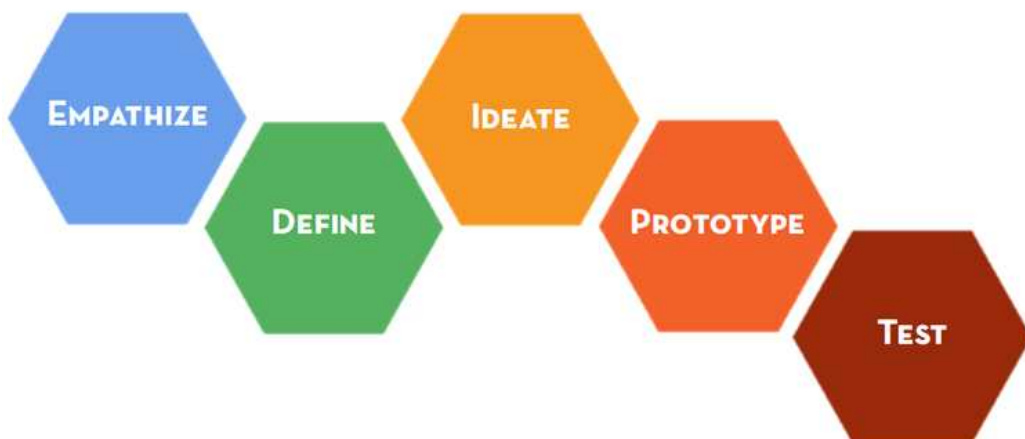
2. 디자인씽킹이란?

가. 디자이너가 생각하는 방식으로 문제를 해결하는 방법

→ 새로운 시각으로 세상을 바라보고 예전에는 몰랐던 방식으로 문제 해결

나. 사람을 중심으로 하는 창의적인 생각하는 태도

→ 자신과 타인을 더욱 잘 이해할 수 있고 삶의 융통성 증가



3. 디자인 씽킹의 절차

노인 문제 해결 프로젝트

<1단계: 공감 단계> 해결해야 할 문제들이 무엇인가 고민하는 단계

※ observe(관찰), interview(면담), immerse(경험) / 사용자의 요구 파악(선택 사항이 있는 질문지, 가설의 증명여부 x)

- 인터뷰: '생활하면서 불편한 점은?' '평소에 아드님 댁을 자주 가세요?'
- 노인에 대한 선입견: 나이가 많으니 일하기 싫어할 것이며, 무척 단조로운 생활을 할 것이다.



<2단계: 정의 단계> 명확하게 어떤 것이 문제인지 정의 내리는 단계

※ real(진짜 문제인가), valuable(해결할 만한 가치가 있는가), Inspiring (영감을 주는가?)

- 노인: 나이가 많고 체력이 약하지만 도움이 주는 존재가 되어 자신의 가치 증명 원함
- 젊은이: 노인들이 힘들거나 다칠까봐 호의 대부분을 거절
- 왜 갈등이 생겼는지 생각. 어떻게 하면 노인들이 자신의 가치를 증명하는 일을 도울 수 있을까?



<3단계: 아이디어 발상 단계> 문제에 대한 해결방법을 찾는 단계

- 5분 안에 백 개 이상의 해결 방법 생각하기/ 많은 아이디어 만들기
- 떠오른 아이디어를 즉시 노트에 적거나 그린 후, 크게 소리내어 읽기



<4단계: 시안 만들기 단계> 손을 움직여 아이디어에 등장했던 물건 만들기

- 좋은 아이디어는 시각화, 구체화가 필요하므로 시제품으로 만들/ 빠르고 저렴해야 함/ 소통, 협업 등이 이루어짐.
- 손을 움직여 아이디어 도출 단계에서 나왔던 물건 만들기(노인용 텔레비전, 기내 표시등, 시각 장애인용 스위치)
- 추상적인 아이디어가 좀 더 구체화 됨.
(노인용 기내 표시등- 노인들 시력 - 글자체 크게)



<5단계: 검증하기 단계> 생산된 시제품을 사용자들로부터 검증 받는 단계

- 만든 시제품을 사용자에게 사용하게 함으로서 문제를 재정비함.
- 필요시 다시 그 전 단계로 돌아감.

4. 디자인 씽킹과 개인의 삶

- 가. 현재 내가 생각한 문제: 비만(살을 빼기 위해 다 해 봤으나 살이 빠지지 않음. 인간관계 단절)
- 나. 나의 내면과 공감을 시도한 후 찾은 문제: 살을 빼고 싶은 것이 아니라 자신감을 찾고 싶음.
→ 헬스장 대신 인간관계에 집중하고 일찍 자고 체력을 보충하는 방향으로 생활을 개선하자 자연스럽게 식생활도 개선됨.

3. 도형 그리기

학습목표: 순차, 반복을 이용하여 여러 가지 정다각형을 그릴 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]


- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 정삼각형, 정사각형, 정오각형... 등 그리기 원리를 아는 대로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 정다각형을 그리는 프로젝트를 만들어 봅시다.

- 1)  (확장 기능 추가하기) 버튼을 클릭하여 펜 기능을 추가합니다.

2) 이동하기, 오른쪽으로 회전하기, 반복하기 블록을 이용해서 정삼각형을 그립니다.

<pre> when clicked clear all pen down repeat (3) move 100 units turn right 120 degrees pen up </pre>	
--	--

3) 이동하기, 오른쪽으로 회전하기, 반복하기 블록을 이용해서 정사각형을 그립니다.

<pre> when clicked clear all pen down repeat (4) move 100 units turn right 90 degrees pen up </pre>	
---	--

4) 이동하기, 오른쪽으로 회전하기, 반복하기 블록을 이용해서 정오각형을 그립니다.

<pre> when clicked clear all pen down repeat (5) move 100 units turn right 72 degrees pen up </pre>	
---	--

5) 프로그램 사용자에게 데이터를 직접 입력 받아서 그리는 프로그램을 작성합니다.



The image shows a Scratch script on the left and a drawing area on the right. The script starts with a 'when clicked' event, followed by 'clear all', 'pen down', a 'say "Please enter the number of polygons you want to draw." and wait' block, a 'repeat' block with 'times' set to 100, and 'pen up' at the end. The drawing area contains a yellow pencil icon and a speech bubble with the text '그리고 싶은 정다각형을 숫자로 작성하세요.' (Please enter the number of polygons you want to draw.). Below the drawing area is a text input field with a blue checkmark icon on the right.

5. 검증하기 단계 [Test]

1) 도형 그리기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 도형 그리기 프로그램의 장점을 적어주세요.

3) 도형 그리기 프로그램에 추가하면 좋을 내용을 적어주세요.

4. 창의적인 모양 그리기(1)

학습목표: 정다각형을 이용하여 창의적인 모양을 그릴 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 정다각형을 중복해서 그릴 때 어떤 원리가 있는지 아는 대로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 정다각형을 중복해서 그리는 프로젝트를 만들어 봅시다.

- 1)  (확장 기능 추가하기) 버튼을 클릭하여 펜 기능을 추가한다.

2) 정삼각형을 이용하여 6번 회전하면서 그려 봅시다.

<pre> when clicked clear all pen down repeat (3) move 50 turn 120 clockwise turn [] clockwise pen up </pre>	
--	--

3) 정사각형을 이용하여 10번 회전하면서 그려 봅시다.

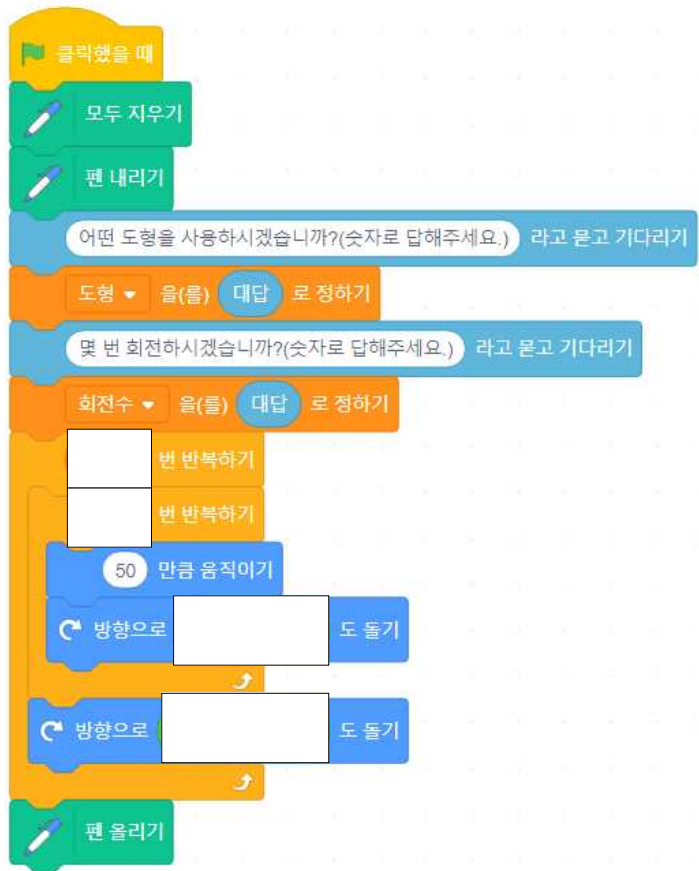
<pre> when clicked clear all pen down repeat (4) move 50 turn 90 clockwise turn [] clockwise pen up </pre>	
---	--

4) 도형과 회전수를 입력 받아 동작하는 프로그램을 만들어 봅시다.

- (1) 도형과 회전수 변수를 만듭니다.
- (2) 도형과 회전수 변수에 프로그램 사용자로부터 값을 입력받도록 합니다.
- (3) 도형 변수를 이용하여 반복하기와 회전하는 값을 수식으로 생각해 봅시다.

(4) 회전수 변수를 이용하여 도형을 그리는 횟수와 회전하는 값을 수식으로 생각해 봅시다.

(5) 프로그램을 완성해 봅시다.



5. 검증하기 단계 [Test]

1) 창의 도형 그리기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 창의 도형 그리기 프로그램의 장점을 적어주세요.

3) 창의 도형 그리기 프로그램에 추가하면 좋을 내용을 적어주세요.

5. 창의적인 모양 그리기(2)

학습목표: 원을 이용하여 태극 문양을 그릴 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]


- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]


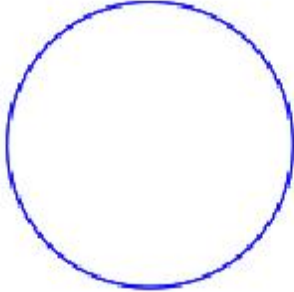
- 원을 이용하여 태극 문양을 그릴 때 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

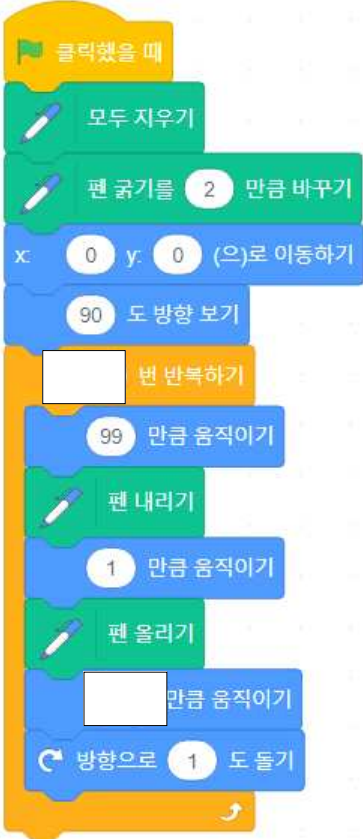
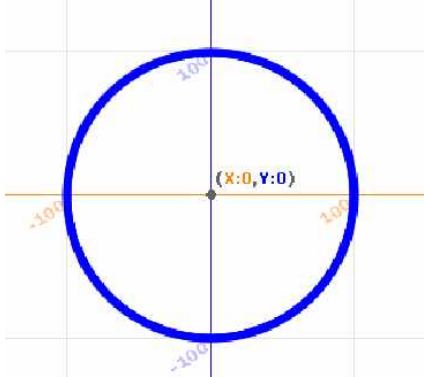
- 정다각형을 중복해서 그리는 프로젝트를 만들어 봅시다.

- 1)  (확장 기능 추가하기) 버튼을 클릭하여 펜 기능을 추가한다.

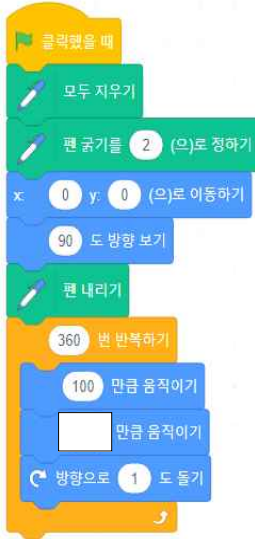
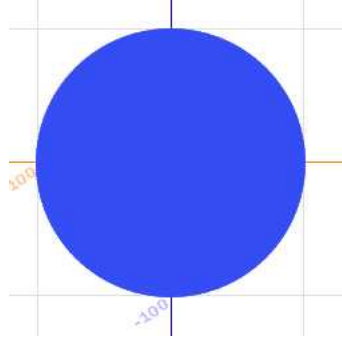
2) 원을 그려 봅시다.

 <pre> when clicked clear pen down repeat () times move 1 units turn 1 degrees clockwise </pre>	
---	--

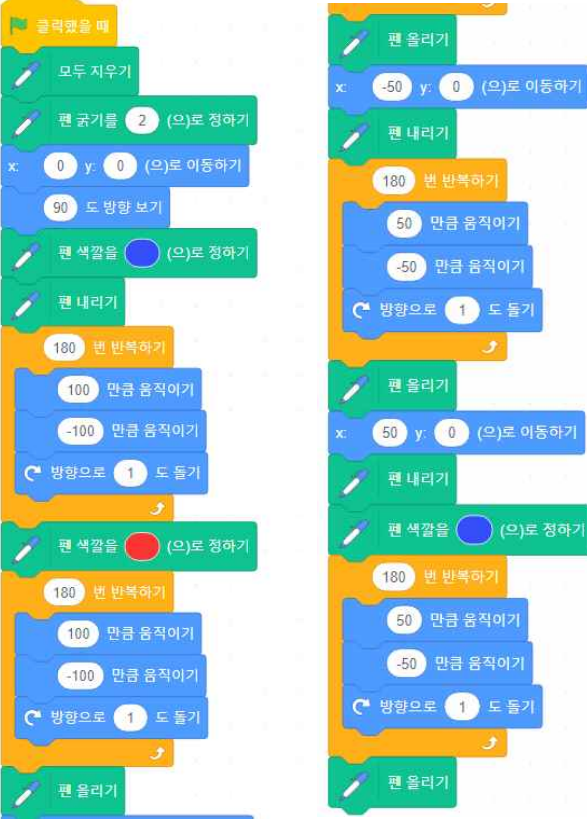
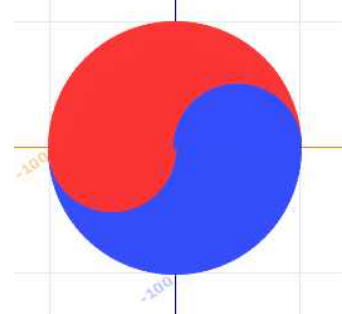
3) 반지름이 100 픽셀인 원을 그려 봅시다.

