



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

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

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



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

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

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

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

[Disclaimer](#)

박사학위논문

컴퓨팅 사고력 신장을 위한
프로그래밍 중심 STEAM 교육 프로그램



제주대학교 대학원

과학교육학부 컴퓨터교육전공

김태훈

2015년 2월

박사학위논문

컴퓨팅 사고력 신장을 위한
프로그래밍 중심 STEAM 교육 프로그램



제주대학교 대학원

과학교육학부 컴퓨터교육전공

김태훈

2015년 2월

컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램

指導教授 金 鍾 勳

金 泰 勳

이 論文을 教育學 博士學位 論文으로 提出함

2014년 12월

김태훈의 教育學 博士學位 論文을 認准함

審査委員長	김 성 백	(印)
委 員	박 찬 정	(印)
委 員	이 재 무	(印)
委 員	이 민 제	(印)
委 員	김 종 훈	(印)

濟州大學校 大學院

2014年 12月

STEAM Education Program based on Programming to Improve Computational Thinking Ability

TaeHun Kim

(Supervised by professor JongHoon Kim)

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

2014. 12.



This thesis has been examined and approved.

Major of Computer Education

Faculty of Science Education

GRADUATE SCHOOL

JEJU NATIONAL UNIVERSITY

목 차

표 목 차	iv
그림목차	ix
국문초록	x
I. 서론	1
1. 연구의 필요성	1
2. 연구의 내용	3
3. 연구의 기대 효과	4
4. 연구의 제한점	5
II. 이론적 배경	7
1. 컴퓨팅 사고력(Computational Thinking)	7
1) 컴퓨팅 사고력의 정의	7
2) 컴퓨팅 사고력의 구성 요소	11
3) 컴퓨팅 사고력 검사 도구	12
2. STEAM 교육	14
1) 등장 배경	15
2) STEAM 교육의 요소	18
3) 프로그램 개발 모형	20
3. 프로그래밍 교육	23
1) 컴퓨터 과학 교육과정과 프로그래밍	23
2) 프로그래밍과 STEAM 교육	24
3) 교육용 프로그래밍 언어	26
4) 프로그래밍 교수 전략	35
III. 프로그래밍 중심 STEAM 교육 프로그램	39
1. STEAM 교육 프로그램을 위한 요구 분석	39

1) 요구 분석 목적 결정	40
2) 출처 확인	40
3) 도구 선택	41
4) 요구 분석 실시	41
5) 요구 분석 의사 결정	44
2. STEAM 교육 프로그램의 설계	45
1) 교육과정 분석	45
2) 학습준거 선정	49
3) 통합유형 선정	51
4) 수업모형 선정	52
3. STEAM 교육 프로그램의 개발	54
1) 학습 주제 및 내용 선정	54
2) 교육 프로그램 진행 단계에 따른 개발	56
4. STEAM 교육 프로그램의 적용	59
1) 코듀 프로그래밍 중심 STEAM 교육 프로그램	59
2) 스크래치 프로그래밍 중심 STEAM 교육 프로그램	59
(1) 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램	59
(2) 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램	60
(3) 정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램	60
5. STEAM 교육 프로그램의 평가	60
1) 검사 도구	61
(1) 창의력 검사도구	61
(2) 논리적 사고력 검사도구	63
(3) 과학과 관련된 정의적 영역 검사도구	64
IV. STEAM 교육 프로그램 개발의 실제	66
1. 코듀 프로그래밍 중심의 STEAM 교육 프로그램	66
1) 요구분석	66
2) 설계 및 개발	66
3) 적용 및 효과분석	69

2. 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램	85
1) 요구 분석	85
2) 설계 및 개발	85
3) 적용 및 효과 분석	89
3. 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램	105
1) 요구 분석	105
2) 설계 및 개발	105
3) 적용 및 효과 분석	107
4. 정규 교육과정에서의 스크래치 프로그래밍 중심의 STEAM 교육 프로그램	119
1) 요구분석	119
2) 설계 및 개발	120
3) 적용 및 효과 분석	122
 V. 결론 및 제언	 138
 참고문헌	 143
 Abstract	 152
 부록	 155



표 목 차

<표 I-1> ISD 모형 단계에 따른 연구 방법 및 산출물	4
<표 II-1> 컴퓨팅 사고 관련 국내·외 관련 연구	9
<표 II-2> 컴퓨팅 사고력 측정 도구 개발 관련 연구	13
<표 II-3> ADDIE 모형의 단계별 특성	22
<표 II-4> 스크래치 프로그래밍 관련 연구	26
<표 II-5> 코딩을 활용한 프로그래밍 교육에 대한 연구	30
<표 II-6> 투입 내용	34
<표 III-1> 정보원과 정보 내용	41
<표 III-2> 프로그래밍 경험	42
<표 III-3> 프로그래밍 학습에 대한 흥미와 자신감	43
<표 III-4> 과학 학습에 대한 흥미와 자신감	43
<표 III-5> STEAM 교육 경험	44
<표 III-6> STEAM 교육에 대한 흥미	44
<표 III-7> 2009 개정 과학과 교육과정 내용(4~6학년)	46
<표 III-8> 2009 개정 과학과 4학년 교육과정 분석 결과	47
<표 III-9> 2009 개정 과학과 5학년 교육과정 분석 결과	47
<표 III-10> 2009 개정 과학과 6학년 교육과정 분석 결과	48
<표 III-11> 학습준거 선정	49
<표 III-12> 내용 요소 선정	50
<표 III-13> 순환 학습 모형	53
<표 III-14> 교육 프로그램 선정 주제(1차)	55
<표 III-15> 교육 프로그램 선정 주제(2차)	55
<표 III-16> GALT 검사지 축소본의 문항 구성	64
<표 III-17> 과학과 관련된 정의적 영역 평가 세부요소	65
<표 IV-1> 선정 주제	67
<표 IV-2> 교육 프로그램의 단계별 구성	67

<표 IV-3> 학습준거 선정	68
<표 IV-4> 내용 요소 선정	69
<표 IV-5> 연구 대상	70
<표 IV-6> 프로그램 일정	71
<표 IV-7> 실험 설계	71
<표 IV-8> 실험집단 창의력 검사 정규성 검정	72
<표 IV-9> 비교집단 창의력 검사 정규성 검정	73
<표 IV-10> 창의력 사전검사 결과 비교(독립표본 t검정)	74
<표 IV-11> 창의력 사전검사 결과 비교(Mann-Whitney U검정)	74
<표 IV-12> 창의력 사후검사 결과 비교(독립표본 t검정)	74
<표 IV-13> 집단에 따른 독창성, 창의력 평균, 창의력 지수에 대한 서술통계	75
<표 IV-14> 집단에 따른 교정된 하위요소에 대한 공분산분석 결과	76
<표 IV-15> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	77
<표 IV-16> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)	77
<표 IV-17> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	78
<표 IV-18> 실험집단 논리적 사고력 검사 정규성 검정	79
<표 IV-19> 비교집단 논리적 사고력 검사 정규성 검정	79
<표 IV-20> 논리적 사고력 사전검사 결과 비교(독립표본 t검정)	80
<표 IV-21> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)	80
<표 IV-22> 논리적 사고력 사후검사 결과 비교(독립표본 t검정)	80
<표 IV-23> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)	81
<표 IV-24> 실험집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 T 검정)	82
<표 IV-25> 실험집단 논리적 사고력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)	82
<표 IV-26> 비교집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정)	82
<표 IV-27> 비교집단의 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정)	83
<표 IV-28> 과학과 관련된 정의적 영역 검사 정규성 검정	83
<표 IV-29> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교	84
<표 IV-30> 실험의 결과 분석	84
<표 IV-31> 선정 주제	86