 <pre> when clicked clear pen radius 2 units move to x: 0 y: 0 turn 90 degrees repeat () times move 99 units pen down move 1 units pen up move units turn 1 degrees clockwise </pre>	
---	--

4) 색이 칠해진 원을 그려 봅시다.

	
---	--

5) 태극 문양을 그려 봅시다.

	
---	--

5. 검증하기 단계 [Test]

1) 태극 문양 그리기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 태극 문양 그리기 프로그램의 장점을 적어주세요.

3) 태극 문양 그리기 프로그램에 추가하면 좋을 내용을 적어주세요.

6. 계산 프로그램 만들기(1)

학습목표: 계산 프로그램의 원리를 알고 간단한 사칙연산 계산기를 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 사칙연산에 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 3개의 스프라이트를 사용합니다.



1) 덧셈만 가능한 계산기 프로젝트를 만들어 봅시다.



```

클릭했을 때
첫째수 > 음(음) 0 로 정하기
둘째수 > 음(음) 0 로 정하기
계산값 > 음(음) 0 로 정하기
첫 수를 입력해 주세요. 라고 묻고 기다리기
첫째수 > 음(음) 대답 로 정하기
둘째 수를 입력해 주세요. 라고 묻고 기다리기
둘째수 > 음(음) 대답 로 정하기
계산값 > 음(음) [ ] 로 정하기
    
```

2) 조건문을 활용하여 연산 종류에 따라 계산하는 프로젝트를 만들어 봅시다.



```

클릭했을 때
첫째수 > 음(음) 0 로 정하기
둘째수 > 음(음) 0 로 정하기
계산값 > 음(음) 0 로 정하기
첫 수를 입력해 주세요. 라고 묻고 기다리기
첫째수 > 음(음) 대답 로 정하기
둘째 수를 입력해 주세요. 라고 묻고 기다리기
둘째수 > 음(음) 대답 로 정하기
연산 기호를 입력해 주세요.(1. 덧셈, 2. 뺄셈, 3. 곱셈, 4. 나눗셈) 라고 묻고 기다리기
만약 < 대답 = 1 또는 대답 = 덧셈 > (이)라면
    덧셈 > 신호 보내기
    계산값 > 음(음) [ ] 로 정하기
만약 < 대답 = 2 또는 대답 = 뺄셈 > (이)라면
    뺄셈 > 신호 보내기
    계산값 > 음(음) [ ] 로 정하기
만약 < 대답 = 3 또는 대답 = 곱셈 > (이)라면
    곱셈 > 신호 보내기
    계산값 > 음(음) [ ] 로 정하기
만약 < 대답 = 4 또는 대답 = 나눗셈 > (이)라면
    나눗셈 > 신호 보내기
    계산값 > 음(음) [ ] 로 정하기
    
```

3) 연산 기호에 따라 화면의 연산기호가 바뀌도록 설정합니다.



5. 검증하기 단계 [Test]

1) 사칙연산 계산기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 사칙연산 계산기 프로그램의 장점을 적어주세요.

3) 사칙연산 계산기 프로그램에 추가하면 좋을 내용을 적어주세요.

7. 계산 프로그램 만들기(2)

학습목표: 계산 프로그램의 원리를 알고 최대공약수를 구하는 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

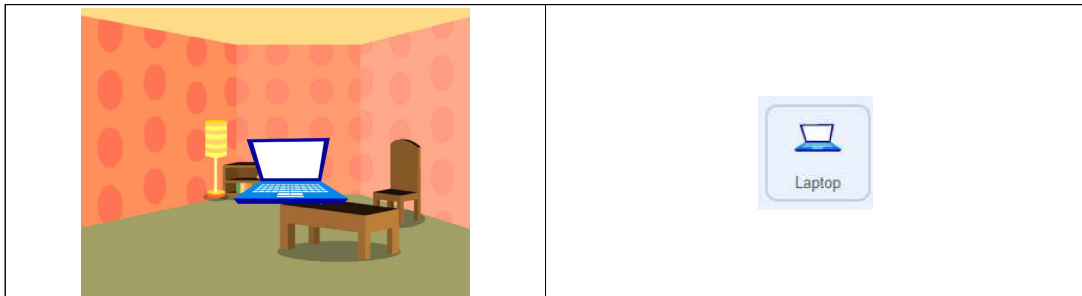
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 최대공약수에 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 1개의 스프라이트를 사용합니다.

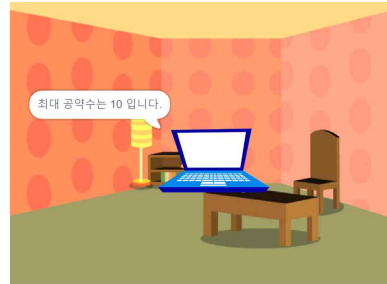


- 1) 작은 수와 큰 수를 입력 받아 1부터 작은 수까지 숫자를 늘려가며 작은 수와 큰

수를 모두 나눌 수 있는 수 중 가장 큰 수를 찾는 프로젝트를 만들어 봅시다.

```

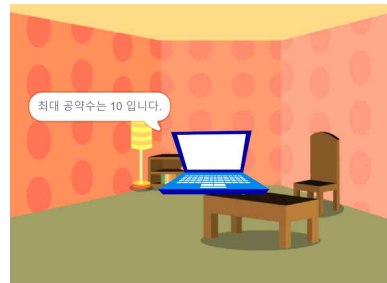
클릭했을 때
  A ← 음(음) 0 로 정하기
  B ← 음(음) 0 로 정하기
  작은 수를 입력하십시오. 라고 묻고 기다리기
  A ← 음(음) 대답 로 정하기
  큰 수를 입력하십시오. 라고 묻고 기다리기
  B ← 음(음) 대답 로 정하기
  n ← 음(음) 1 로 정하기
  n > A 까지 반복하기
  만약 [ ] 그리고 [ ] (아)라면
    최대공약수 ← 음(음) n 로 정하기
  n ← 음(음) 1 만큼 바꾸기
  최대공약수는 와(과) 최대공약수 와(과) 입니다. 결합하기 결합하기 말하기
  
```



2) 유클리드 호제법을 이용하여 최대공약수를 구하는 프로젝트를 만들어 봅시다.

```

클릭했을 때
  A ← 음(음) 0 로 정하기
  B ← 음(음) 0 로 정하기
  나머지 ← 음(음) 0 로 정하기
  첫번째 수를 입력하십시오. 라고 묻고 기다리기
  A ← 음(음) 대답 로 정하기
  두번째 수를 입력하십시오. 라고 묻고 기다리기
  B ← 음(음) 대답 로 정하기
  무한 반복하기
  나머지 ← 음(음) 로 정하기
  만약 나머지 = 0 (아)라면
    최대 공약수는 와(과) B 와(과) 입니다. 결합하기 결합하기 말하기
  A ← 음(음) B 로 정하기
  B ← 음(음) 나머지 로 정하기
  
```



5. 검증하기 단계 [Test]

1) 최대공약수 구하기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 최대공약수 구하기 프로그램의 장점을 적어주세요.

3) 최대공약수 구하기 프로그램에 추가하면 좋을 내용을 적어주세요.

8. 기초 게임 만들기

학습목표: 조건문을 이용하여 레이싱 게임을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

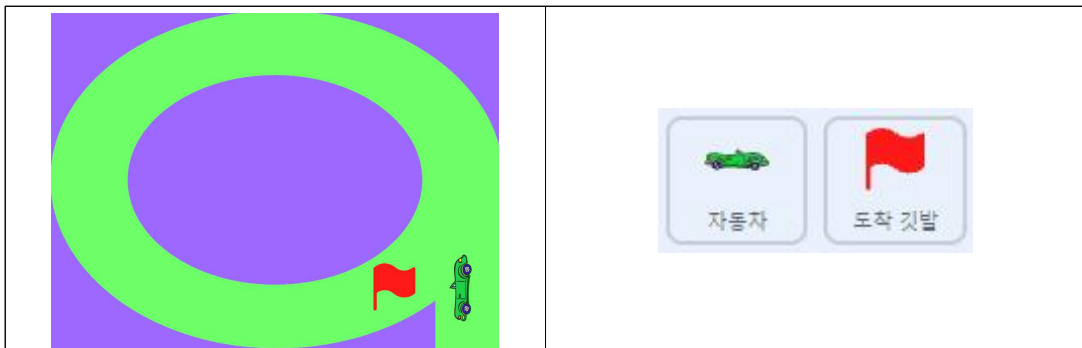
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 레이싱 게임과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 2개의 스프라이트, 무대 배경 그리기를 사용합니다.



1) 움직이는 자동차 스프라이트를 작성해 봅시다.



```

클릭했을 때
  타이머 초기화
  0 도 방향 보기
  x: 185 y: -140 (으)로 이동하기
  무한 반복하기
    5 만큼 움직이기
    만약 [ ] 키를 눌렀는가? (이)라면
      방향으로 3 도 회전하기
    만약 [ ] 키를 눌렀는가? (이)라면
      방향으로 3 도 돌기
  
```

2) 자동차가 도로 밖으로 나가면 원래 위치로 되돌아오도록 작성해 봅시다.



```

클릭했을 때
  무한 반복하기
    만약 [ ] 색에 닿았는가? (이)라면
      x: [ ] (으)로 이동하기
  
```


3) 도착 깃발에 닿으면 시간을 알려주고 게임을 종료합니다.



5. 검증하기 단계 [Test]

1) 레이싱 게임 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 레이싱 게임 프로그램의 장점을 적어주세요.

3) 레이싱 게임 프로그램에 추가하면 좋을 내용을 적어주세요.

9. 간단한 게임 완성하기(1)

학습목표: 변수를 이용하여 간단한 게임을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

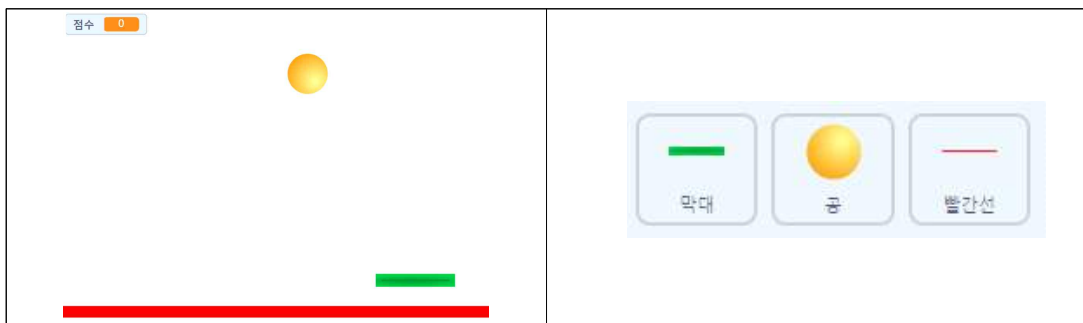
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 공 튀기기 게임과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 3개의 스프라이트를 사용합니다.



1) 막대 스프라이트로 좌우 화살표를 이용하여 양쪽으로 이동하게 합니다.



```

클릭했을 때
  무한 반복하기
    만약 왼쪽 화살표 키를 눌렀는가? (아)라면
      [ ] 도 방향 보기
      10 만큼 움직이기
    만약 오른쪽 화살표 키를 눌렀는가? (아)라면
      [ ] 도 방향 보기
      10 만큼 움직이기
  
```

2) 공 스프라이트를 이용하여 녹색 막대와 벽을 만나면 반대 방향으로 튕기고 빨간선을 만나면 게임이 끝나도록 합니다.



```

클릭했을 때
  x: -150 부터 150 사이의 난수 y: 0 (으)로 이동하기
  -60 부터 60 사이의 난수 도 방향 보기
  무한 반복하기
    10 만큼 움직이기
    벽에 닿으면 튕기기
    만약 막대 에 닿았는가? (아)라면
      [ ] 도 방향 보기
    만약 빨간선 에 닿았는가? (아)라면
      배경을 배경 2 (으)로 바꾸기
      멈추기 모두
  
```

3) 시간의 흐름에 따라 점수를 부여합니다.



```

클릭했을 때
  배경을 배경 1 (으)로 바꾸기
  점수 을(를) 0 로 정하기
  무한 반복하기
    1 초 기다리기
    [ ]
  
```

5. 검증하기 단계 [Test]

1) 공 튀기기 게임 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 공 튀기기 게임 프로그램의 장점을 적어주세요.

3) 공 튀기기 게임 프로그램에 추가하면 좋을 내용을 적어주세요.

10. 간단한 게임 완성하기(2)

학습목표: 변수를 이용하여 간단한 게임을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

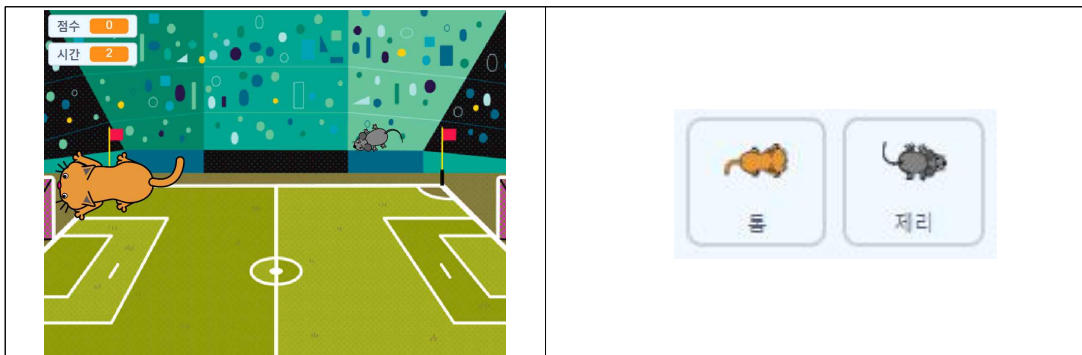
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 톰과 제리 게임과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 2개의 스프라이트를 사용합니다.



1) 톰 스프라이트는 상하, 좌우 화살표를 이용하여 이동하게 합니다.



```

클릭했을 때
x: -180 y: 0 (으)로 이동하기
무한 반복하기
  만약 < > 키를 눌렀는가? (아)라면
    0 도 방향 보기
    10 만큼 움직이기
  만약 < > 키를 눌렀는가? (아)라면
    180 도 방향 보기
    10 만큼 움직이기
  만약 < > 키를 눌렀는가? (아)라면
    90 도 방향 보기
    10 만큼 움직이기
  만약 < > 키를 눌렀는가? (아)라면
    -90 도 방향 보기
    10 만큼 움직이기
  
```

2) 게임 시간은 30초간 운영합니다. 무대에서 코딩하도록 합니다.



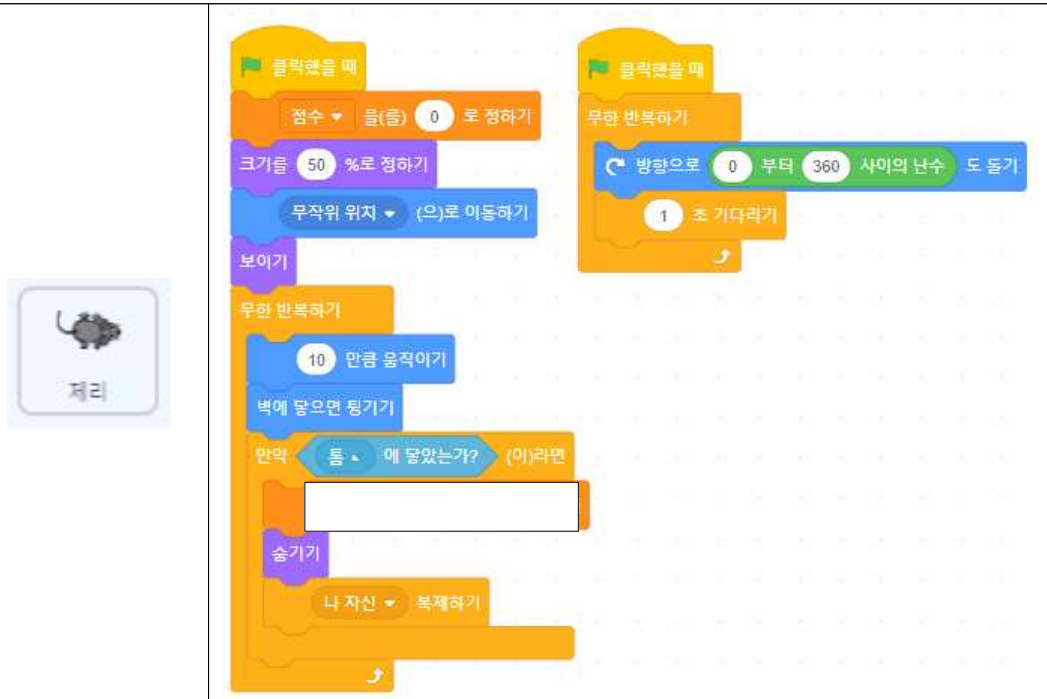
```

클릭했을 때
시간 > 들(음) 0 로 정하기
< > 까지 반복하기
  1 초 기다리기
  시간 > 들(음) 1 만큼 바꾸기
  
```

멈추기 모두 >

3) 제리 스프라이트를 작성해 봅시다.

- 방향을 전환하며 돌아다니고 톱과 달을 때마다 점수를 주고 사라집니다.



The image shows a Scratch code editor for a character named '제리' (Jerry). On the left is a blue square button with a black silhouette of Jerry and the text '제리'. To the right are two code blocks:

- 클릭했을 때 (When clicked):**
 - 점수 (Score) 을(을) 0 로 정하기 (Set score to 0)
 - 크기를 50 %로 정하기 (Set size to 50%)
 - 무작위 위치 (으)로 이동하기 (Move to random position)
 - 보이기 (Show)
 - 무한 반복하기 (Repeat forever):
 - 10 만큼 움직이기 (Move 10)
 - 벽에 닿으면 튕기기 (Bounce off wall)
 - 만약 톱 (Saw) 에 닿았는가? (이)라면 (If hit saw, then):
 - 점수 (Score) 을(을) 1 만큼 바꾸기 (Change score by 1)
 - 숨기기 (Hide)
 - 나 자신 (Myself) 복제하기 (Duplicate)
- 클릭했을 때 (When clicked):**
 - 무한 반복하기 (Repeat forever):
 - 방향으로 0 부터 360 사이의 난수 도 돌기 (Turn random degrees from 0 to 360)
 - 1 초 기다리기 (Wait 1 seconds)

- 사라졌다가 다시 나타나도록 합니다.



The image shows a Scratch code editor for the same character '제리' (Jerry). On the left is the same blue square button with the Jerry silhouette and text '제리'. To the right are two code blocks:

- 클릭했을 때 (When clicked):**
 - 무작위 위치 (으)로 이동하기 (Move to random position)
 - 보이기 (Show)
 - 무한 반복하기 (Repeat forever):
 - 10 만큼 움직이기 (Move 10)
 - 벽에 닿으면 튕기기 (Bounce off wall)
 - 만약 톱 (Saw) 에 닿았는가? (이)라면 (If hit saw, then):
 - 점수 (Score) 을(을) 1 만큼 바꾸기 (Change score by 1)
 - 숨기기 (Hide)
 - 나 자신 (Myself) 복제하기 (Duplicate)
- 클릭했을 때 (When clicked):**
 - 무한 반복하기 (Repeat forever):
 - 방향으로 0 부터 360 사이의 난수 도 돌기 (Turn random degrees from 0 to 360)
 - 1 초 기다리기 (Wait 1 seconds)

5. 검증하기 단계 [Test]

1) 톰과 제리 게임 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 톰과 제리 게임 프로그램의 장점을 적어주세요.

3) 톰과 제리 게임 프로그램에 추가하면 좋을 내용을 적어주세요.

11. 계산 게임 만들기

학습목표: 변수를 이용하여 간단한 암산 게임을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

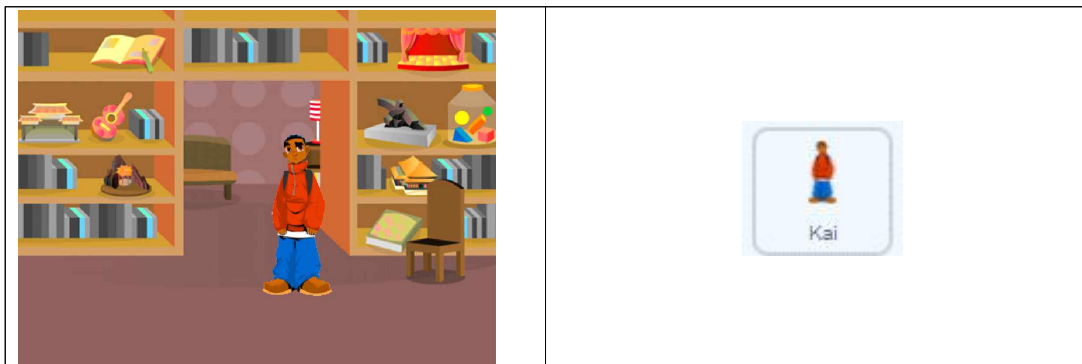
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 암산 게임과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 1개의 스프라이트를 사용합니다.



1) 연산 종류, 단계, 문항수를 사용자가 정할 수 있도록 합시다.

```

클릭했을 때
  무한 반복하기
    연산 종류를 선택하세요.(1 덧셈 2 뺄셈) 라고 묻고 기다리기
    연산종류 > 음(음) > 대답 > 로 정하기
    단계를 선택하세요.(1. 1단계 2. 2단계) 라고 묻고 기다리기
    단계 > 음(음) > 대답 > 로 정하기
    문항수를 입력하세요 라고 묻고 기다리기
    문항수 > 음(음) > 대답 > 로 정하기
    만약 < > (이)라면
      덧셈 > 신호 보내고 기다리기
    아니면
      뺄셈 > 신호 보내고 기다리기
    문항수 < 와(과) > 문제 중 < 와(과) > 맞은 문항수 < 와(과) > 문제 맞았습니다. 결합하기 > 결합하기 > 결합하기 > 음(음) > 2 초 동안 말하기
  멈추기 > 모두 >
  
```

2) 덧셈 게임을 할 수 있도록 만들어 봅시다.

```

덧셈 > 신호를 받았을 때
  맞은 문항수 > 음(음) > 0 > 로 정하기
  문항수 > 번 반복하기
    만약 < 단계 = 1 > (이)라면
      a > 음(음) > 1 > 부터 > 9 > 사이의 난수 > 로 정하기
      b > 음(음) > 1 > 부터 > 9 > 사이의 난수 > 로 정하기
    아니면
      a > 음(음) > 10 > 부터 > 99 > 사이의 난수 > 로 정하기
      b > 음(음) > 10 > 부터 > 99 > 사이의 난수 > 로 정하기
    a < 와(과) > + < 와(과) > b < 와(과) > = > 결합하기 > 결합하기 > 결합하기 > 라고 묻고 기다리기
    < > 로 정하기
    만약 < 정답 = 대답 > (이)라면
      정답! 음(음) > 2 초 동안 말하기
      맞은 문항수 > 음(음) > 1 > 만큼 바꾸기
    아니면
      더 공부하세요! 음(음) > 2 초 동안 말하기
  
```

3) 뽕셈 게임을 할 수 있도록 만들어 봅시다.

```

when green flag clicked
  set correct count to 0
  repeat (문항수) times
    if (단계 = 1) then
      set a to random number from 1 to 9
      set b to random number from 1 to 9
    else
      set a to random number from 10 to 99
      set b to random number from 10 to 99

    if (a < b) then
      set temp to a
      set a to b
      set b to temp

    ask (a 와(과) - 와(과) b 와(과) =) and wait for answer
    set correct to a - b

    if (정답 = 대답) then
      say (정답! 을(름) 2 초 동안 말하기) for 2 sec
      add 1 to correct count
    else
      say (더 공부하세요! 을(름) 2 초 동안 말하기) for 2 sec
  
```

5. 검증하기 단계 [Test]

1) 암산 게임 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 암산 게임 프로그램의 장점을 적어주세요.

3) 암산 게임 프로그램에 추가하면 좋을 내용을 적어주세요.

12. 동물 영어 퀴즈 프로그램 만들기

학습목표: 리스트를 이용하여 동물 영어 퀴즈 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

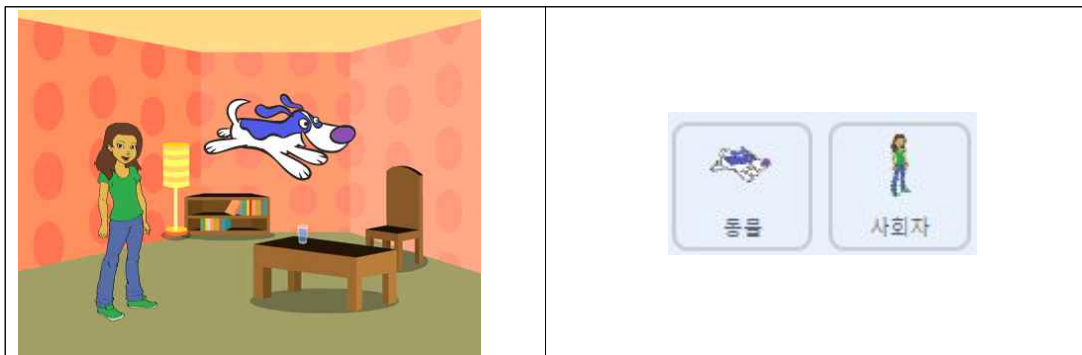
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 동물 영어 퀴즈 프로그램과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 2개의 스프라이트를 사용합니다.



1) 동물 모양에 7가지 모양으로 구성합니다.

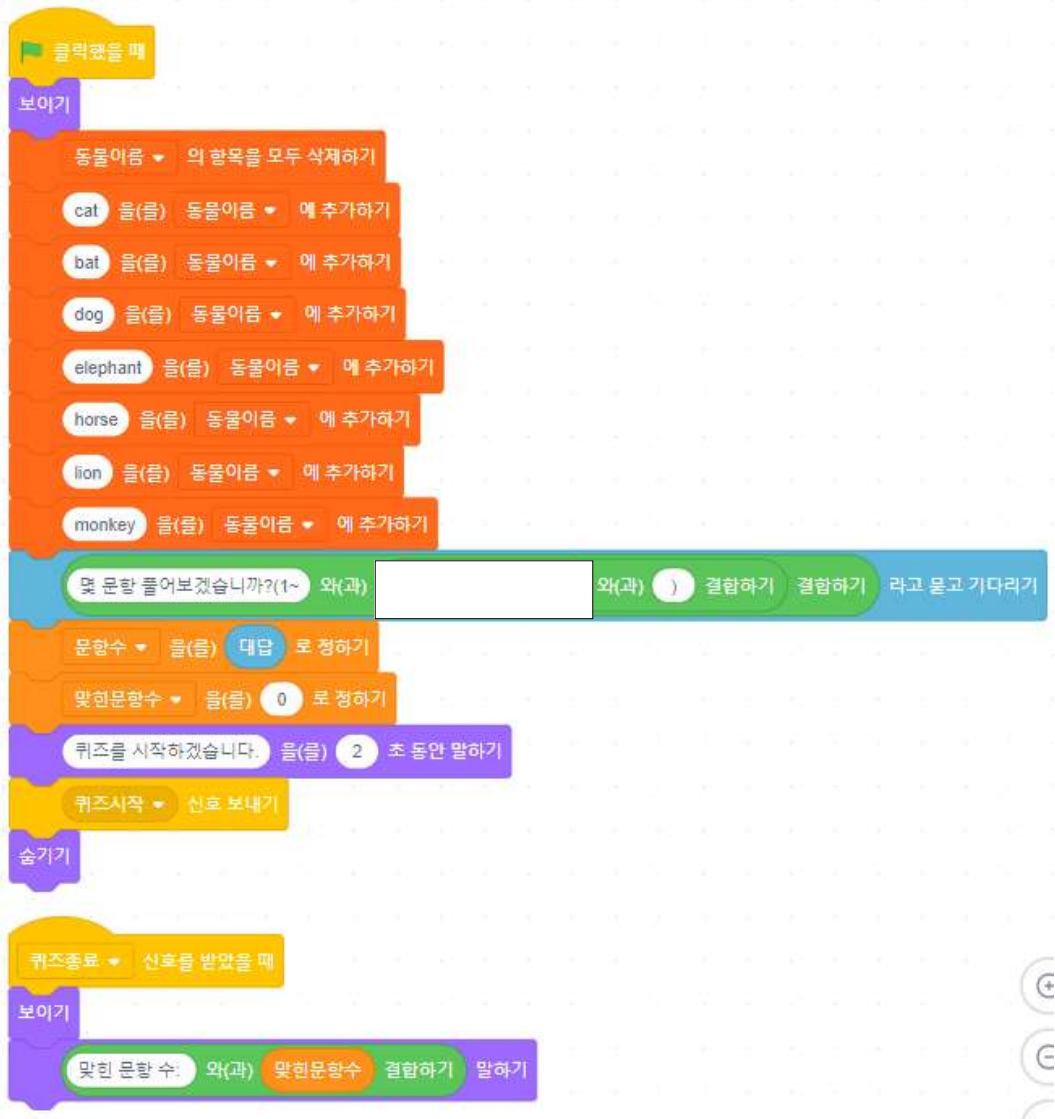


2) 동물 스프라이트는 모양을 바꾸어 가면 나타납니다.

```

when clicked
  hide
  quiz start
  show
  문항수 번 반복하기
  위치 음(음) 1 부터 동물이름 의 길이 사이의 난수 로 정하기
  모양을 [ ] (으)로 바꾸기
  영어 이름은? 라고 묻고 기다리기
  만약 대답 = 동물이름 리스트의 위치 번째 항목 (이)라면
    Suspense 재생하기
    맞았습니다! 음(음) 2 초 동안 말하기
    맞힌문항수 음(음) 1 만큼 바꾸기
  아니면
    String Pluck 재생하기
    틀렸습니다! 음(음) 2 초 동안 말하기
    위치 번째 항목을 동물이름 에서 삭제하기
  quiz 종료
  신호 보내기
  숨기기
  
```

3) 사회자 스프라이트는 문제를 제시하고 맞힌 문항수를 말해줍니다.