<표 IV-32> 교육 프로그램의 단계별 구성	87
<표 IV-33> 학습 모형	87
<표 IV-34> 연구 대상	90
<표 IV-35> 프로그램 일정	91
<표 IV-36> 실험 설계	91
<표 IV-37> 실험집단 창의력 검사 정규성 검정	92
<표 IV-38> 비교집단 창의력 검사 정규성 검정	92
<표 IV-39> 창의력 사전검사 결과 비교(독립표본 t검정)	93
<표 IV-40> 창의력 사전검사 결과 비교(Mann-Whitney U검정)	93
<표 IV-41> 집단에 따른 유창성, 창의력 평균, 창의력 지수에 대한 서술통계	95
<표 IV-42> 집단에 따른 교정된 하위요소에 대한 공분산분석 결과	95
<표 IV-43> 창의력 사후검사 결과 비교(Mann-Whitney U검정)	96
<표 IV-44> 실험집단 창의력 사전·사후검사 결과 비교(독립표본 t검정)	96
<표 IV-45> 실험집단의 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)	97
<표 IV-46> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	97
<표 IV-47> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)	98
<표 IV-48> 실험집단 논리적 사고력 검사 정규성 검정	99
<표 IV-49> 비교집단 논리적 사고력 검사 정규성 검정	99
<표 IV-50> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)	100
<표 IV-51> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)	101
<표 IV-52> 집단별 논리적 사고력 검사결과 비교(Wilcoxon 부호순위 검정)	102
<표 IV-53> 과학과 관련된 정의적 영역 검사 정규성 검정	103
<표 IV-54> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교	104
<표 IV-55> 실험의 결과 분석	104
<표 IV-56> 선정 주제	106
<표 IV-57> 교육 프로그램의 단계별 구성	106
<표 IV-58> 연구 대상	108
<표 IV-59> 프로그램의 투입 기간	109
<표 IV-60> 실험 설계	109

<표 IV-61> 실험집단 창의력 검사 정규성 검정	110
<표 IV-62> 비교집단 창의력 검사 정규성 검정	110
<표 IV-63> 창의력 사전검사 결과 비교(독립표본 t검정)	111
<표 IV-64> 창의력 사전검사 결과 비교(Mann-Whitney U검정)	112
<표 IV-65> 창의력 사후검사 결과 비교(독립표본 t검정)	112
<표 IV-66> 창의력 사후검사 결과 비교(Mann-Whitney U검정)	113
<표 IV-67> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	113
<표 IV-68> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정) ..	114
<표 IV-69> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	114
<표 IV-70> 비교집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정) ..	115
<표 IV-71> 실험집단 논리적 사고력 검사 정규성 검정	115
<표 IV-72> 비교집단 논리적 사고력 검사 정규성 검정	116
<표 IV-73> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)	117
<표 IV-74> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)	117
<표 IV-75> 집단별 논리적 사고력 결과 비교(Wilcoxon 부호순위 검정)	118
<표 IV-76> 실험의 결과 분석	119
<표 IV-77> 교과별 차시 운영 계획	120
<표 IV-78> 여러 가지 기체의 쓰임 주제 개요	121
<표 IV-79> 연구대상	123
<표 IV-80> 프로그램 투입 일정	124
<표 IV-81> 실험 설계	124
<표 IV-82> 실험집단 창의력 검사 정규성 검정	125
<표 IV-83> 비교집단 창의력 검사 정규성 검정	125
<표 IV-84> 창의력 사전검사 결과 비교(독립표본 t검정)	126
<표 IV-85> 창의력 사전검사 결과 비교(Mann-Whitney U검정)	126
<표 IV-86> 창의력 사후검사 결과 비교(독립표본 t검정)	127
<표 IV-87> 창의력 사후검사 결과 비교(Mann-Whitney U검정)	127
<표 IV-88> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)	128
<표 IV-89> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정) ..	128

<표 IV-90> 비교집단 창의적 사고력 사전·사후 결과 비교(대응표본 t검정) ……	129
<표 IV-91> 비교집단 창의적 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정) ……	129
<표 IV-92> 실험집단 논리적 사고력 검사 정규성 검정 ……	130
<표 IV-93> 비교집단 논리적 사고력 검사 정규성 검정 ……	130
<표 IV-94> 논리적 사고력 사전검사 결과 비교(독립표본 t검정) ……	131
<표 IV-95> 논리적 사고력 사전 결과 비교(Mann-Whitney U검정) ……	131
<표 IV-96> 논리적 사고력 사후검사 결과 비교(독립표본 t검정) ……	132
<표 IV-97> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정) ……	132
<표 IV-98> 실험집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정) ……	133
<표 IV-99> 실험집단 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정) ……	133
<표 IV-100> 비교집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정) ……	134
<표 IV-101> 비교집단 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정) ……	134
<표 IV-102> 과학과 관련된 정의적 영역 검사 정규성 검정 ……	135
<표 IV-103> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교 ……	135
<표 IV-104> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교(세부요소) ……	136
<표 IV-105> 실험의 결과 분석 ……	137

그림 목 차

[그림 II-1] 컴퓨팅 사고와 공유되는 사고들	8
[그림 II-2] STEAM 교육 프레임워크	17
[그림 II-3] ADDIE 모형의 과정	21
[그림 II-4] PDIE 절차 모형의 단계별 세부 절차	23
[그림 II-5] 코듀의 프로그래밍 예제	29
[그림 III-1] Rossett 모형	39
[그림 III-2] 통합 유형 분석 모형	51
[그림 III-3] 통합 유형 선정	52
[그림 III-4] 교육 프로그램 단계별 구성	56
[그림 III-5] 프로그래밍 학습 단계	57
[그림 III-6] STEAM 주제 학습 단계	57
[그림 IV-1] 프로그램 구상 및 학생 작품 예시	88
[그림 IV-2] 스크래치 소양 익히기 단계 학습 순서	121
[그림 IV-3] 디지털 스토리텔링 단계 학습 모형	122

<국문초록>

컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램

김 태 훈

제주대학교 대학원 과학교육학부 컴퓨터교육전공

지도교수 김 중 훈

이 연구의 목적은 초등학교 학습자의 컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램을 개발하고 이를 학교 현장에 적용하여 교육적 효과를 검증하는 것이다.

첨단 과학기술의 발전은 현대 사회를 빠르게 변화시키고 있다. 21세기는 환경오염, 식량부족, 에너지 고갈 등 융합적 지식과 사고를 기반으로 해야 하는 문제가 증가하고, 이러한 문제들은 기존의 지식 기반 접근방법으로 해결하기에는 한계가 있어 이를 해결하기 위해 지식을 넘나드는 융합적이고 창의적인 사고력과 실천력을 필요로 한다. 융합의 시대를 살아갈 학습자들이 융합 사회를 이끌어가는 핵심 인재로 성장하기 위해서는 융합적 소양과 융합의 핵심을 관통하는 컴퓨팅의 원리와 기술을 학습하여 창의적이고 효율적으로 문제를 해결하는 데 필요한 능력을 기르도록 해야 한다. 이에 본 연구에서는 학생들의 컴퓨팅 사고력과 융합적 소양을 신장시키기 위한 프로그래밍 중심 STEAM 교육 프로그램을 개발하였다.

본 연구에서는 프로그래밍 중심 STEAM 교육 프로그램을 개발하기 위하여 ADDIE 모형의 단계에 따라 연구를 진행하였다.

요구 분석 단계에서는 컴퓨팅 사고력과 STEAM 교육, 프로그래밍 교육에

대한 문헌연구 분석과 프로그래밍을 활용한 STEAM 교육에 대한 선행연구 분석, 학습자의 요구 분석을 진행하였다.

설계 단계에서는 분석 단계의 요구 분석을 반영한 교육 프로그램을 개발하기 위해 초등학교 2009 개정 교육과정을 분석하고 이를 바탕으로 학습준거를 선정하여 STEAM 각 교과요소의 역할을 결정하였다. 융합의 형태를 정하기 위해 통합유형을 선정하고 STEAM 교육 프로그램을 위하여 순환학습모형을 수업 모형으로 선정하였다.

개발 단계에서는 설계 단계에서 작성된 설계 내용을 바탕으로 4개의 프로그래밍 중심 STEAM 교육 프로그램을 개발하였다. 개발한 교육 프로그램은 교수전략과 투입 방법에 따라 코딩 프로그래밍 중심의 STEAM 교육 프로그램, 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램, 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램, 정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램으로 구분하였다.

적용 단계에서는 총 4회에 걸쳐 교육기부, 방과후학교, 정규수업을 통해 프로그래밍 중심 STEAM 교육 프로그램을 실험집단에게 투입하였고 일반적인 프로그래밍 학습 또는 전통적인 분과수업을 처치한 비교집단을 선정하여 프로그램을 적용하였다.