5. 검증하기 단계 [Test]

1) 동물 영어 퀴즈 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 동물 영어 퀴즈 프로그램의 장점을 적어주세요.

3) 동물 영어 퀴즈 프로그램에 추가하면 좋을 내용을 적어주세요.

13. 학생 추천 프로그램 만들기

학습목표: 리스트를 이용하여 학생 추천 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 학생 추천 프로그램과 관련된 원리를 아는 데로 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 화면의 구성은 아래와 같습니다. 6개의 스프라이트를 사용합니다.



1) 추천자 스프라이트는 각 버튼을 누르면 학생추가, 추천, 명단보기, 명단 삭제를 수행합니다.



```

클릭했을 때
  학생명부 > 의 항목을 모두 삭제하기






학생추가 > 신호를 받았을 때
  추가할 학생 이름을 적어주세요? 라고 묻고 기다리기
  만약 학생명부 > 이(가) 대답 을(를) 포함하는가? (이)라면
    다른 학생 이름을 입력하세요 을(를) 2 초 동안 말하기
  아니면
    [ ]

학생추천 > 신호를 받았을 때
  만약 학생명부 > 의 길이 > 0 (이)라면
    위치 > 을(를) 1 부터 학생명부 > 의 길이 사이의 난수 로 정하기
    학생명부 > 리스트의 위치 번째 항목 을(를) 2 초 동안 말하기
    [ ] 삭제하기
  아니면
    항목이 없습니다 을(를) 2 초 동안 말하기

명단보기 > 신호를 받았을 때
  만약 학생명부 > 의 길이 > 0 (이)라면
    학생명부 을(를) 2 초 동안 말하기
  아니면
    항목이 없습니다 을(를) 2 초 동안 말하기
  
```

 추가자	<pre> 명단삭제 ▾ 신호를 받았을 때 삭제할 항목? 라고 묻고 기다리기 만약 학생명부 ▾ 이(가) 대답 을(를) 포함하는가? (이)라면 학생명부 ▾ 리스트에서 대답 항목의 위치 번째 항목을 학생명부 ▾ 에서 삭제하기 아니면 삭제할 항목이 없습니다. 을(를) 2 초 동안 말하기 </pre>
--	---

2) 버튼들은 클릭했을 때 모양만 바꾸도록 합니다.

 추가버튼	<pre> 클릭했을 때 모양을 button2-a ▾ (으)로 바꾸기 이 스프라이트를 클릭했을 때 모양을 button2-b ▾ (으)로 바꾸기 0.2 초 기다리기 학생추가 ▾ 신호 보내기 모양을 button2-a ▾ (으)로 바꾸기 </pre>
 학생추가	
 보기버튼	
 삭제버튼	
 초기화버튼	

5. 검증하기 단계 [Test]

1) 학생 추천 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 학생 추천 프로그램의 장점을 적어주세요.

3) 학생 추천 프로그램에 추가하면 좋을 내용을 적어주세요.

<부록 2> 애플리케이션을 활용한 디자인씽킹 기반 소프트웨어 교육 교재

애플리케이션을 활용한
디자인씽킹 기반 소프트웨어 교육

1. 앱인벤터 기본 기능 익히기

학습목표: 앱인벤터 기본 기능을 익힐 수 있다.

1. 앱인벤터 기본 기능 익히기

1) 앱인벤터 활용 전 준비사항

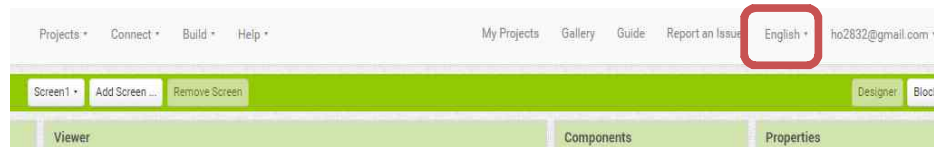
- ① 구글(google.com) 계정 필요
- ② 구글 크롬 브라우저(Google chrome) 설치 권장
 - 사파리(Safari), 파이어폭스(FireFox) 등도 가능
 - 익스플로러(Explorer) 사용 불가능

2) 앱인벤터 실행하기

- ① 앱인벤터 사이트 접속 (<http://appinventor.mit.edu/explore/>)



- ② 구글 아이디로 접속
- ③ 언어 변경하기 - English → 한국어



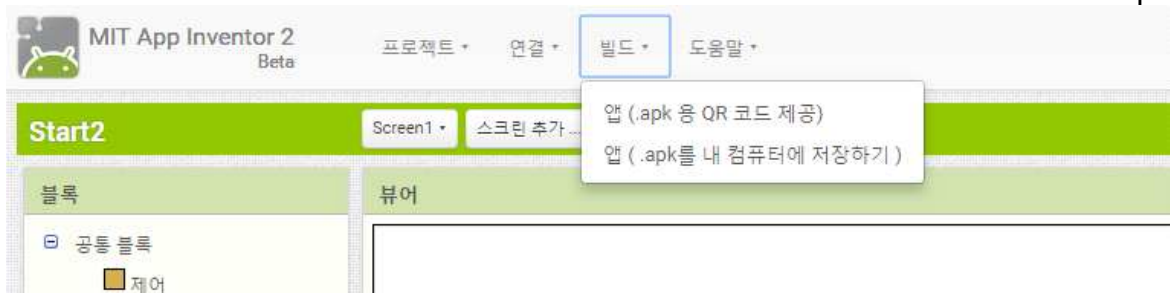
- ④ 프로젝트 - 새 프로젝트 시작하기

3) 라이브 테스트



- ① AI 컴패니언: 컴퓨터와 스마트폰이 같은 Wifi로 연결 되었을 때
 - Google Play Store에서 <MIT AI2 Companion> 앱을 폰에 설치
 - 컴퓨터와 폰에서 같은 Wifi로 접속
 - 앱 인벤터 화면 상단의 [화면] 메뉴를 클릭하고, [AI컴패니언] 선택
 - 폰에서 <MIT AI2 Companion> 앱을 실행하고, [scan QR code] 선택한 후 컴퓨터 화면에 나타난 QR 코드를 스캔
- ② 에뮬레이터: 컴퓨터 화면에서 테스트
 - <http://appinventor.mit.edu/explore/ai2/windows.html> 사이트에서 파일을 다운로드하고 설치
 - 바탕화면에 생성된 [aistarter] 파일 실행
 - 앱 인벤터 화면 상단의 [화면] 메뉴를 클릭하고, [에뮬레이터] 선택
- ③ USB: USB로 연결하여 스마트폰에서 실행
 - <http://appinventor.mit.edu/explore/ai2/windows.html> 사이트에서 파일을 다운로드하고 설치
 - Google Play Store에서 <MIT AI2 Companion> 앱을 폰에 설치
 - 바탕화면에 생성된 [aistarter] 파일 실행
 - 개발자 옵션 켜기 → 설정 - 휴대폰 정보 - 빌드번호(7번 누르기)
 - 컴퓨터와 스마트폰 연결하기

4) 앱인벤터 앱 설치하기

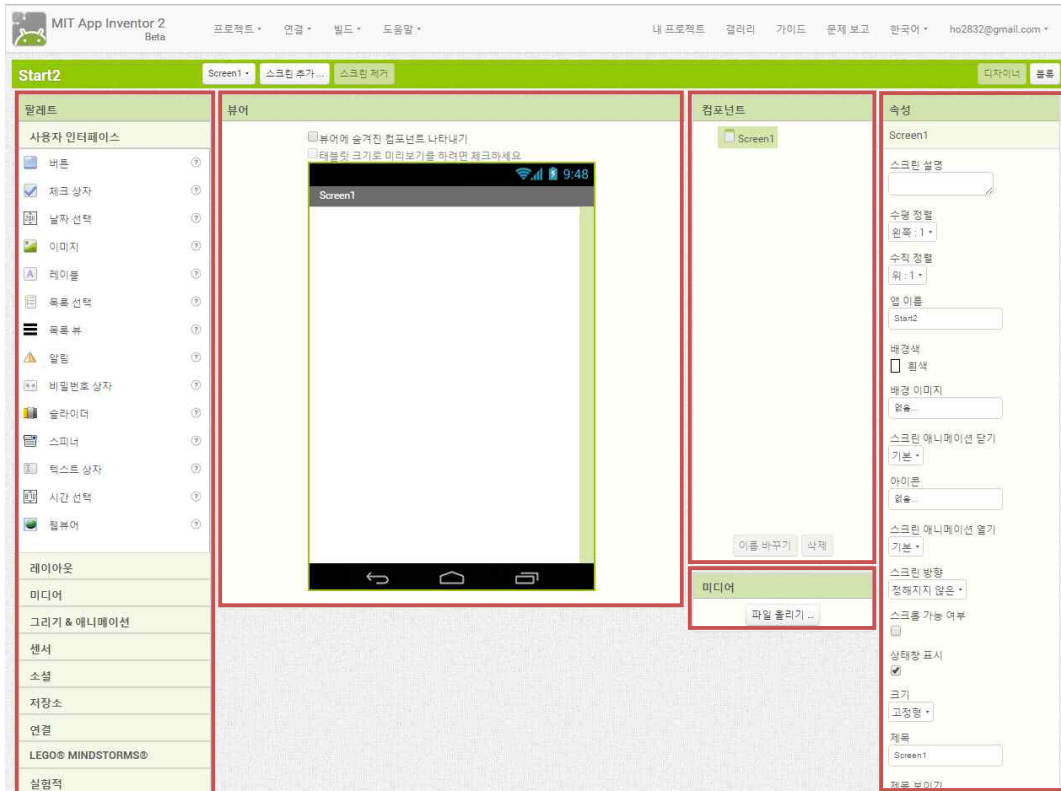


- ① 앱(.apk용 QR 코드 제공)
 - 클릭 후 생성된 QR코드를 QR코드 리더기로 읽으면 다운로드됨.
 - 설치하고 실행

② 앱(.apk용 내 컴퓨터에 저장하기)

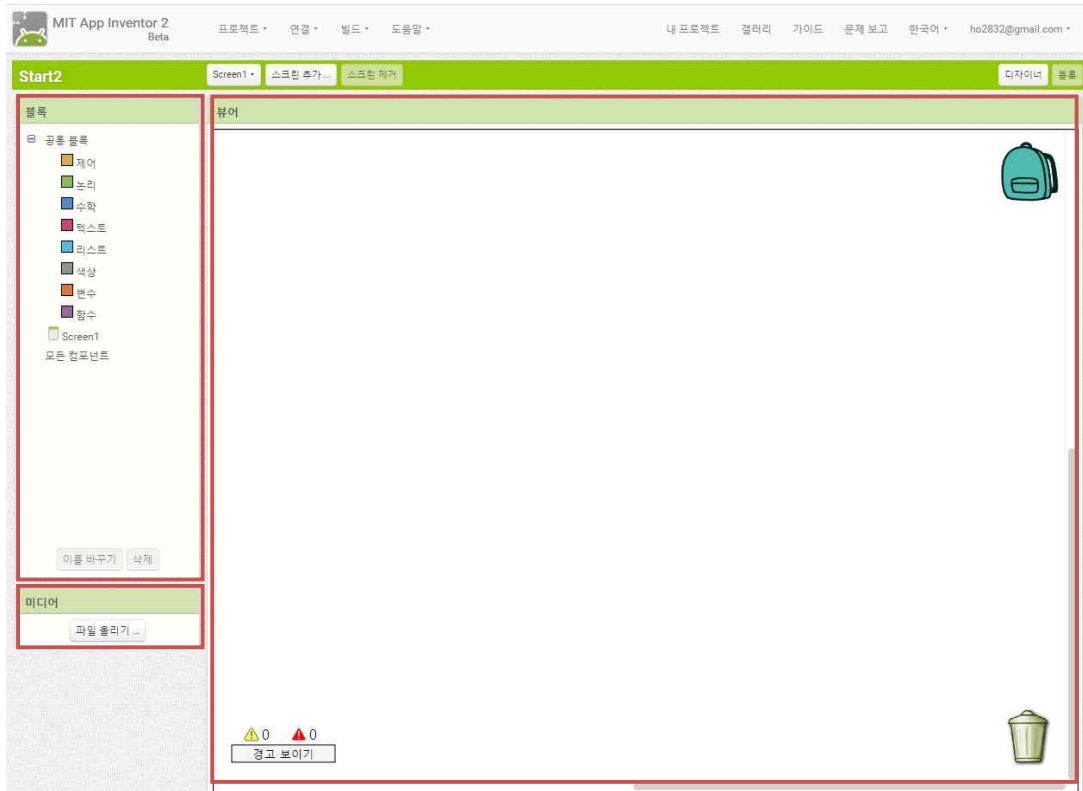
- USB로 연결하기 등을 이용하여 스마트폰으로 저장하여 실행

5) 앱 화면 설계 - 컴포넌트 디자이너(디자이너)



- ① 팔레트 - 뷰어에 배치할 수 있는 컴포넌트 목록을 보여줌.
- ② 뷰어 - 컴포넌트를 배치하여 앱의 화면 설계. 앱 화면을 간단하게 보여줌.
- ③ 컴포넌트 - 프로젝트에서 사용하고 있는 컴포넌트를 보여줌.
- ④ 미디어 - 프로젝트에 포함된 그림이나 소리와 같은 미디어 파일을 보여줌.
- ⑤ 속성 - 컴포넌트의 속성을 보여줌.

6) 앱 동작 지정 - 블록 에디터(블록)



- ① 블록 - 뷰어에 배치할 수 있는 블록 목록을 보여줌.
- ② 뷰어 - 뷰어에 배치된 블록들에 따라 앱이 실행됨.
- ③ 미디어 - 프로젝트에 포함된 그림이나 소리와 같은 미디어 파일을 보여줌.

2. 디자인씽킹 이해하기

학습목표: 디자인씽킹을 이용한 문제 해결 절차를 알 수 있다.

1. 디자인씽킹이란?

- 1) 디자이너가 생각하는 방식으로 문제를 해결하는 방법
→ 새로운 시각으로 세상을 바라보고 예전에는 몰랐던 방식으로 문제 해결
- 2) 사람을 중심으로 하는 창의적인 생각하는 태도
→ 자신과 타인을 더욱 잘 이해할 수 있고 삶의 융통성 증가

3. 디자인씽킹의 절차

노인 문제 해결 프로젝트

<1단계: 공감 단계> 해결해야 할 문제들이 무엇인가 고민하는 단계

※ observe(관찰), interview(면담), immerse(경험) / 사용자의 요구 파악(선택 사항이 있는 질문지, 가설의 증명여부 X)

- 인터뷰: '생활하면서 불편한 점은?' '평소에 아드님 댁을 자주 가세요?'
- 노인에 대한 선입견: 나이가 많으니 일하기 싫어할 것이며, 무척 단조로운 생활을 할 것이다.



<2단계: 정의 단계> 명확하게 어떤 것이 문제인지 정의 내리는 단계

※ real(진짜 문제인가), valuable(해결할 만한 가치가 있는가), Inspiring (영감을 주는가?)

- 노인: 나이가 많고 체력이 약하지만 도움이 주는 존재가 되어 자신의 가치 증명 원함
- 젊은이: 노인들이 힘들거나 다칠까봐 호의 대부분을 거절
→ 왜 갈등이 생겼는지 생각. 어떻게 하면 노인들이 자신의 가치를 증명하는 일을 도울 수 있을까?



<3단계: 아이디어 발상 단계> 문제에 대한 해결방법을 찾는 단계

- 5분 안에 백 개 이상의 해결 방법 생각하기/ 많은 아이디어 만들기
- 떠오른 아이디어를 즉시 노트에 적거나 그린 후, 크게 소리내어 읽기



<4단계: 시안 만들기 단계> 손을 움직여 아이디어에 등장했던 물건 만들기

- 좋은 아이디어는 시각화, 구체화가 필요하므로 시제품으로 만들/ 빠르고 저렴해야 함/ 소통, 협업 등이 이루어짐.
- 손을 움직여 아이디어 도출 단계에서 나왔던 물건 만들기(노인용 텔레비전, 기내 표시등, 시각 장애인용 스위치)
- 추상적인 아이디어가 좀 더 구체화 됨.
(노인용 기내 표시등- 노인들 시력 - 글자체 크게)



<5단계: 검증하기 단계> 생산된 시제품을 사용자들로부터 검증 받는 단계

- 만든 시제품을 사용자에게 사용하게 함으로서 문제를 재정비함.
- 필요시 다시 그 전 단계로 돌아감.

3. 청각 장애 도우미 앱

학습목표: 미디어 기능을 이용하여 청각 장애 도우미 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 청각 장애인과 대화를 하기 위해 어떤 기능이 필요한지 적어보세요.

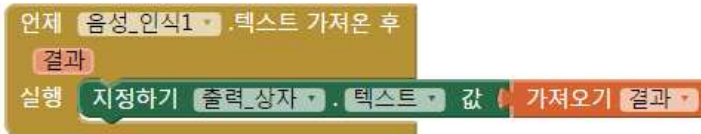
4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

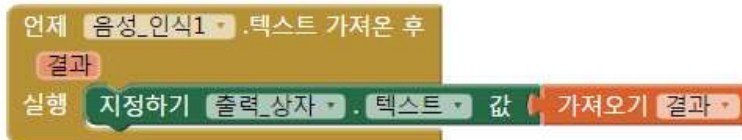
(1) 듣기 버튼을 클릭하면 음성 인식을 작동시켜 텍스트를 가지고 온 후 출력 상자 텍스트에 값을 가지고 온다.



(2) 말하기 버튼을 클릭하면 입력 상자의 텍스트를 음성 변환을 통하여 말하게 된다.



(3) 입력상자와 출력상자가 계속 비어 있을 수 있도록 변경합니다.



5. 검증하기 단계 [Test]

1) 청각 장애 도우미 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 청각 장애 도우미 앱의 장점을 적어주세요.

3) 청각 장애 도우미 앱에 추가하면 좋을 내용을 적어주세요.

4. 받아쓰기 앱

학습목표: 미디어 기능을 이용하여 받아쓰기 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 받아쓰기 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 사용되는 리스트와 변수를 초기화합니다.

```

전역변수 초기화 문제리스트 값 빈 리스트 만들기
전역변수 초기화 문제제출 값 "안녕"
전역변수 초기화 순서 값 0
  
```

(2) 문제입력버튼의 기능을 입력합니다.

```

언제 문제입력버튼 .클릭
실행 0 만약 문제확인상자 .텍스트 값 " "
그러면 지정하기 문제확인상자 .텍스트 값 " "
호출 음성_인식1 .텍스트 가져오기
아니면 호출 음성_인식1 .텍스트 가져오기

언제 음성_인식1 .텍스트 가져온 후
결과
실행 지정하기 문제확인상자 .텍스트 값 가져오기 결과
  
```

(3) 등록버튼의 기능을 입력합니다.

```

언제 등록버튼 .클릭
실행 0 만약 문제확인상자 .텍스트 값 " "
그리고 0 아니라 리스트에 포함되어 있나요? 값 문제확인상자 .텍스트
리스트 가져오기 global 문제리스트
그러면 0 리스트에 항목 추가하기 리스트 가져오기 global 문제리스트
Item 문제확인상자 .텍스트
지정하기 문제확인상자 .텍스트 값 " "
  
```

(4) 문제듣기버튼의 기능을 입력합니다.

```

언제 문제듣기버튼 .클릭
실행 0 만약 0 아니라 리스트가 비어있나요? 리스트 가져오기 global 문제리스트
그러면 지정하기 global 순서 값 임의의 정수 시작 1 끝 리스트 길이 리스트 가져오기 global 문제리스트
지정하기 global 문제제출 값 리스트에서 항목 선택하기 리스트 가져오기 global 문제리스트
위치 가져오기 global 순서
호출 음성_변환1 .말하기
메시지 가져오기 global 문제제출
아니면 호출 음성_변환1 .말하기
메시지 "문제를 등록하세요."
  
```

(5) 초기화버튼의 기능을 입력합니다.

```

언제 초기화버튼 .클릭
실행 각각 반복 항목 리스트 가져오기 global 문제리스트
실행 리스트에서 항목 삭제하기 리스트 가져오기 global 문제리스트
위치 1
  
```


(6) 확인버튼의 기능을 입력합니다.



5. 검증하기 단계 [Test]

1) 받아쓰기 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 받아쓰기 앱의 장점을 적어주세요.

3) 받아쓰기 앱에 추가하면 좋을 내용을 적어주세요.

5. 그림판 앱

학습목표: 그리기와 애니메이션의 기능을 이용하여 그림판 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 그림판 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작 설정해 봅시다.

가) 색상 버튼에 따라 캔버스 페인트 색상을 설정합니다.



나) 지우기 버튼을 클릭하면 캔버스를 지우고 페인트 색과 선 두께를 설정합니다.



다) 캔버스에 터치하면 점을 찍는 블록입니다.



라) 캔버스를 드래그하면 선을 그리는 블록입니다.



5. 검증하기 단계 [Test]

1) 그림판 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 그림판 앱의 장점을 적어주세요.

3) 그림판 앱에 추가하면 좋을 내용을 적어주세요.

6. 두더지 잡기 앱

학습목표: 그리기와 애니메이션의 기능을 이용하여 두더지 잡기 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

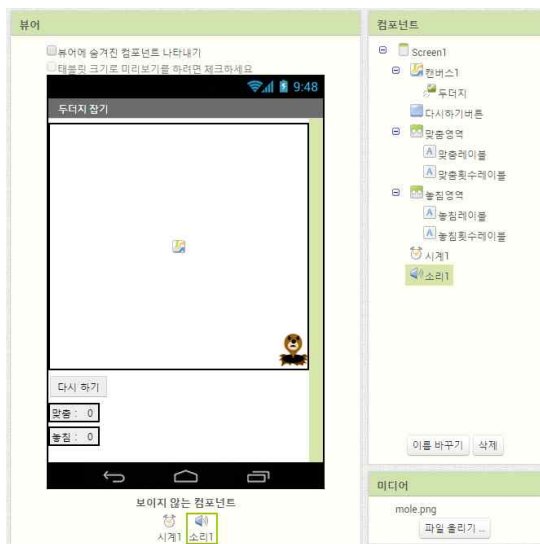
- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 두더지 잡기 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 화면이 초기화 될 때 두더지가 임의의 장소에서 출현하도록 하는 블록입니다.



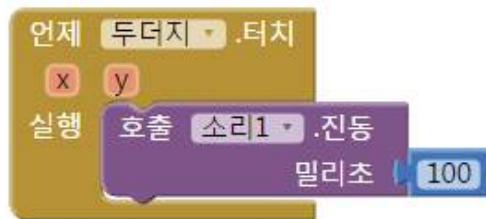
(2) 1밀리초(1/1000초) 간격으로 두더지가 임의의 장소에서 출현합니다.



(3) 스프라이트(두더지)를 터치하면 맞춤횟수를 1씩 증가시키고, 터치하지 못하면 놓침 횟수를 1씩 증가시켜 레이블에 표현합니다.



(4) 두더지가 터치될 때마다 0.1초간 진동이 울립니다.



(5) 다시하기버튼을 클릭하면 맞춤횟수레이블과 놓침횟수레이블을 초기화 합니다.



5. 검증하기 단계 [Test]

1) 두더지 잡기 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 두더지 잡기 앱의 장점을 적어주세요.

3) 두더지 잡기 앱에 추가하면 좋을 내용을 적어주세요.

7. 사칙연산 앱

학습목표: 변수를 이용하여 사칙연산 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 사칙연산 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 화면에 처음 보일 때 테스트영역과 버튼들영역이 보이도록 지정하고, 변수 값을 초기화 합니다.



The code block for screen initialization consists of the following blocks:

- 언제 Screen1 .초기화
- 실행 지정하기 테스트영역 . 보이기 값 참
- 실행 지정하기 버튼들영역 . 보이기 값 참
- 전역변수 초기화 a 값 0
- 전역변수 초기화 b 값 0
- 전역변수 초기화 정답 값 0

(2) 덧셈버튼을 눌렀을 때 덧셈을 실행하는 블록입니다.



The code block for the addition button click event consists of the following blocks:

- 언제 덧셈버튼 .클릭
- 실행 지정하기 대답상자 . 텍스트 값 ""
- 실행 지정하기 결과레이블 . 텍스트 값 " 결과>> "
- 호출 더하기
- 함수 더하기
- 실행 지정하기 global a 값 임의의 정수 시작 1 끝 10
- 실행 지정하기 global b 값 임의의 정수 시작 1 끝 10
- 실행 지정하기 global 정답 값 가져오기 global a + 가져오기 global b
- 실행 지정하기 문제레이블 . 텍스트 값 합치기 가져오기 global a
" +
가져오기 global b
" = ? "

(3) 대담버튼을 정답을 확인하도록 하는 블록입니다.



5. 검증하기 단계 [Test]

1) 사칙연산 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 사칙연산 앱의 장점을 적어주세요.

3) 사칙연산 앱에 추가하면 좋을 내용을 적어주세요.

8. 권장칼로리 계산 앱

학습목표: 변수를 이용하여 권장칼로리 계산 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 권장칼로리 계산 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

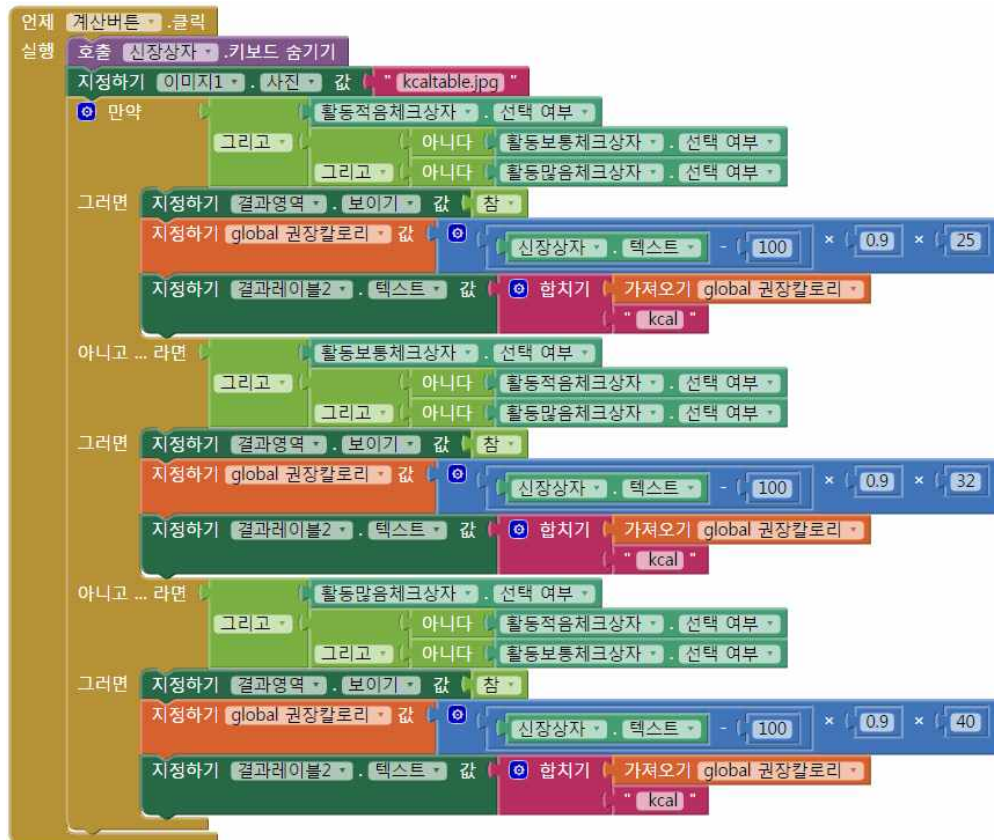
(1) 사용되는 변수와 화면 초기화를 합니다.



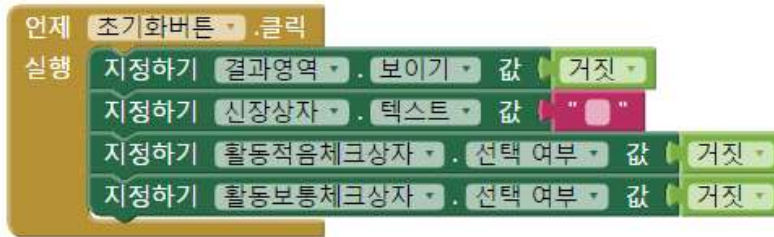
(2) 권장버튼을 클릭했을 때 체크영역, 입력영역, 버튼영역을 보이게 합니다.



(3) 계산버튼을 클릭하면 활동양에 따라 권장칼로리를 계산하도록 합니다.



(4) 초기화 버튼의 기능을 입력합니다.



5. 검증하기 단계 [Test]

1) 권장칼로리 계산 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 권장칼로리 계산 앱의 장점을 적어주세요.