평가 단계에서는 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램의 교육적 효과를 살펴보기 위해 창의력, 논리적 사고력, 과학과 관련된 정의적 영역의 검사를 실시하여 그 결과를 분석하였다.

평가 결과, 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램을 학습한 실험집단이 일반적인 프로그래밍 수업이나 전통적인 분과수업을 학습한 비교집단에 비해 창의력, 논리적 사고력, 과학과 관련된 정의적 영역에서 유의하게 향상되었음을 확인하였다. 이러한 연구 결과는 프로그래밍 중심의 STEAM 교육 프로그램이 학습자의 컴퓨팅 사고력과 융합적 소양에 긍정적인 영향을 준 것으로 해석할 수 있다.

본 연구의 시사점은 다음과 같다.

첫째, 프로그래밍 중심 STEAM 교육 프로그램이 컴퓨팅 사고력 향상에 효

과적이다. 과학 주제를 학습하는 과정에서 프로그래밍을 활용하여 문제를 해결하는 경험을 통해 컴퓨팅 사고력과 융합적 소양을 기를 수 있다. 또한 이는 교육현장에서 프로그래밍과 컴퓨터과학 교육을 자연스럽게 도입할 수 있는 방법적인 측면에서도 기여할 수 있을 것이다.

둘째, 스크래치나 코듀 등 교육용 프로그래밍 언어를 활용한 프로그래밍 활동이 STEAM 교육 프로그램의 주요 내용 및 도구로 사용할 수 있음을 제시하였다. 프로그래밍은 컴퓨팅 사고력을 바탕으로 이루어지며 이런 컴퓨팅 사고력은 컴퓨터과학만이 아니라 과학, 수학 등의 다른 교과에도 효과적으로 사용될 수 있다. 또한 프로그래밍 활동이 갖는 창의적이고 교과 융합적인 특징이 예술(Arts) 분야와 자연스럽게 연계될 수 있는 사례를 보여줌으로써 프로그래밍의 교육적 활용 범위를 확장시키는데 기여할 수 있다.

셋째, 프로그래밍 중심 STEAM 교육을 위한 교수 전략으로 교육용 프로그래밍 언어를 통한 디지털 스토리텔링, 피지컬 컴퓨팅, 로보틱스 등을 제시하였다. 각 전략을 활용한 STEAM 교육 프로그램이 충분한 교육적 효과를 가졌음이 검증되었으므로 앞으로 관련 분야의 후속 연구에도 참고가 될 수 있을 것이다.

향후 후속 연구를 통해 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램을 지속적으로 개선하여 다양한 학습자들에게 미치는 효과를 좀 더 체계적으로 분석한다면 컴퓨팅 사고력 신장을 위한 프로그래밍 교육과 융합적 소양을 위한 STEAM 교육의 학습효과를 극대화 할 수 있을 것이다.

주요어: 컴퓨팅 사고력, STEAM 교육, 프로그래밍 교육, 교육용 프로그래밍 언어(EPL), 스크래치, 코듀

I. 서론

1. 연구의 필요성

첨단 과학기술의 발전은 현대 사회를 빠르게 변화시키고 있다. 국가의 과학 기술의 수준은 국가의 경쟁력인 동시에 미래 사회를 위한 원동력이다. 전 세계에서 과학기술의 발전을 위한 경쟁이 계속되고 있으며 이를 위한 과학기술 교육에 대한 관심이 높아지고 있다. 우리나라 역시 과학기술의 발전을 위하여 미래사회를 이끌어 갈 인재를 육성하기 위해 노력하고 있다. 하지만 21세기는 환경오염, 식량부족, 에너지 고갈 등 융합적 지식과 사고를 기반으로 해야 하는 문제가 증가하고, 이러한 문제들은 기존의 지식 기반 접근방법으로 해결하기에는 한계가 있어 이를 해결하기 위해 지식을 넘나드는 융합적이고 창의적인 사고력과 실천력을 필요로 한다(백운수 외, 2011; 백운수 외, 2012). 이에 기존의 과학기술 교육 방법에는 한계가 있고 이를 극복하기 위한 연구가 계속되고 있다.

교육과학기술부는 2011년 추진 업무보고에서 미래사회를 이끌어갈 창의인재를 육성하기 위한 정책으로 ‘과학기술-예술융합’ 강화를 제시하고 STEM 교육에 예술을 포함한 STEAM 교육을 주요 정책으로 발표하였다. 우리나라 국가 경쟁력의 자시인 미래 과학기술 발전을 주도할 창조적이고 융합적인 인재 양성을 위해 초중등교육부터 과학기술에 대한 흥미와 이해를 높이고 융합적 사고와 문제해결능력을 배양할 수 있는 STEAM 교육이 필요하다는 것이다(교육과학기술부, 2010).

과학 분야를 살펴보면, 생명 공학에서는 컴퓨팅을 활용한 유전자의 분석을 통해 기존에 해결하지 못했던 문제들의 해답을 찾고 있으며 화학에서는 나노 기술에 의해 나노단위 분자를 제어하고 합성하여 현재까지의 연구 방법으로는 불가능하던 일들을 해내고 있으며 이러한 연구 활동들은 미시적 세계를 시뮬

레이션 할 수 있는 컴퓨팅 기술을 통해 가능해졌다. 이와 같은 과학 분야의 변화에는 논리적인 사고를 실제로 구현할 수 있도록 도와주는 컴퓨팅 능력이 중요한 역할을 담당하고 있다(이영준 외, 2014). 비단 과학 뿐 만이 아니라 다양한 분야에서 컴퓨팅 기술은 다양한 학문 간의 경계를 허물고 융합의 동력을 제공하는 중요한 역할을 차지하고 있다.

융합의 시대를 살아갈 학습자들이 융합 사회를 이끌어가는 핵심 인재로 성장하기 위해서는 융합의 핵심을 관통하는 컴퓨팅의 원리와 기술을 학습하여 창의적이고 효율적으로 문제를 해결하는 데 필요한 능력을 기르도록 해야 한다. 이를 위해서 단순한 컴퓨터 및 첨단 정보통신기기의 사용법이나 소프트웨어의 사용법을 가르칠 것이 아니라 미래 사회의 핵심적인 사고과정이라고 할 수 있는 컴퓨팅 사고력(Computational Thinking)을 함양시키는데 초점을 맞춰야 한다(이영준 외, 2014).

컴퓨팅 사고력은 Wing에 의해 처음 제시된 이후, 지난 몇 년간 교육의 여러 분야에서 광범위 하게 논의되고 있으며, 컴퓨터과학 분야의 많은 연구자와 교육자들의 관심분야였던 컴퓨팅 사고가 최근 소프트웨어 중심의 사회로 변모하는 과정에서 융합된 미래사회의 비전을 제시하고자 하는 많은 연구에서 주목하고 있는 이슈가 되었다(김병수, 2013; 이은경, 2013). 컴퓨팅 사고력이 이렇게 많은 관심을 받는 이유는 하드웨어 중심이었던 사회가 점차 소프트웨어 중심의 사회로 변화하고 있기 때문이다. 소프트웨어는 21세기 국가경쟁력을 견인하는 핵심으로 지식창출의 도구일 뿐만 아니라 제품의 고도화와 서비스의 혁신을 촉진시킬 수 있는 요소로 인정받고 있다. 미래창조과학부(2013)는 소프트웨어 혁신전략에서 우리 경제를 이끌어 갈 미래 산업으로서 소프트웨어를 지목하였으며 창조경제 성장 동력으로 국가경쟁력을 혁신하려는 전략을 발표하였다. 소프트웨어 산업 발전을 위한 첫 번째 과제로 제시한 것이 바로 소프트웨어 인력양성 및 교육 강화이다. 초·중·고등학생부터 누구나 프로그래밍을 배울 수 있도록 재미있게 배우는 교육프로그램을 만들고 교과반영을 확대 하겠다는 것이다.