3) 권장칼로리 계산 앱에 추가하면 좋을 내용을 적어주세요.

9. 비만도 계산 앱

학습목표: 변수를 이용하여 비만도 계산 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 비만도 계산 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 변수를 초기화 합니다.



(2) 계산하기 버튼으로 비만도를 계산합니다.



(3) 계산을 초기화하는 초기화버튼을 만듭니다.



5. 검증하기 단계 [Test]

1) 비만도 계산 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 비만도 계산 앱의 장점을 적어주세요.

3) 비만도 계산 앱에 추가하면 좋을 내용을 적어주세요.

10. 영어 단어 맞히기 앱

학습목표: 리시트를 이용하여 영어 단어 맞히기 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 영어 단어 맞히기 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.

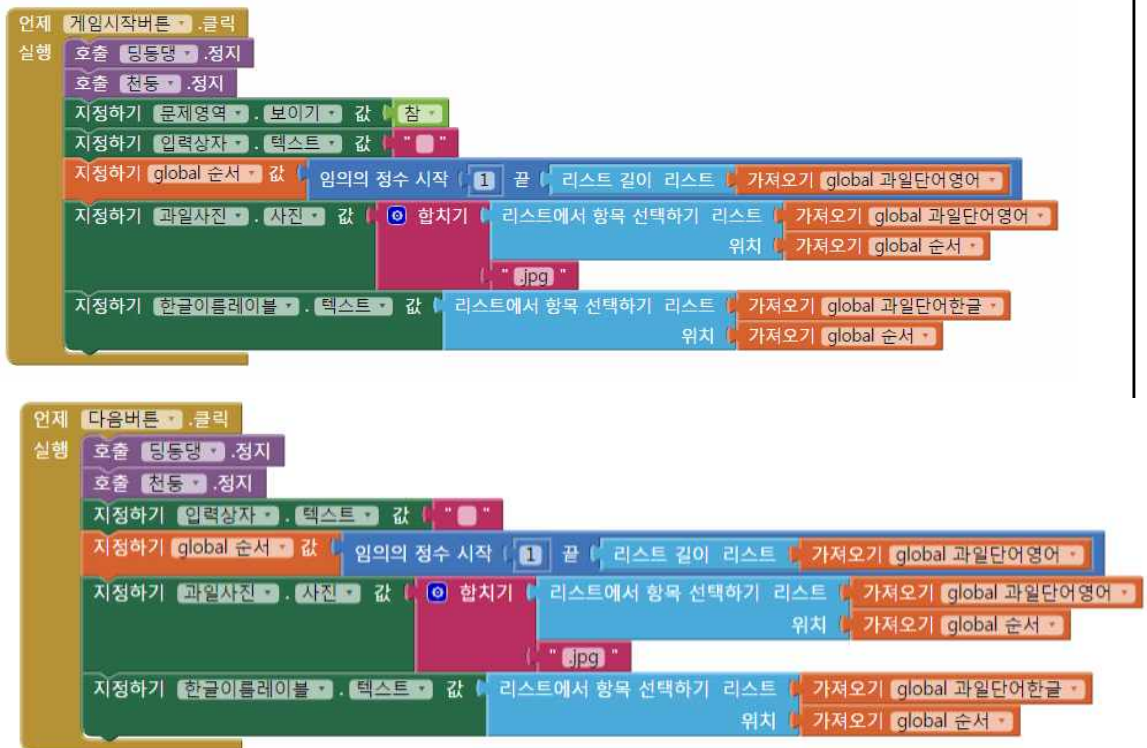


2) 블록으로 동작을 설정해 봅시다.

(1) 화면에 처음 보일 때 문제영역이 보이지 않게 하고, 리스트와 변수 값을 초기화 합니다.



(2) 게임시작버튼과 다음버튼의 기능을 입력합니다.



(3) 확인버튼의 기능을 입력합니다.



5. 검증하기 단계 [Test]

1) 영어 단어 맞히기 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 영어 단어 맞히기 앱의 장점을 적어주세요.

3) 영어 단어 맞히기 앱에 추가하면 좋을 내용을 적어주세요.

11. 만보계 앱

학습목표: Tiny DB의 기능을 이용하여 만보계 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 만보계 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 변수를 초기화하고 TinyDB 데이터값을 기록합니다.

```
전역변수 초기화 기록 값 0
전역변수 초기화 최고기록 값 0
언제 Screen1 초기화
실행 지정하기 카운터레이블 텍스트 값 0
지정하기 global 최고기록 값 호출 TinyDB1 값 가져오기
태그 "highscoreboardDB"
찾는 값이 없을 경우 0
호출 최고기록_업데이트
```

(2) 최고기록을 업데이트하는 함수를 설정합니다.

```
함수 최고기록_업데이트
실행 지정하기 global 최고기록 값 가져오기 global 기록
지정하기 최고기록레이블 텍스트 값 합치기 "최고 기록:"
가져오기 global 최고기록
호출 TinyDB1 값 저장
태그 "highscoreboardDB"
저장할 값 가져오기 global 최고기록
```

(3) 흔들림이 감지되면 1씩 커지면 숫자를 셉니다.

```
언제 가속도_센서1 흔들림
실행 지정하기 카운터레이블 텍스트 값
카운터레이블 텍스트 + 1
```

(4) 새로고침버튼으로 초기화 합니다.

```
언제 새로고침버튼 클릭
실행 지정하기 global 기록 값 카운터레이블 텍스트
지정하기 카운터레이블 텍스트 값 0
만약 가져오기 global 기록 > 가져오기 global 최고기록
그러면 호출 음성_변환1 말하기
메시지 "최고기록을 경신하셨습니다."
호출 최고기록_업데이트
아니면 지정하기 최고기록레이블 텍스트 값 합치기 "최고기록:"
가져오기 global 최고기록
```

5. 검증하기 단계 [Test]

1) 만보계 앱 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 만보계 앱의 장점을 적어주세요.

3) 만보계 앱에 추가하면 좋을 내용을 적어주세요.

12. SOS 구조 요청 앱

학습목표: Tiny DB의 기능을 이용하여 SOS 구조 요청 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- SOS 구조 요청 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 화면에 처음 보일 때 설정화면 입니다.

```

언제 Screen1 .초기화
실행 지정하기 Screen1 .배경색 값 색상 만들기 리스트 만들기 216
      191
      216
지정하기 이름상자 .너비 값 Screen1 .너비 × 0.4
지정하기 번호상자 .너비 값 Screen1 .너비 × 0.4
지정하기 관리영역 .보이기 값 거짓
지정하기 sos영역 .보이기 값 참
지정하기 이름상자 .텍스트 값 호출 TinyDB1 .값 가져오기
      태그 "sosnameDB"
      찾는 값이 없을 경우 ""
만약 리스트인가? 값 호출 TinyDB1 .값 가져오기
      태그 "sosphoneListDB"
      찾는 값이 없을 경우 ""
그러면 지정하기 global 전화번호 값 호출 TinyDB1 .값 가져오기
      태그 "sosphoneListDB"
      찾는 값이 없을 경우 ""
      호출 번호출력
  
```

(2) 번호출력 함수를 만듭니다.

```

함수 번호출력
실행 지정하기 번호레이블 .텍스트 값 ""
      각각 반복 항목 리스트 가져오기 global 전화번호
      실행 지정하기 번호레이블 .텍스트 값 합치기 번호레이블 .텍스트
      가져오기 항목
      " \n "
  
```

(3) SOS버튼을 누르면 긴급상황에 대한 메시지를 보내는 블록입니다.

```

언제 sos버튼 .클릭
실행 만약 아니다 리스트가 비어있나요? 리스트 가져오기 global 전화번호
      그러면 지정하기 문자메시지 .메시지 값 합치기 이름상자 .텍스트
      " 위급상황! 현재 위치: "
      위치센서 .현재 주소
      각각 반복 항목 리스트 가져오기 global 전화번호
      실행 지정하기 문자메시지 .전화번호 값 가져오기 항목
      호출 문자메시지 .메시지 보내기
  
```


(4) 관리버튼을 클릭하면 텍스트가 '설정'일 때와 '설정 닫기'일 때를 나누어 영역의 보임을 설정합니다.

```

언제 관리버튼 .클릭
실행 만약
  관리버튼 .텍스트 = "설정"
  그러면
    지정하기 sos영역 .보이기 값 거짓
    지정하기 관리영역 .보이기 값 참
    지정하기 관리버튼 .텍스트 값 "설정 닫기"
  아니면
    지정하기 sos영역 .보이기 값 참
    지정하기 관리영역 .보이기 값 거짓
    지정하기 관리버튼 .텍스트 값 "설정"
  
```

(5) 이름변경버튼 클릭했을 때 저장값을 변경합니다.

```

언제 이름변경버튼 .클릭
실행 호출 TinyDB1 .값 저장
  태그 "sosnameDB"
  저장할 값 이름상자 .텍스트
호출 이름상자 .키보드 숨기기
  
```

(6) 전화번호 추가, 삭제 블록들입니다.

```

전역변수 초기화 전화번호 값 빈 리스트 만들기

언제 추가버튼 .클릭
실행 만약
  번호상자 .텍스트 ≠ ""
  그리고 아니다 리스트에 포함되어 있나요? 값 번호상자 .텍스트
  리스트 가져오기 global 전화번호
  그러면
    리스트에 항목 추가하기 리스트 가져오기 global 전화번호
    item 번호상자 .텍스트
    호출 TinyDB1 .값 저장
    태그 "sosphoneListDB"
    저장할 값 가져오기 global 전화번호
    호출 번호출력
    호출 번호상자 .키보드 숨기기
    지정하기 번호상자 .텍스트 값 ""
  
```

```

언제 삭제목록선택 .선택 전
실행 지정하기 삭제목록선택 .요소 값 가져오기 global 전화번호
  
```

```

언제 삭제목록선택 .선택 후
실행 리스트에서 항목 삭제하기 리스트 가져오기 global 전화번호
      위치 삭제목록선택 .선택된 항목 번호
호출 TinyDB1 .값 저장
      태그 "sosphoneListDB"
      저장할 값 가져오기 global 전화번호
호출 번호출력

```

```

언제 전체삭제버튼 .클릭
실행 지정하기 global 전화번호 값 빈 리스트 만들기
호출 TinyDB1 .값 저장
      태그 "sosphoneListDB"
      저장할 값 가져오기 global 전화번호
지정하기 번호레이블 .텍스트 값 ""

```

5. 검증하기 단계 [Test]

1) SOS 구조 요청 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) SOS 구조 요청 앱의 장점을 적어주세요.

3) SOS 구조 요청 앱에 추가하면 좋을 내용을 적어주세요.

13. 운전 중 문자금지 앱

학습목표: Social 기능을 이용하여 운전 중 문자금지 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 운전 중 문자금지 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 현재위치 변수를 초기화 합니다.

전역변수 초기화 현재위치 값 "모르겠음"

(2) 응답메시지레이블에 텍스트 값을 TinyDB에서 가지고 옵니다.

언제 Screen1 초기화
실행 지정하기 응답메시지레이블 . 텍스트 값 호출 TinyDB1 . 값 가져오기 태그 responseMessage 찾는 값이 없을 경우 운전 중입니다. 곧 연락드릴게요

(3) TinyDB에 응답 내용을 바꿀 때 사용하는 응답바꾸기 버튼의 기능을 입력합니다.

언제 응답바꾸기버튼 . 클릭
실행 지정하기 응답메시지레이블 . 텍스트 값 응답메시지상자 . 텍스트 지정하기 응답메시지상자 . 텍스트 값 " " 호출 TinyDB1 . 값 저장 태그 responseMessage 저장할 값 응답메시지레이블 . 텍스트

(4) 문자메시지가 도착했을 때 상대방에게 보내는 메시지와 받은 메시지를 읽어 주는 기능을 입력합니다.

언제 위치_센서1 . 위치 변경
위도 경도 고도 속도
실행 지정하기 global 현재위치 값 위치_센서1 . 현재 주소

언제 문자_메시지1 . 메시지 받음
전화번호 메시지 텍스트
실행 지정하기 문자_메시지1 . 전화번호 값 가져오기 전화번호 지정하기 문자_메시지1 . 메시지 값 합치기 응답메시지레이블 . 텍스트 " 지금 위치는 " 가져오기 global 현재위치 " 입니다. " 호출 문자_메시지1 . 메시지 보내기 호출 음성_변환1 . 말하기 메시지 합치기 가져오기 전화번호 " 에서 문자가 왔습니다. 내용은 " 가져오기 메시지 텍스트 " 입니다. "

5. 검증하기 단계 [Test]

1) 운전 중 문자금지 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 운전 중 문자금지 앱의 장점을 적어주세요.

3) 운전 중 문자금지 앱에 추가하면 좋을 내용을 적어주세요.

14. 주문 앱

학습목표: Social 기능을 이용하여 주문 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 주문 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 화면에 처음 보일 때 선택하기영역이 보이지 않게 하고, 리스트와 변수 값을 초기화 합니다.



(2) 메뉴보기버튼을 클릭하면 메뉴레이블에 음식이름과 가격이 나타나도록 합니다.



(3) 메뉴선택하기버튼을 누르면 주문메뉴상자와 음식리스트의 값이 같으면 주문할 음식 리스트와 주문할 음식 가격 리스트에 추가합니다.

5. 검증하기 단계 [Test]

1) 주문 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 주문 앱의 장점을 적어주세요.

3) 주문 앱에 추가하면 좋을 내용을 적어주세요.

15. 악기연주 앱

학습목표: 센서 기능을 이용하여 악기연주 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 악기연주 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

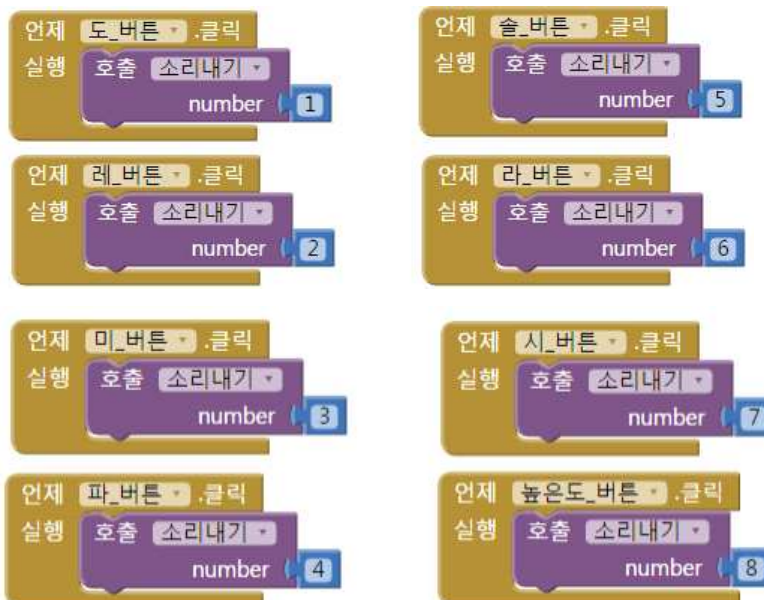
(1) 리스트 및 화면 초기화 설정입니다.