프로그래밍은 자신에게 주어진 문제를 해결하고 창의적인 아이디어를 생성해내는 도구이다. 오로지 프로그래밍 자체를 위한 교육, 소프트웨어 개발자를

육성하기 위한 교육이 아니라 학생들이 실생활에서 만나는 문제를 컴퓨팅 사고력을 활용해 해결하도록 이끌어낼 수 있는 교육이 필요하다. 이를 위해서는 창의적 사고력과 문제해결능력을 갖춘 인재를 양성하는 기반을 마련하여 어릴 적부터 프로그래밍 교육을 체계적으로 배울 수 있는 환경이 조성되어야 한다. 하지만 현재 초, 중등교육에서 정규교육과정 속 컴퓨터과학 교육은 그 시수가 매우 적을 뿐 아니라 그것마저도 소프트웨어의 단순 활용에만 치중되어 있다. 프로그래밍 교육을 통해 창의적 사고력과 문제해결능력을 신장시키고 궁극적으로 21세기 학습자에게 필요한 컴퓨팅 사고력을 함양하기 위해 어떻게 프로그래밍을 학습해야 할지 교육 방법에 대한 논의가 필요하다.

요약하면 미래사회를 이끌어갈 학습자들에게 필요한 핵심 역량으로서 컴퓨팅 사고력과 융합적 소양을 기르기 위한 교육이 필요하며 이에 본 연구에서는 프로그래밍 활동을 통해 학습자의 컴퓨팅 사고력과 융합적 소양을 기를 수 있도록 프로그래밍 중심 STEAM 교육 프로그램을 개발하였다.

2. 연구의 내용



본 연구에서의 연구 내용은 다음과 같다.

첫째, 컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램을 개발하기 위해 기존 컴퓨팅 사고력, STEAM, 프로그래밍 교육에 대한 연구에 대해 조사하고 그 특징을 비교, 분석하여 제시한다.

둘째, 프로그래밍 중심의 STEAM 교육 프로그램에서 프로그래밍과 STEAM의 교과 학습을 적절하게 융합하기 위한 아이디어로 디지털 스토리텔링, 피지컬 컴퓨팅, 과학적 글쓰기를 제안하고 그 방법을 제시한다.

셋째, 컴퓨팅 사고력을 증진시키기 위한 프로그래밍 중심의 STEAM 교육 프로그램을 설계, 개발한다.

넷째, 개발한 교육 프로그램을 학습자에게 투입하고 창의력, 논리적 사고력, 과학에 대한 흥미 등의 검사도구를 이용하여 본 연구에서 개발한 교육 프로그

램의 연구 결과를 도출·분석하여 효과를 검증한다.

ISD 모형에 따른 연구 방법 및 연구 산출물을 좀 더 구체적으로 살펴보면 <표 I-1>과 같다.

<표 I-1> ISD 모형 단계에 따른 연구 방법 및 산출물

단계	연구 내용	연구 방법	산출물
분석	컴퓨팅 사고력 내용 연구	문헌 연구	컴퓨팅 사고력 요소 분석 및 조작적 정의
	선행 연구 분석	사례 연구	컴퓨팅 사고력 내용 분석
	학습자 요구 분석	조사 연구	학습자 요구 추출
설계	통합 유형 선정	문헌 연구	STEAM 교육프로그램 개발을 위한 통합 모형
	교육과정 분석	문헌 연구 개발 연구	교육과정 분석표
	학습 준거 선정	개발 연구	프로그램의 STEAM 교과요소 선정
	수업모형 선정	문헌 연구	교육 프로그램에 적합한 수업 모형 선정
	평가도구 선정	문헌 연구	교육적 효과를 검증하기 위한 검사지 선정
개발	프로그래밍 중심 STEAM 교육 프로그램 개발	개발 연구	프로그래밍 중심 STEAM 교육 프로그램
적용 및 평가	적용 및 학습 효과 분석	실험 연구	프로그램 적용 결과 분석표

3. 연구의 기대 효과

본 연구의 기대 효과는 다음과 같다.

첫째, 프로그래밍 활동을 통한 과학교과의 학습 주제의 학습이 STEAM 교육에서 추구하는 컴퓨팅 사고력의 증진과 과학에 대한 정의적 요소의 긍정적인 변화에 영향을 줄 것이다.

둘째, 현재 정식교과가 없는 컴퓨터과학 및 프로그래밍을 초등교육 현장에 도입하기 위한 방법으로서 프로그래밍 중심의 STEAM 교육 프로그램이 대안

이 될 수 있으며 이는 앞으로의 교육현장 속 프로그래밍 교육에 도움이 될 수 있을 것이다.

셋째, 교육용 프로그래밍 언어가 가지는 교과 융합적인 요소와 프로그래밍 활동이 가지는 교육적인 요소들에 대한 재평가의 기회가 될 것이다.

넷째, 프로그래밍 중심의 STEAM 교육 프로그램의 교과 요소나 주제가 고정되어진 것이 아니기 때문에 과학, 수학 등의 교과 주제를 학습하는 다양한 교육 프로그램 개발에 대한 모델을 제시할 수 있을 것이다.

다섯째, 프로그래밍 활동을 STEAM 교육에 접목하기 위한 방법으로 디지털 스토리텔링, 피지컬 컴퓨팅, 과학 글쓰기 등을 제시하였는데 이를 활용하는 이후의 관련 연구에 도움이 될 수 있을 것이다.

4. 연구의 제한점



본 연구의 제한점은 다음과 같다

첫째, 본 연구에서 개발한 교육 프로그램의 평가를 위해 실험대상으로 4개의 집단을 선정하여 효과를 분석하였다. 하지만 각 집단의 개체 수 부족으로 정규성 확보에 어려움이 있었고 비모수 검정을 통해 이 문제를 최소화하려고 했으나 전체적인 실험집단의 부족으로 인하여 모든 학생에게 일반화 하는 데에는 어려움이 있다.

둘째, 본 연구의 목적 중 하나인 프로그래밍 중심의 STEAM 교육 프로그램을 통해 학습자의 컴퓨팅 사고를 신장시키는 것이다. 컴퓨팅 사고력을 평가할 수 있는 공신력 있는 검사 도구가 존재하지 않는 까닭에 컴퓨팅 사고력의 구성 요인으로 여겨지는 창의력, 논리적 사고력의 검사지를 이용하여 간접적으로 컴퓨팅 사고력의 신장을 점검하였다. 따라서 본 연구의 프로그램이 학습자의 컴퓨팅 사고력을 신장시켰는지에 대한 직접적인 검증이 불가하였다.

셋째, 본 연구에서 개발한 STEAM 교육 프로그램은 과학 교과의 학습내용을 중심 주제로 개발 하였다. 교육과정을 분석한 내용을 토대로 프로그래밍 학

습에 적합한 주제를 연구자가 임의로 선정하였으며 본 연구의 결과를 초등학교 전체 과학 주제를 학습하는 효과로 일반화 하는 데에는 어려움이 있을 수 있다.

마지막으로 본 연구에서는 STEAM 교육의 목적 중 하나인 융합적 소양에 검증에 제한이 있었다. 그 이유로는 STEAM 교육을 통한 융합적 소양을 검증할 수 있는 공신력 있는 검사지가 없고 초등학생 수준에 높은 수준의 융합적 소양을 요구하는 데에도 힘들다는 판단을 내렸기 때문이다. 따라서 본 연구에서는 융합적 소양을 STEAM 교육의 목표 중 하나인 학습자가 과학에 대한 흥미를 갖게 하는 것으로 설정하여 과학과 관련된 정의적 영역을 평가하였다.



II. 이론적 배경

1. 컴퓨팅 사고력(Computational Thinking)

컴퓨팅 사고력은 인간이 실생활에서 직면할 수 있는 다양하고 복잡한 문제를 어떻게 해결할 것인지를 절차적으로 사고하고, 문제의 해결 과정을 컴퓨팅 기기가 제공하는 강력한 능력들을 통해 효과적이고 효율적으로 해결하고자 하는 종합적인 사고과정이라 할 수 있다.

컴퓨팅 사고력은 컴퓨팅 기술이 사회의 근간이 되고 이를 활용하여 복잡한 문제를 해결하는 능력이 필요한 21세기에 학습자가 갖추어야 할 핵심 역량으로 강조되고 있으며 읽기, 쓰기, 셈하기처럼 모든 사람이 반드시 알아야 할 기초 소양으로 인식되고 있다(Wing, 2006).