(2) 소리내기 함수의 기능을 입력합니다.



(3) 게임 버튼의 기능을 입력합니다.



(4) 사용자가 연주한 내용을 듣는 연주 버튼의 기능을 입력합니다.



5. 검증하기 단계 [Test]

1) 악기연주 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 악기연주 앱의 장점을 적어주세요.

3) 악기연주 앱에 추가하면 좋을 내용을 적어주세요.

16. 초시계 앱

학습목표: 센서 기능을 이용하여 초시계 앱을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 초시계 앱에 어떤 기능이 필요한지 적어보세요.

4. 시안 만들기 단계 [Prototype]

- 1) 디자이너에서 컴포넌트를 구성해 봅시다.



2) 블록으로 동작을 설정해 봅시다.

(1) 사용되는 변수와 화면 초기화를 합니다.

Scratch code block for initialization. The '언제' (When) section contains 'Screen1 > .초기화' (Screen1 > .initialize). The '실행' (Do) section contains: '지정하기 Screen1 > .배경색 > 값 > []', '지정하기 시계1 > .타이머 활성화 여부 > 값 > 거짓', '전역변수 초기화 조 > 값 > 0', and '전역변수 초기화 기록 > 값 > 빈 리스트 만들기'.

(2) 시작정지버튼을 클릭했을 때 타이머를 활성화 여부를 결정하고 시작버튼 텍스트를 바꾼다.

Scratch code block for the '시작정지버튼 > .클릭' (Start/Stop button > .click) event. The '실행' (Do) section contains: '만약 > 시작정지버튼 > .텍스트 > = > 시작 > 참 > 지정하기 시계1 > .타이머 활성화 여부 > 값 > 참 > 지정하기 시작정지버튼 > .텍스트 > 값 > 멈춤 > 아니면 > 지정하기 시계1 > .타이머 활성화 여부 > 값 > 거짓 > 지정하기 시작정지버튼 > .텍스트 > 값 > 시작'.

(3) 시계 타이머가 실행될 때 0.01초씩 증가하여 초입력상자 텍스트에 기록한다.

Scratch code block for the '시계1 > .타이머' (Clock1 > .timer) event. The '실행' (Do) section contains: '지정하기 global 조 > 값 > 가져오기 global 조 > + > 0.01' and '지정하기 초입력상자 > .텍스트 > 값 > 가져오기 global 조'.

(4) 기록버튼을 클릭하면 기록리스트에 항목이 추가되고 기록레이블의 텍스트에 기록한다.

Scratch code block for the '기록버튼 > .클릭' (Record button > .click) event. The '실행' (Do) section contains: '리스트에 항목 추가하기 리스트 > 가져오기 global 기록 > item > 가져오기 global 조 > 지정하기 기록레이블 > .텍스트 > 값 > []', '각각 반복 항목 리스트 > 가져오기 global 기록 > 실행 지정하기 기록레이블 > .텍스트 > 값 > 합치기 > 기록레이블 > .텍스트 > 가져오기 항목 > []'.

5. 검증하기 단계 [Test]

1) 초시계 앱이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 초시계 앱의 장점을 적어주세요.

3) 초시계 앱에 추가하면 좋을 내용을 적어주세요.

<부록 3> 마이크로비트를 활용한 디자인씽킹 기반 소프트웨어 교육 교재

마이크로비트를 활용한
디자인씽킹 기반 소프트웨어 교육

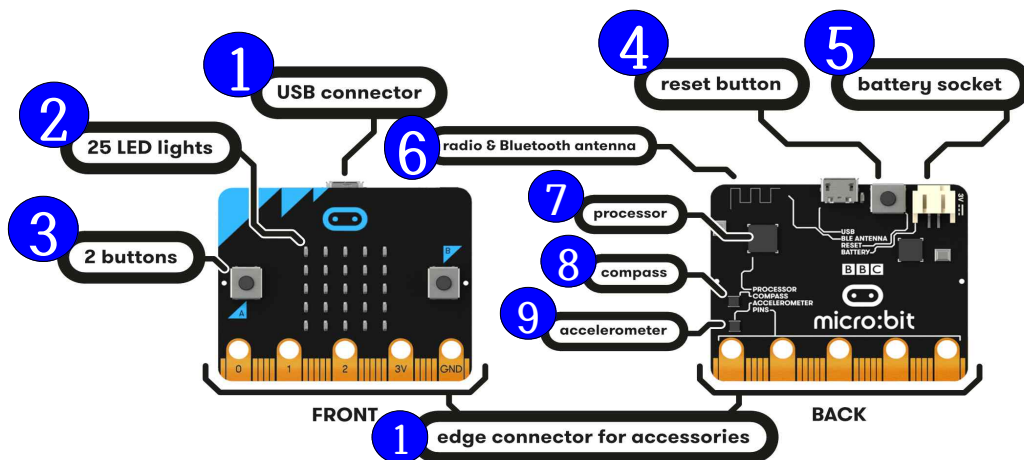
1. 마이크로비트 기초 기능 이해하기

1. 마이크로비트(Micro:bit)란?

1) 마이크로비트(Micro:bit)

마이크로비트는 아주 단순한 컴퓨터라고 할 수 있음. 컴퓨터는 입력을 받고, 이를 처리하여, 결과를 보여주는 기계인데 마이크로비트는 아주 작은 기판의 형태이지만 이 세 가지 기능을 모두 포함하고 있음. 작고 단순한 마이크로비트는 내가 코딩하여 만든 프로그램을 컴퓨터와 같이 실행시켜보고, 직접 만져 조작해보는 것을 가능하게 함.

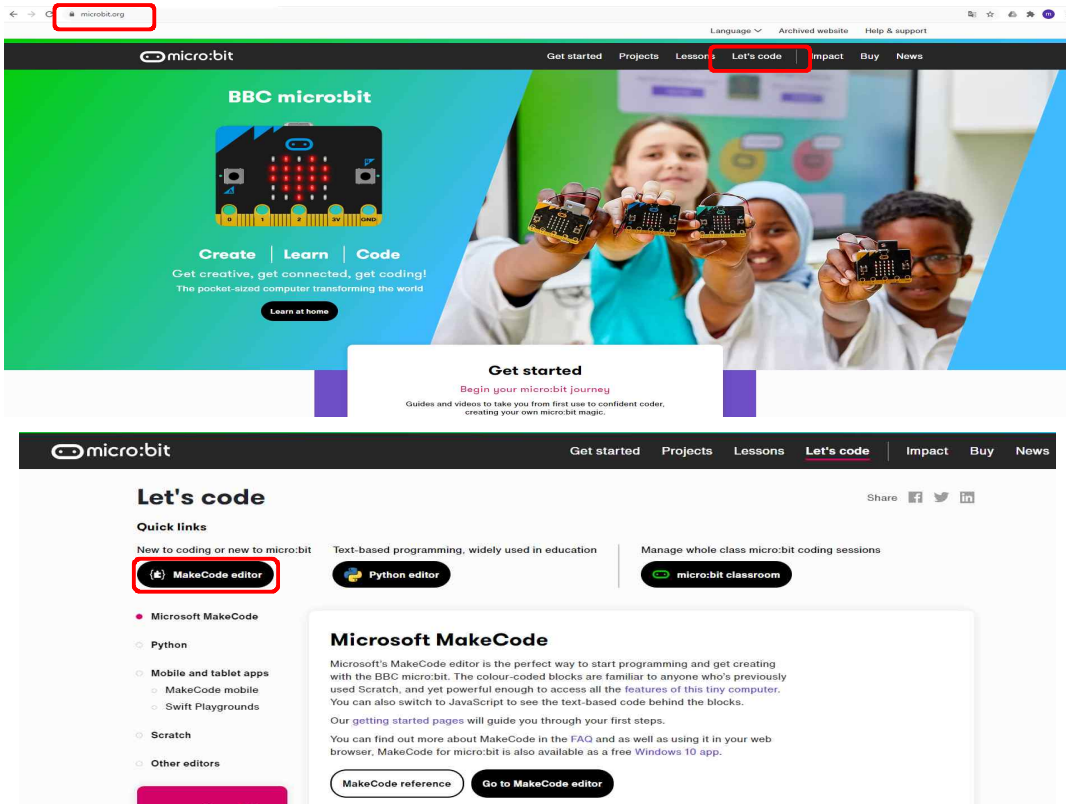
<마이크로비트 구성>



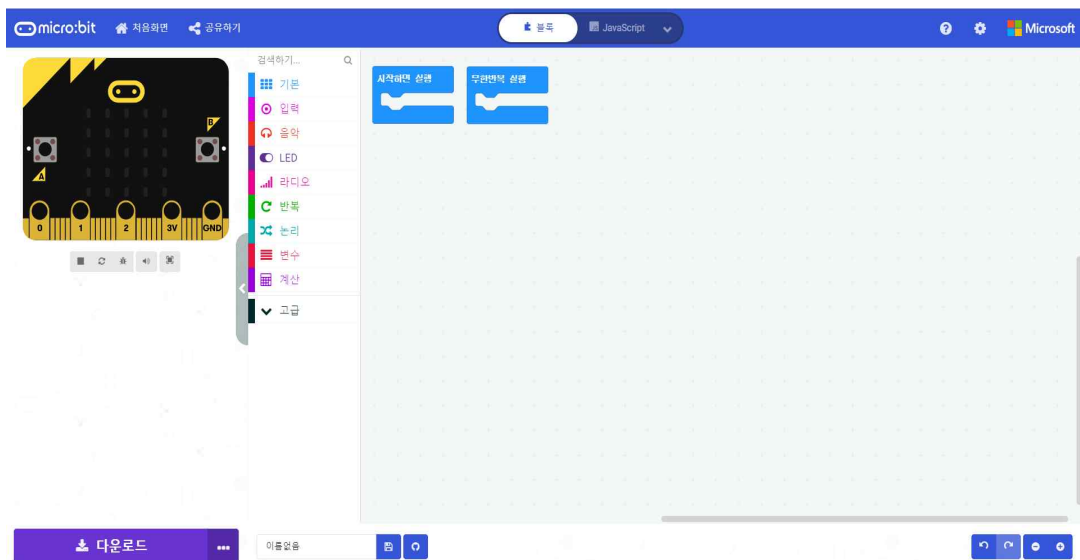
① USB케이블로 컴퓨터와 연결할 수 있음	⑥ 블루투스 안테나로 컴퓨터, 스마트폰 등과 연결할 수 있음
② LED램프이며 왼쪽 상단이 (0,0)좌표	⑦ 프로세서
③ A 버튼, B 버튼이며 프로그래밍이 가능함	⑧ 컴퍼스(나침반) 센서
④ 재시작 버튼	⑨ 가속도센서
⑤ 배터리 팩을 연결할 때 사용	⑩ 센서 연결 핀

2) Micro:bit 프로그래밍(블록 코드 편집기)

http://microbit.org 접속 - Let's Code 클릭 - MakeCode editor 클릭



< Micro:bit 블록 코드 편집기 화면 구성 >



< Micro:bit 명령블록 >

기본	문자, 숫자 출력/LED 모두 끄기 등의 기본 기능을수행하는 명령어 모음
입력	버튼/흔들기/센서 등의 외부 입력과 관련된 명령어 모음
음악	소리와 음악을 출력하는 명령어 모음
LED	마이크로비트의 25개 LED 출력을 제어하는 명령어 모음
라디오	블루투스를 이용하여 다른 마이크로비트와 데이터를 송수신하는 명령어 모음
반복	반복 수행과 관련된 명령어 모음
논리	논리적 판단을 추가하고자 할 때 사용하는 명령어 모음
변수	데이터를 저장하는 변수를 관리하는 명령어 모음
계산	수학적인 연산과 관련된 명령어 모음

< Micro:bit.hex 파일 저장 방법_webUSB>

< web USB 사용 환경 >



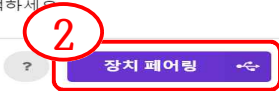
한 번 클릭으로 프로그램이 업로드 되도록 페어링하기



1 micro:bit 를 USB 케이블로 컴퓨터에 연결합니다. micro:bit 의 뒷 부분에 있는 microUSB 포트 를 사용합니다.



2 micro:bit 페어링 아래에 있는 '장치 페어링' 을 클릭한 후, BBC micro:bit CMSIS-DAP 나 DAPLink CMSIS-DAP 를 선택하세요.



2. 디자인씽킹 이해하기

학습목표: 디자인씽킹을 이용한 문제 해결 절차를 알 수 있다.

1. 디자인씽킹이란?

- 1) 디자이너가 생각하는 방식으로 문제를 해결하는 방법
→ 새로운 시각으로 세상을 바라보고 예전에는 몰랐던 방식으로 문제 해결
- 2) 사람을 중심으로 하는 창의적인 생각하는 태도
→ 자신과 타인을 더욱 잘 이해할 수 있고 삶의 융통성 증가

3. 디자인씽킹의 절차

노인 문제 해결 프로젝트

<1단계: 공감 단계> 해결해야 할 문제들이 무엇인가 고민하는 단계

※ observe(관찰), interview(면담), immerse(경험) / 사용자의 요구 파악(선택 사항이 있는 질문지, 가설의 증명여부 ×)

- 인터뷰: '생활하면서 불편한 점은?' '평소에 아드님 댁을 자주 가세요?'
- 노인에 대한 선입견: 나이가 많으니 일하기 싫어할 것이며, 무척 단조로운 생활을 할 것이다.



<2단계: 정의 단계> 명확하게 어떤 것이 문제인지 정의 내리는 단계

※ real(진짜 문제인가), valuable(해결할 만한 가치가 있는가), Inspiring (영감을 주는가?)

- 노인: 나이가 많고 체력이 약하지만 도움이 주는 존재가 되어 자신의 가치 증명 원함
- 젊은이: 노인들이 힘들거나 다칠까봐 호의 대부분을 거절
→ 왜 갈등이 생겼는지 생각. 어떻게 하면 노인들이 자신의 가치를 증명하는 일을 도울 수 있을까?



<3단계: 아이디어 발상 단계> 문제에 대한 해결방법을 찾는 단계

- 5분 안에 백 개 이상의 해결 방법 생각하기/ 많은 아이디어 만들기
- 떠오른 아이디어를 즉시 노트에 적거나 그린 후, 크게 소리내어 읽기



<4단계: 시안 만들기 단계> 손을 움직여 아이디어에 등장했던 물건 만들기

- 좋은 아이디어는 시각화, 구체화가 필요하므로 시제품으로 만들/ 빠르고 저렴해야 함/ 소통, 협업 등이 이루어짐.
- 손을 움직여 아이디어 도출 단계에서 나왔던 물건 만들기(노인용 텔레비전, 기내 표시등, 시각 장애인용 스위치)
- 추상적인 아이디어가 좀 더 구체화 됨.
(노인용 기내 표시등- 노인들 시력 - 글자체 크게)



<5단계: 검증하기 단계> 생산된 시제품을 사용자들로부터 검증 받는 단계

- 만든 시제품을 사용자에게 사용하게 함으로서 문제를 재정비함.
- 필요시 다시 그 전 단계로 돌아감.

3. 자동 전등 프로그램

학습목표: 빛 센서와 LED를 활용하여 자동 전등 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 자동 전등 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 자동 전등 프로그램을 만들어 봅시다.

- 1) 센서값을 출력하기 위한 준비를 합니다.



2) 빛 센서를 활용하여 빛 센서의 밝기가 30 미만일 때 LED가 켜지게 합니다.



3) A 버튼을 누르면 LED 밝기가 점점 밝아지고, B 버튼을 누르면 LED 밝기가 점점 어두워집니다.



5. 검증하기 단계 [Test]

1) 자동 전등 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 자동 전등 프로그램의 장점을 적어주세요.

3) 자동 전등 프로그램에 추가하면 좋을 내용을 적어주세요.

4. 걱정 온도 알림 프로그램

학습목표: 온도 센서와 LED를 활용하여 걱정 온도 알림 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 걱정 온도 알림 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 걱정 온도 알림 프로그램을 만들어 봅시다.

- 1) 센서값을 출력하기 위한 준비를 합니다.



2) 온도 센서 값이 15도 이상 20도 이하일 때는 '행복' 아이콘 출력, 그 외의 온도일 때는 '슬픔' 아이콘을 출력합니다.



5. 검증하기 단계 [Test]

1) 자동 전등 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 자동 전등 프로그램의 장점을 적어주세요.

3) 자동 전등 프로그램에 추가하면 좋을 내용을 적어주세요.

5. 나침반 프로그램

학습목표: 자기 센서와 LED를 활용하여 나침반 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 나침반 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 나침반 프로그램을 만들어 봅시다.

- 1) 자기센서값을 0-360으로 표시합니다.



2) 자기센서값에 따라서 동서남북의 방위를 표시합니다.



5. 검증하기 단계 [Test]

1) 나침반 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 나침반 프로그램의 장점을 적어주세요.

3) 나침반 프로그램에 추가하면 좋을 내용을 적어주세요.

6. 타이머 프로그램

학습목표: 스피커와 LED를 활용하여 타이머 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

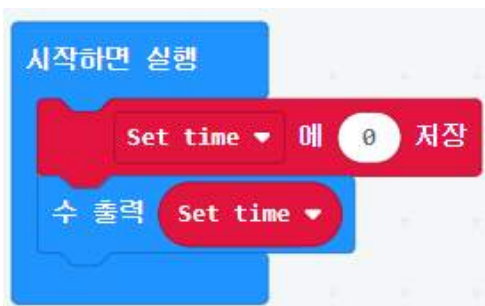
3. 아이디어 발상 단계 [Ideate]

- 타이머 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 타이머 프로그램을 만들어 봅시다.

- 1) 변수를 초기화 합니다.



2) A와 B버튼을 누르면 시간을 설정할 수 있도록 합니다.



3) A+B버튼을 누르면 1초 줄어들게 하고, 시간이 다 되면 멜로디가 나오게 합니다.



5. 검증하기 단계 [Test]

1) 타이머 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 타이머 프로그램의 장점을 적어주세요.

Empty rounded rectangular box for writing the advantages of the timer program.

3) 타이머 프로그램에 추가하면 좋을 내용을 적어주세요.

Empty rounded rectangular box for writing content that would be good to add to the timer program.

7. 통신하기 프로그램

학습목표: 라디오 기능을 활용하여 통신하기 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

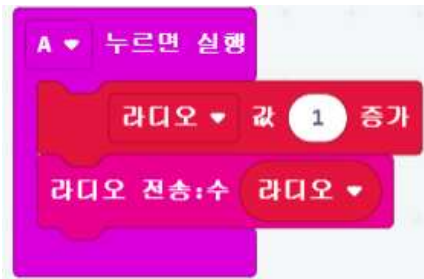
- 통신하기 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 통신하기 프로그램을 만들어 봅시다.
 - 1) 라디오 그룹과 라디오 변수를 초기화 합니다.



2) A버튼을 누르면 라디오 변수값이 1씩 증가하고, 라디오 변수값을 그룹 친구에게 보냅니다.



3) 라디오 기능을 통해 숫자를 수신하면 아이콘을 출력합니다.



5. 검증하기 단계 [Test]

1) 통신하기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 통신하기 프로그램의 장점을 적어주세요.

3) 통신하기 프로그램에 추가하면 좋을 내용을 적어주세요.

8. 도난 방지 프로그램

학습목표: 충돌 센서와 LED를 활용하여 도난 방지 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 도난 방지 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 도난 방지 프로그램을 만들어 봅시다.

- 1) 센서값을 받을 수 있도록 시리얼통신을 초기화 합니다. 충돌 센서를 P1, LED를 P2에 연결하고 LED를 끕니다.



- 2) 충돌 센서에 물건이 놓여 있어 눌러 있으면 LED를 끄고, 물건을 가져가면 충돌 센서가 눌러지지 않아 LED를 켜서 비상 상황을 알립니다.



5. 검증하기 단계 [Test]

- 1) 도난 방지 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

- 2) 도난 방지 프로그램의 장점을 적어주세요.

- 3) 도난 방지 프로그램에 추가하면 좋을 내용을 적어주세요.

9. 식물 습도 알림 프로그램

학습목표: 습도 센서와 OLED를 활용하여 식물 습도 알림 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 식물 습도 알림 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

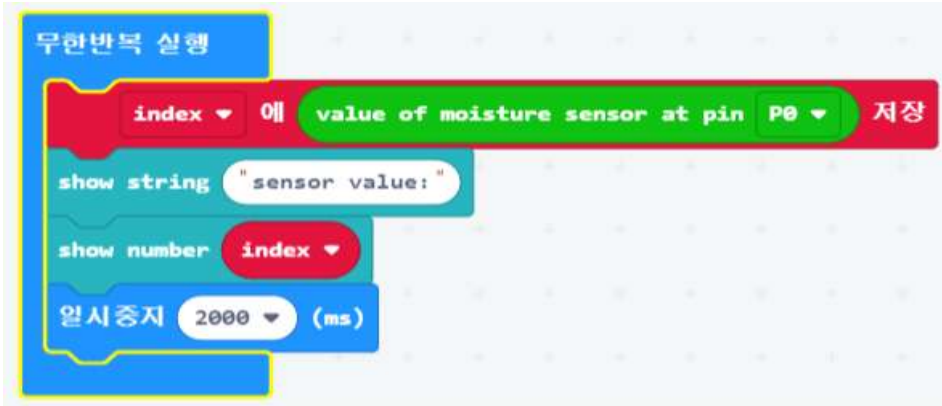
4. 시안 만들기 단계 [Prototype]

- 식물 습도 알림 프로그램을 만들어 봅시다.

- 1) OLED를 화면 크기를 설정합니다.



2) 습도 센서를 P0에 연결하고 OLED에 습도값을 2초에 한번씩 보여줍니다.



3) 습도값이 50보다 작으면 슬픔 아이콘, 50 이상이면 웃는 아이콘을 보여줍니다.



5. 검증하기 단계 [Test]

1) 식물 습도 알림 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 식물 습도 알림 프로그램의 장점을 적어주세요.

Empty rounded rectangular box for writing the advantages of the plant humidity notification program.

3) 식물 습도 알림 프로그램에 추가하면 좋을 내용을 적어주세요.

Empty rounded rectangular box for writing additional content that would be good to add to the plant humidity notification program.

10. 전자 피아노 프로그램

학습목표: AD키보드와 스피커를 활용하여 사회적 거리두기 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

3. 아이디어 발상 단계 [Ideate]

- 전자 피아노 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 전자 피아노 프로그램을 만들어 봅시다.

- 1) AD키보드를 P1에 연결하고, 스피커를 P0에 연결합니다. AD키보드 5개와 마이크로비트 자체 키보드를 이용하여 도~시까지 표현합니다.



5. 검증하기 단계 [Test]

1) 전자 피아노 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 전자 피아노 프로그램의 장점을 적어주세요.

Empty rounded rectangular box for writing the advantages of the digital piano program.

3) 전자 피아노 프로그램에 추가하면 좋을 내용을 적어주세요.

Empty rounded rectangular box for writing content that would be good to add to the digital piano program.

11. 움직임 감지 프로그램

학습목표: 움직임 감지 센서와 LED를 활용하여 움직임 감지 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

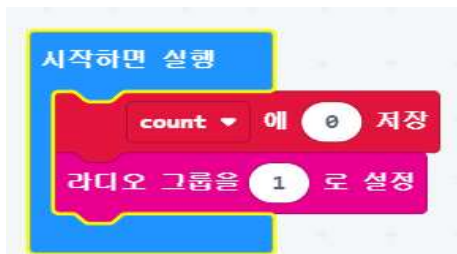
3. 아이디어 발상 단계 [Ideate]

- 움직임 감지 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

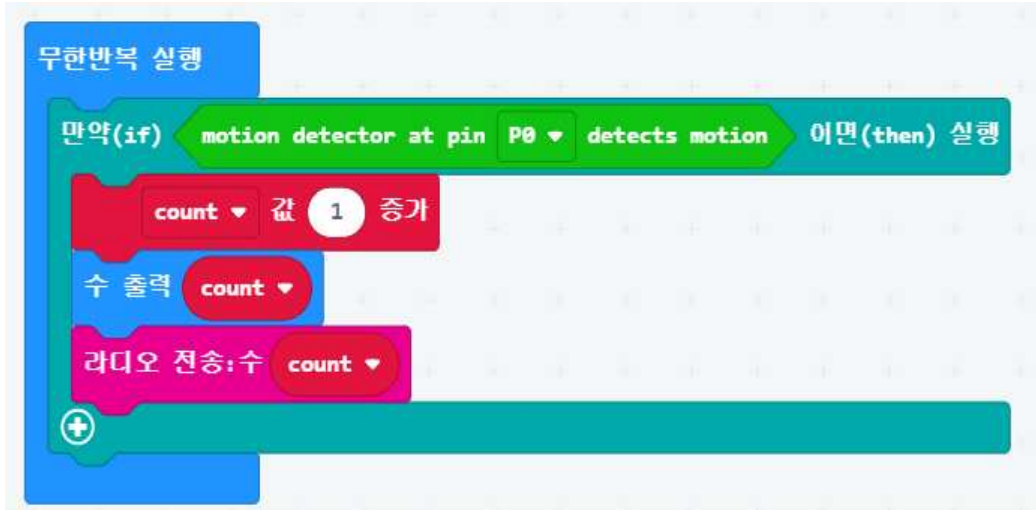
4. 시안 만들기 단계 [Prototype]

- 움직임 감지 프로그램을 만들어 봅시다.

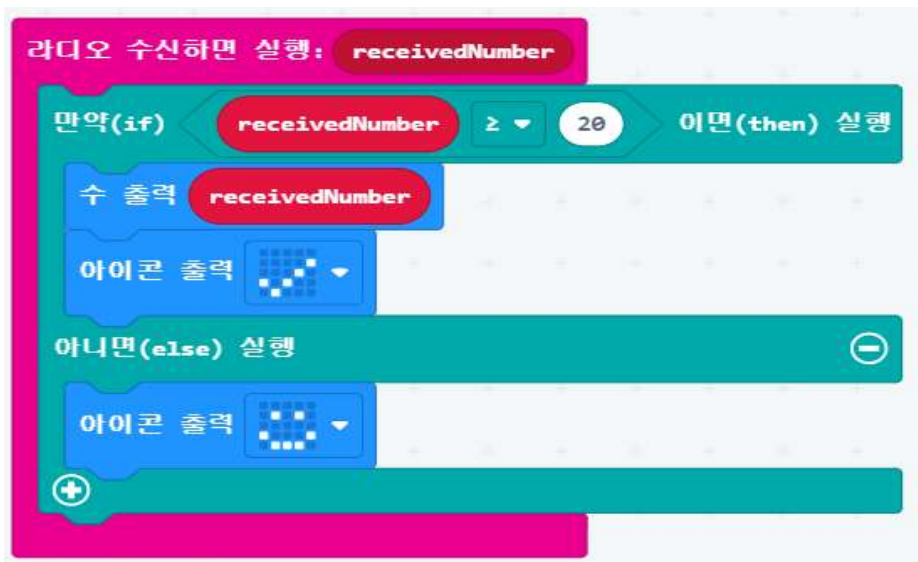
- 1) 변수값을 count로 저장하고 라디오 그룹을 1로 설정한다.



2) 움직임이 감지되면 변수값이 1씩 증가하고 이 값을 라디오로 전송합니다.



3) 수신값이 일정량 이상이 되면 마이크로비트 LED로 표시합니다.



5. 검증하기 단계 [Test]

1) 움직임 감지 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

Empty rounded rectangular box for answer to question 1.

2) 움직임 감지 프로그램의 장점을 적어주세요.

Empty rounded rectangular box for answer to question 2.

3) 움직임 감지 프로그램에 추가하면 좋을 내용을 적어주세요.

Empty rounded rectangular box for answer to question 3.

12. 사회적 거리두기 프로그램

학습목표: 초음파 센서와 스피커를 활용하여 사회적 거리두기 프로그램을 만들 수 있다.

1. 공감 단계 [Empathize]

- 어떤 문제를 가지고 있는 사람들에게 도움이 될지 적어보세요.

2. 문제 정의 단계 [Define]

- 어떤 문제를 해결할 것인지 적어보세요.

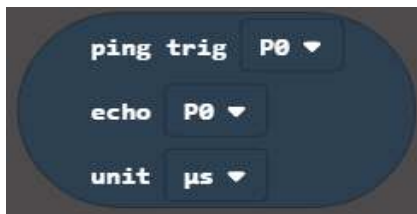
3. 아이디어 발상 단계 [Ideate]

- 사회적 거리두기 프로그램에 사용될 센서와 액추에이터를 생각해 보세요.

4. 시안 만들기 단계 [Prototype]

- 사회적 거리 유지를 위한 프로그램을 만들어 봅시다.

- 1) 초음파 센서 블록을 가져온다.



- 2) '거리' 변수를 설정하고 거리값을 바꾼다.



3) 조건문 활용하여 일정 거리 안에 사물이 감지되면 소리를 출력하고 해당 없으면 아이콘을 출력한다.



4) 거리가 가까워질수록 소리가 더 빠르게 나도록 설정한다.



5) OLED 기능을 추가하여 거리를 보여준다.



5. 검증하기 단계 [Test]

1) 사회적 거리두기 프로그램이 오류 없이 잘 동작하나요? 동작하지 않는다면 어떻게 해결해야 할까요?

2) 사회적 거리두기 프로그램의 장점을 적어주세요.

3) 사회적 거리두기 프로그램에 추가하면 좋을 내용을 적어주세요.