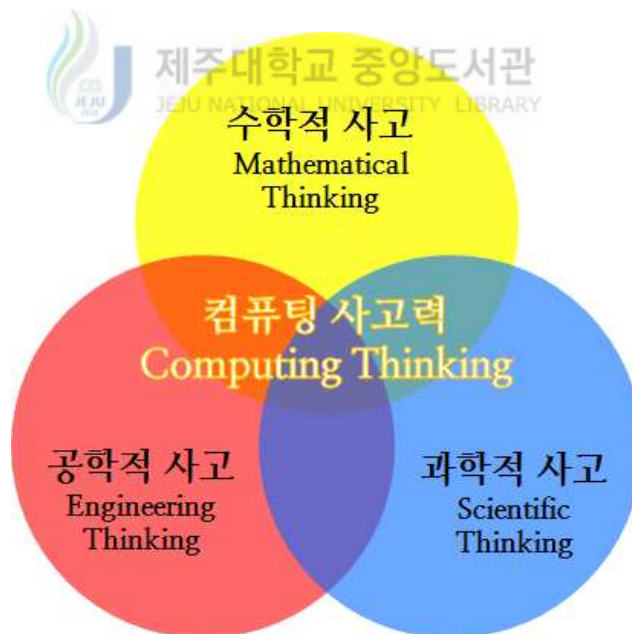
이렇게 미래 사회에서 중요한 역할을 차지할 컴퓨팅 사고력을 학습자가 체득하고 활용하기 위해서 다양한 교육 장면에서 컴퓨팅을 활용한 문제 해결 과정의 경험을 폭넓게 제공하여 자연스럽게 컴퓨팅 사고력을 익히도록 도와주어야 한다.

1) 컴퓨팅 사고력의 정의

컴퓨팅 사고력은 Wing에 의해 처음 그 개념이 정의되어 소개되었다. Wing(2006)은 컴퓨팅 사고력을 ‘해결해야 할 문제를 만났을 때 컴퓨터 과학자처럼 사고하는 것’이라 정의하면서 ‘컴퓨터 과학의 기초적인 개념들에 기반을 둔 문제해결, 시스템 설계, 인간 행동의 이해를 포함하는 개념’이라고 주장하였다. 인간이 실제적인 문제를 접했을 때 자신이 컴퓨터보다 더 잘할 수 있는 능력과 컴퓨터가 자신보다 더 잘 수행할 수 있는 능력을 안다면 이를 활용해 문제를 효과적으로 해결할 수 있으며, 컴퓨팅 기기의 장점을 알고 컴퓨터 과학자처럼 문제를 해결하는 과정을 사고하는 것은 21세기를 살아갈 모든 사람이 갖

추어야 할 기본적인 태도 및 기술이라고 주장하였다.

Wing(2008)은 미래의 모든 과학 및 공학에 관련된 분야는 컴퓨팅의 영향력 아래에 놓여있다고 주장하면서 컴퓨팅 사고력이 다른 학문 분야들에 미치는 강력한 영향력 아래에 놓여 있다고 주장하면서 컴퓨팅 사고력이 다른 학문 분야들에 미치는 강력한 영향력에 대한 근거를 제시하였다. “컴퓨팅 사고력은 하나의 분석적 사고로서 문제를 해결하는 일반적인 방법에서는 수학적 사고를 공유하고, 실세계의 제약 안에서 작동되는 크고 복잡한 시스템을 설계하고 평가하는 접근에는 공학적 사고를 공유하며, 인간의 행동, 마음, 지능, 계산가능성에 대한 이해를 위한 접근에서는 과학적 사고를 공유한다.”고 설명하였다. 이러한 Wing의 관점은 컴퓨팅이 자연과학(Natural science)이며 컴퓨팅의 범위를 단순히 수학, 공학적 문제의 범위를 벗어나 자연에서부터 인간의 일상생활, 물리적 생활공간과 사회과학까지를 포함하여야 한다는 Denning(2007, 2010)의 주장과도 일치한다(김병수, 2014).



[그림 11-1] 컴퓨팅 사고와 공유되는 사고들

Wing(2006)은 컴퓨팅 사고력이 3R(읽기, 쓰기, 셈하기)과 마찬가지로 모든 학습자가 갖추어야 할 기본 능력이며, 추상화와 자동화를 통한 문제해결능력이

라고 하였다. 컴퓨팅 사고력이 문제 해결 과정에서 추상화 과정을 통해 문제의 핵심 요소를 추출하고 모델링하여 컴퓨팅 기기를 통해 해법을 자동화하는 능력을 의미한다고 하였다.

Wing의 주장은 컴퓨팅 사고력에 대한 전반적인 논의의 틀을 제공하고 있고, 기본적인 컴퓨팅 사고력의 개념에 대하여 규정하고 있어 그 의미를 인정받고 있다(이영준 외, 2014). 컴퓨터 과학 분야에서는 컴퓨팅을 활용한 문제 해결 과정이 보편적인 문제해결능력과 사고력에 긍정적인 영향을 준다는 점에 착안하여 컴퓨팅 사고력에 대한 논의를 진행해 왔으며 다양한 연구자에 의해 컴퓨팅 사고력에 대한 정의가 내려지고 있다. 계산적 사고에 대한 국내·외의 관련 연구를 <표 II-1>에 제시하였다.

<표 II-1> 컴퓨팅 사고 관련 국내·외 관련 연구

연구자 (발행연도)	연구 내용	
Wing (2006, 2008)	제목	Computational thinking
	정의	Computational thinking is taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing. (컴퓨팅 사고는 컴퓨팅의 기본 개념과 원리에 기반하여 문제를 해결하고, 시스템을 설계하고, 인간의 행동양식을 이해하고자 하는 접근 방법이다.)
CSTA Standards Task Force (2011)	제목	CSTA K-12 Computer Science Standards (revised 2011)
	정의	Computational thinking is an approach to solving problems in a way that can be implemented with a computer. (컴퓨팅 사고는 컴퓨터를 활용해 구현할 수 있는 방법으로 문제를 해결하는 접근 방법이다.)
National Research Council (2010)	제목	Report of a Workshop on the Scope and Nature of Computational Thinking
	정의	<ul style="list-style-type: none"> • Peter Lee - 컴퓨팅 사고력은 인간 지능을 확대하여 실제적인 적용을 할 수 있는 인간 지능의 매커니즘에 관한 연구이다. 다시 말해 인간의 정신적인 능력의 복잡도를 관리하거나 일을 자동적으로 처리하도록 하는 추상화 도구를 통해 확장하는 것이라 할 수 있다. • Bill Wulf - 과학은 물리적인 대상에 관련된 것이고, 컴퓨팅 사고력은 어떠한 문제를 해결하는 과정과 그 과정의 진행을 가능하게 하는 추상적인 현상들에 초점을 맞춘다. • Gerald Sussman - 컴퓨팅 사고력은 일을 처리하는 정확한 방법을 공식화하는 방법이다. 다시 말해 특정한 문제를 효율적으로 처리하기

		<p>위해 그 문제를 철저하게 분석하고 해결하기 위한 엄밀한 절차를 만드는 과정이 컴퓨팅 사고력이다.</p> <ul style="list-style-type: none"> • Edward Fox - 인간이 세상에 접근하고, 과정들을 생각하고, 디지털로 표현된 것들을 다룰 때 하는 일들로 표현하였다.
National Research Council (2011)	제목	Report of a Workshop on the Pedagogical Aspects of Computational Thinking
	정의	<p>컴퓨팅 사고에 대한 교육학적 관점들에 대한 논의가 이루어졌다.</p> <ul style="list-style-type: none"> • Peter Henderson - Computational thinking as generalized problem solving with constraints. (제약을 가진 일반화된 문제 해결로서의 컴퓨팅 사고)
이은경 (2009)	제목	Computational Thinking 능력 향상을 위한 로봇 프로그래밍 교수 학습 모형
	정의	<p>21세기를 살아가는 모든 사람이 갖추어야 할 기본적인 사고 능력으로 컴퓨터 과학의 기본 개념과 원리에 따른 문제 해결, 시스템 설계, 인간 행동의 이해를 포함하는 추상적 사고 능력이다. 이는 문제 해결을 위해 적합한 추상적 개념을 선택하고 구성하기 위한 추상화 능력과 추상적 개념을 자동화하기 위해 적합한 컴퓨팅 도구를 선택하고 사용하기 위한 능력을 포함한다.</p>
김형철 (2011)	제목	컴퓨터과학 교육용 계산 원리 학습도구의 기능요소 고찰
	정의	<p>컴퓨팅 사고는, 좁은 의미로는 계산 시스템(computational system)을 활용해 효과적으로 작업하기 위해 습득해야 할 사고방식이나 태도이며, 넓은 의미로는 세상을 이해하는 양식(단순한 방법을 초월한 양식, 광범위한 인간노력에 두루 접목 가능한 양식)이다.</p>
권대용 (2011)	제목	Algorithmic brick based tangible robot and hybrid programming environments for enhancing computational thinking
	정의	<p>컴퓨팅 사고는 21세기 중반까지 모든 사람들이 사용하게 될 필수적 기술(skill)로써, 컴퓨터 과학의 기본 개념과 원리에 따른 문제 해결, 시스템 설계, 인간 행동의 이해를 포함하며 컴퓨터 과학의 다양한 영역을 반영하는 정신적 도구의 범위를 포함한다.</p>
김병수 (2014)	제목	계산적 사고력 신장을 위한 PPS 기반 프로그래밍 교육 프로그램
	정의	<p>실세계에서부터 디지털세계를 망라하여 자연적으로 또는 인간 사회에 의해 존재하는 다양한 현상 속의 계산을 계산 대행자(computational agent)를 이용하여 발견하거나 새롭게 창조하기 위해 습득해야 할 인간의 사고 양식과 태도이다.</p>
이영준 외 (2014)	제목	초중등 단계 Computational Thinking 도입을 위한 기초 연구
	정의	<p>컴퓨팅 사고력은 컴퓨팅 시스템의 역량을 활용하여 해결하고자 하는 문제를 효과적이고 효율적으로 해결할 수 있는 절차적 사고 능력이다.</p>

2) 컴퓨팅 사고력의 구성 요소

컴퓨팅 사고력은 문제해결을 위한 기본 패러다임으로서 추상화(Abstraction)와 자동화(Automation)를 주된 도구로서 사용한다.

(1) 추상화

컴퓨터과학에서 추상화에 대한 중요성에 대한 언급은 계속되어 왔다 (Devlin, 2003; Hazzan, 1999; Kramer, 2007). 추상화라는 의미는 크게 2가지 측면의 의미를 갖는다. 추상화는 '무엇을 단순화하고 관심을 집중시키기 위해 세부적인 것을 제거하는 과정'이며, '그것들의 공통적인 핵심과 본질을 찾기 위해 이를 일반화하는 과정'이다. 추상화는 실제 세계의 문제를 해결 가능한 형태로 표현하기 위한 사고과정이라고 할 수 있으며, 문제를 해결하기 위하여 필요한 자료를 수집 및 분석하고, 필요한 표현 방법(도표, 그래프 등)을 활용하여 눈으로 보기 쉽게 나타내고, 복잡한 요소를 작은 단위로 분해하고, 해결에 필요한 변수들을 추출하여 적절한 해결 모델을 설계하는 과정이다.

추상화는 컴퓨터과학 전반에 걸쳐 가장 핵심이라고 할 수 있다(Bennedssen & Caspersen, 2008; Koppelman & van Dijk, 2010; Kramer, 2007). 추상화의 기술은 인간이 이미 습득한 능력이며 많은 영역에서 활용되고 있다. 인간은 태어나서 성장과정에서 수 세기, 색 이름의 사용, 사람과의 관계에 대한 명칭 사용 등에서 보편적인 추상화에 대한 반복적인 연습을 하게 된다. 수학 분야에서 이러한 추상화에 대한 연구와 학습이 많이 이루어지기는 하지만 그 영역은 수학적 사고에서 크게 벗어나지 못한다.

Wing(2008)은 컴퓨팅 사고력의 핵심 요소인 추상화가 다른 학문 분야에서 다루는 추상화와는 차이점을 제시하였다.

첫째, 컴퓨팅 영역에서의 추상화는 시간, 공간, 물리적 차원을 넘어선 개념을 추상화하기 때문에 특정 사례로 표현되는 수학적 추상화에 비해 매우 일반적인 특성을 지닌다.

둘째, 컴퓨팅의 추상화는 수학이나 물리학의 추상화 보다 더 풍부하고 복잡한 경향이 있고 수학에서처럼 명확하고 간결하게 해결할 수 없다.

셋째, 컴퓨팅 영역에서의 추상화는 본질적으로 물리적 실세계의 제약조건 안에서 수행하기 위해 구현되기 때문에 극도로 특수한 사례들과 실패 사례들을

면밀히 고려해야 한다.

컴퓨터과학 분야에서 추상화의 범위는 하드웨어, 소프트웨어, 또는 추상적 개념뿐 아니라 문제 상황 자체, 그리고 그것의 해결과정 등의 영역에서 이루어지며 이 과정에서의 추상적 개념의 추상화, 그리고 이것들의 반복으로 인해 생기는 추상화 레벨에 대한 이해도 필요하게 된다.

(2) 자동화

Wing(2008)은 추상화를 정신적 도구(mental tool)라고 한다면 이러한 추상화의 기능은 기계적 도구(metal tool)인 자동화를 통해 더 증폭되어진다고 말한다. 추상화 과정을 통해 생성한 추상적 개념들은 다양한 컴퓨팅 장치를 활용한 자동화 과정을 통해 강화될 수 있으며, 자동화는 추상적 개념들을 해석하고 구현하기 위한 장치들을 필요로 한다. 자동화란 추상개념과 추상 레이어, 이들 사이의 관계를 기계화하여 작동시킨다. 즉 이해하기 복잡하고 까다로운 추상개념과 추상레이어를 정교하게 해석할 수 있도록 기계적으로 자동화할 수 있다는 것은 컴퓨터 발전 역사의 핵심이라고 할 수 있다. 컴퓨팅 머신의 빠른 처리 능력, 저장 공간, 통신 능력이 있었기에 이러한 발전이 가능했다(김병수, 2014).

자동화에서 가장 일반적으로 활용가능한 장치는 컴퓨터이지만, 추상적 개념의 속성에 따라 컴퓨터보다 인간이 수행하는 것이 더 치밀하고 견고할 수 있다. 따라서 추상적 개념의 자동화 과정은 특정 기계 장치를 필요로 하지 않을 수 있으며, 인간의 처리 능력과 기계의 처리 능력이 통합된 형태를 이용하거나 다양한 기계 장치의 통합 형태를 이용할 수 있다. 따라서 문제 해결을 위해 생성한 추상적 개념의 자동화를 위해서는 상황에 따라 가장 적절한 도구를 선택하는 능력이 가장 중요하다(이은경, 2009).

3) 컴퓨팅 사고력 검사 도구

컴퓨팅 사고력이라는 용어가 사용된 이래 컴퓨팅 사고력이 무엇인지에 대한 논의가 계속되었지만 이것을 어떻게 측정하고 평가할 수 있을 것인가에 대한 연구는 그에 비해 미흡했고 최근에 들어서야 몇몇 연구들이 시작되었다.

컴퓨팅 사고력의 평가 도구 개발에 대한 국내·외의 연구는 <표 II-2>와 같다.

<표 II-2> 컴퓨팅 사고력 측정 도구 개발 관련 연구

연구자 (발행연도)	연구 내용	
김중혜 (2009)	제목	정보과학적 사고 기반의 문제 해결 능력 향상을 위한 중등 교육 프로그램
	정의	정보과학영재 교육원과 PISA의 ‘문제 해결’ 영역에 해당하는 문항들 중 컴퓨팅 사고력 기반의 문제 해결 능력을 평가하기에 적합한 예비 문항을 선별하여 문항의 타당도와 양호도 검증을 통해 중학생 난이도의 10문항과 고등학생 난이도의 15문항을 개발하였다.
이은경 (2009)	제목	Computational Thinking 능력 향상을 위한 로봇 프로그래밍 교수 학습 모형
	정의	컴퓨팅 사고력을 측정하기 위한 문항으로 OECD/PISA의 문제 해결 영역 평가 틀을 이용하여 문제 유형, 문제 상황, 학문 영역, 문제해결 과정, 컴퓨팅 사고력을 구성요소로 검사도구를 개발하였다. 개발된 검사도구를 대학생들을 대상으로 투입하여 로봇프로그래밍 교수 학습 모형이 컴퓨팅 사고력에 미치는 영향을 분석하였다.
김병수 (2014)	제목	계산적 사고력 신장을 위한 PPS 기반 프로그래밍 교육 프로그램
	정의	컴퓨팅 사고력의 구성요소를 계산적 인지력(추상적 사고, 비판적 사고, 논리적 사고, 재귀적 사고, 알고리즘적 사고)와 계산적 창의성(창의적 사고)로 구분하여 설정하였으며 컴퓨터과학 분야의 암기적 지식에서 벗어나서 실제적인 문제 해결 과정에 적용할 수 있는 각각 19문항의 동형의 계산적 인지력 검사도구를 개발하였다. 개발된 검사도구를 GALT 검사와의 상관 분석을 통해 타당성을 입증하였다.

이은경(2009)의 연구에서 개발된 검사 도구는 PISA의 문제를 바탕으로 만들어진 문항들로 컴퓨팅 사고력의 하위요인으로 논리적/분석적/추상적/동시적/선행적/전략적/절차적/재귀적 사고로 선정하고 문제 유형, 문제 상황, 학문 영역, 문제해결과정, 컴퓨팅 사고력을 문항 개발의 틀로 사용하여 검사 도구를 개발하였다. 다만 문제의 발견 영역에 논리적/분석적 사고가 해당하고 문제 분석 및 표현에는 분석적/추상적 사고가 해당하며 문제 해결 전략에는 동시적/선행적/전략적/절차적/재귀적 사고가 해당한다고 제시하였지만 영역과 하위요인 사이에 어떠한 관련이 있는지 명확한 근거를 제시하지 못했다.

김중혜(2009)의 연구에서 개발된 검사 도구는 컴퓨팅 사고력 기반의 문제 해결 능력을 분석하여 정보과학영재 교육원과 PISA의 ‘문제 해결’ 영역에 해당하는 문항을 토대로 ‘도서관, 수에 의한 디자인, 교육과정, 환승체계, 에너지, 영화감상, 관개, 전화통화, 냉동고, 휴가’ 문항을 선별하여 검사 도구를 개발하였다. 하지만 선정된 평가 문항의 주제는 컴퓨터 과학의 핵심 주제에 어떠한 관

계를 갖고 있는지에 대한 명확한 근거를 제시하지 못하였다.

김병수(2014)는 컴퓨팅 사고력을 인지적인 측면과 창의적인 측면의 구성요소를 가진다고 의 연구에서 개발된 검사 도구는 프로그래밍을 중심으로 한 컴퓨터과학의 핵심 주제와 아이디어에 대한 관련 연구와 소프트웨어의 인지복잡도의 측정에서 주제를 도출하였다. 도출된 10가지의 주제는 순차구조, 조건분기, 반복, 동시성(병렬처리), 변수, 난수, 알고리즘, 객체, 함수, 재귀이며 이러한 주제를 실생활 문제와 연계하여 특정한 지식이 없이도 그 개념이나 아이디어에 대한 문제를 해결할 수 있도록 문항을 제작하였고 특히, 어린 학생들을 대상으로도 투입할 수 있도록 문항의 설명을 줄이고 직관적으로 문제를 해결할 수 있도록 그림을 많이 이용하였으며 예시 설명을 추가하여 제작하기도 하였다. 개발된 검사도구가 컴퓨팅 사고력의 인지적인 측면만 다루고 있다는 부분에서 전체적인 컴퓨터 사고력을 판단하는데 제한점이 있다.

위에서 살펴본 것처럼 컴퓨팅 사고력을 측정하는 검사도구에 대한 연구가 진행되고 있지만 아직 그 한계점이 뚜렷하여 학습자의 컴퓨팅 사고력을 측정을 하는 데에는 제한이 있다. 다만 선행 연구에서 컴퓨팅 사고력을 구성하는 요소를 인지적인 측면과 창의적인 측면으로 구성되어 있음을 알 수 있었다. 이에 본 연구에서는 학습자의 컴퓨팅 사고력을 측정하기 위한 도구로 논리적 사고력과 창의력 검사를 선정하였다. 논리적 사고력과 창의력은 많은 연구에서 컴퓨팅 사고력의 하위요소로 다루어지고 있으며 많은 연구에서 프로그래밍 교육의 효과를 검증하는데 사용되고 있으며 이미 많은 데이터를 보유하고 있어 공신력을 확보한 검사지가 많다는 장점이 있기 때문이다.

2. STEAM 교육

21세기는 지구온난화, 자원부족 등과 같은 융합적 지식과 사고를 기반으로 해결해야 하는 문제가 증가하고, 지식기반 사회에서 개념기반 사회로 전환되며, 창조와 문화가 중시되는 시대적 특징을 갖는다. 따라서 학문 분야의 영역

을 넘나드는 융합적이고 창의적인 사고를 하는 통섭적인 인재가 필요하다(백윤수 외, 2012). 창의적인 과학기술인재를 육성하고자 추진되는 STEAM 교육은 미래 과학기술 사회가 요구하는 과학, 기술, 공학, 예술, 수학 등 다양한 분야의 융합적 지식을 기반으로 새로운 가치를 창출하고, 종합적인 문제 해결력을 갖춘 인재를 양성하는 교육이다(백윤수 외, 2011).

1) 등장 배경

STEAM 교육은 시작은 미국 교육개혁의 핵심인 STEM 교육에 뿌리를 두고 있다. STEM 교육은 미국과학재단(National Science Foundation, NSF)에서 과학(Science), 기술(Technology), 공학(Engineering), 수학(Mathematics)의 약칭에서 사용되었던 용어로 교육 정책이나 교육 관련 이슈에서 지속적으로 등장하면서 전 세계 과학 교육의 관심을 받게 되었다. STEM 교육은 21세기의 변화와 도전에 적합한 세계화 시대의 경쟁력을 확보하기 위해 STEM 관련 교과 의 융합과 실천을 강조한다. STEM 교육이 시작될 당시 미국은 STEM 교과에 대한 낮은 학업성취도와 흥미, 직업 관심도로 인하여 국제사회에서의 경쟁력을 걱정하는 상황이었다. 이에 미국 초·중학교 STEM 교육은 국가적인 과학기술 경쟁력 향상과 개인의 삶의 질을 높이기 위하여 STEM 교과의 학업 성취도 향상과 흥미 유발을 통해 우수한 학생들이 STEM 분야의 진로를 선택·유인하는 것을 목표로 설정하고 표준평가, 교사, 교육기술, 학생, 학교시스템 등 각 부문별 방향을 설정하여 현재까지 추진하고 있다(백윤수 외, 2011).

교육과학기술부는 2011년 추진 업무보고에서 창의적 과학기술인재 양성을 통한 인재강국 구현을 목표로 초·중등학교에서 과학기술에 대한 이해·흥미·잠재력을 높이는 STEAM 교육을 실시하는 계획을 발표하였다(교육과학기술부, 2011c; 교육과학기술부, 2010).

여기에는 우리나라 과학, 수학 교육이 갖고 있는 고질적인 문제점을 해결하기 위한 방안에 대한 고민이 담겨있다. 과학, 수학 교육을 통해 과학기술의 발전을 선도할 인재를 육성하고자 노력했지만 그 동안의 과학, 수학교육은 다음과 같은 문제점을 갖고 있었다.

첫째, 2011년 한국 어린이·청소년 행복지수 국제 비교 결과에 의하면 우리나라

라 어린이와 청소년의 주관적 행복지수가 OECD 국가 가운데 ‘최하위’로 나타나 이 부분에서 3년 연속 최하위를 기록하였다. 이는 우리나라 교육의 방향에 대한 전면적인 검토를 시사해 준다.

둘째, 국제 학업성취도 PISA와 TIMSS 조사 결과에 따르면, 우리나라 학생들의 수학과 과학 성취도는 상위권인데 비해 과학에 대한 일반적인 흥미, 수학·과학학습에 대한 자신감, 즐거움 등은 상대적으로 매우 낮게 나타났다.

셋째, 앞서 제시한 과학과 수학에 대한 초·중고생의 낮은 흥미와 과학기술분야에 대한 진로 기피 등은 우리나라 과학기술 국가 경쟁력의 악영향을 미칠 것으로 예상된다. 대학수학능력시험 중 자연계의 비율과 대학입학시험의 자연계 지원율이 감소 추세로 나타나는 등 실제적인 수치로도 확인 가능하다. 수험생들이 자연계열을 기피하는 이유는 대학 진학을 쉽게 하려는 이유도 있겠지만 청소년들의 과학·수학에 대한 흥미 및 자신감의 낮은데서 찾을 수 있다.

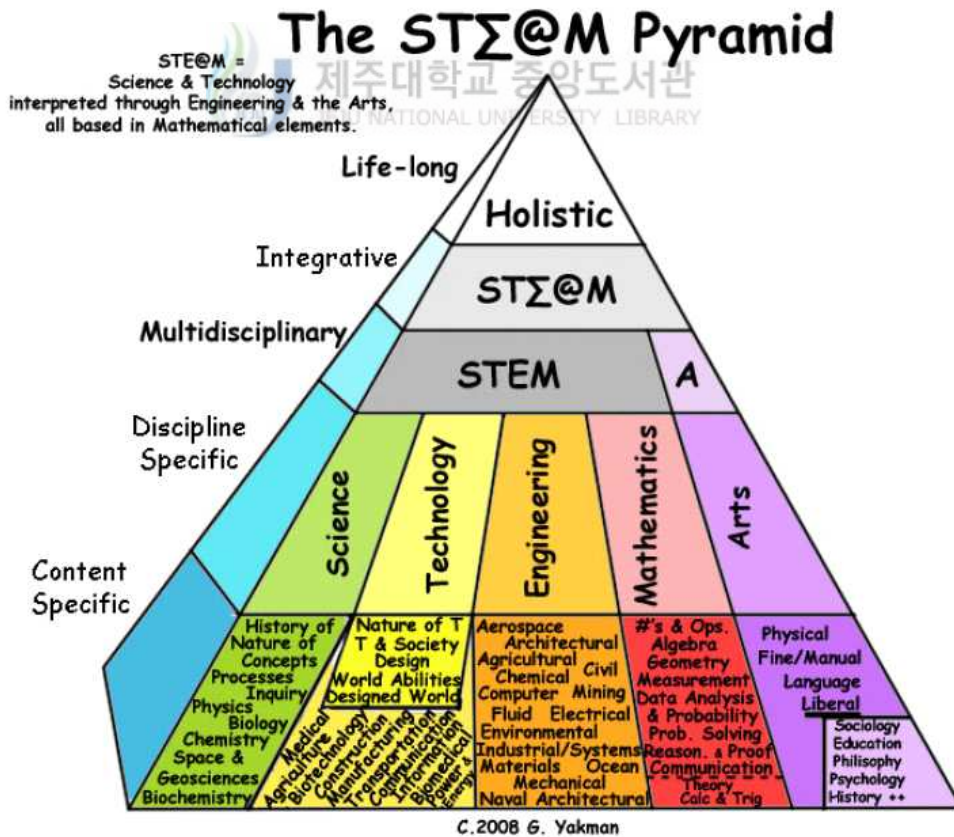
마지막으로 사회 및 시대 변화에 따른 주입식 교육방법이 한계에 도달하였다. 주입식 교육으로 이루어지는 그동안 우리나라 과학기술의 혁신이 모방에서 이루어졌다면 앞으로는 창조의 패러다임으로 전환되어야 한다. 이러한 전환이 이를 수 있는 인재를 기르기 위해서는 주입식 교육방법으로는 한계가 있으며 학생들의 흥미를 유발하고 창의력과 문제해결능력을 신장시켜주는 교육방법이 필요하다(백운수 외, 2011).

앞에서 언급한 과학기술교육의 문제점을 해결하기 위해 주목하게 된 것이 바로 STEAM 교육이다. 교육과학기술부는 국가 경쟁력의 자산인 미래 과학기술 발전을 주도할 창조적이고 융합적인 인재 양성을 위해 초·중·등 교육단계에서 STEAM 교육을 통해서 과학-기술-공학-수학의 학습내용을 핵심 역량 위주로 재구조화하여 과목 간 연계와 예술적 기법을 접목하는 정책을 제안하였다. 즉 STEAM 교육을 통하여 학생들의 흥미와 이해를 높임으로써 우리나라 과학기술교육이 가진 문제를 해결하고자 한 것이다.

미국에서 시작된 STEM의 경우 학생들의 STEM 교과 관련 학업성취도 향상 및 관련 교과에 대한 흥미 유발, 여학생과 소수민족의 이공계 진출 장려에 목적이 있었다면 우리나라의 STEAM 교육은 다른 국가에 비해서 높은 학업성

취도에도 불구하고 상대적으로 낮은 과학학습에 대한 효능감, 자신감, 흥미를 증진시키고 이공계 기피 현상을 해결하고자 시작되었다는데 차이점이 있다(백윤수 외, 2011; 손연아 외, 2012; Sanders, M., 2009). 따라서 우리나라의 STEAM 교육은 학생들의 STEAM 교과에 대한 학업성취도를 높이는 것도 중요하지만, 가장 큰 목표로는 실제적인 문제를 해결하는 과정에서 낮은 과학 효능감, 자신감, 과학에 대한 흥미를 증진시킴으로서 학습에 대한 동기유발을 목적으로 설정하고 있다.

STEM 교육이 진행되면서 다양한 연구들이 진행되었는데 Yakman은 기존의 STEM 학문 분야에 Arts 학문 분야를 통합한 STEAM 교육을 개발하여 발표하였다. 여기서 Arts는 체육, 순수미술, 응용미술, 언어 및 인문학을 모두 포함하는 것으로 기존의 STEM에 Arts가 더해지면서 창의와 융합에 더욱 초점이 맞춰지게 되었다(Yakman, G, 2010; Yakman, G, 2008).



[그림 11-2] STEAM 교육 프레임워크

Yakman은 [그림 II-2]에 제시한 것처럼 STEAM의 교과 요소들을 분석하고 STEAM의 각 교과를 융합해서 지도하는 다섯 단계로 구분하여 피라미드 형태의 STEAM 교육 프레임워크를 제시하였다(Yakman, 2008)

첫 번째는 평생 교육(life-long)의 단계로, 의도되지 않고 피할 수 없는 주변 환경에 적응하며 꾸준히 배우는 단계이다.

두 번째는 융합 교육(integrative)의 단계이다. 이 단계에서 학생은 모든 학문에 대한 광범위한 시각과 그 학문들이 실제 어떻게 연관이 있는지 기본적인 개관에 대하여 학습하게 된다. 이 단계에서 가장 좋은 학습의 방법은 주제 중심으로 학습하는 것이고 초·중등 교육에 적합하다.

세 번째는 학제 간 교육(multidisciplinary) 단계이다. 이 단계에는 학습자가 선택한 학문에 관한 시간과 실제와는 어떠한 연관이 있는지 학습을 하는 단계이다. 실제 기반의 내용을 학습하는 것이 가장 좋은 방법으로 중학교 교육에 적합하다.

네 번째 단계는 학문 분류 교육(discipline specific)단계이다. 특정 학문을 깊이 있게 학습하면서 관련된 학문의 분야까지 확대해 나가는 단계로 중등 교육에 적합하다.

다섯 번째 단계는 내용 분류 교육단계(content specific)이다. 이 단계는 각 세부 내용에 대하여 상세한 연구가 진행되는 단계로 전문적인 학습이 이루어지기 때문에 고등 교육에 적합하다(Yakman, G, 2008).

Yakman의 단계 구분을 종합하면 초등학생에 적합한 STEAM 교육의 형태는 STEAM 교과를 따로 학습하는 것보다는 교과를 아우르는 주제 중심, 실제적인 체험 중심의 활동을 통한 학습이 필요하다는 결론을 내릴 수 있다.

2) STEAM 교육의 요소

백윤수 외(2011)의 연구에서는 STEAM 교육을 다양한 분야의 융합적 내용을 창의적 설계와 감성적 체험으로 경험함으로써 과학기술과 관련된 다양한 분야의 융합적 지식, 과정, 본성에 대한 흥미와 이해를 높여 창의적이고 종합적으로 문제를 해결할 수 있는 융합적 소양을 갖춘 인재를 양성하는 것으로 정의하였다.

STEAM 교육의 요소에는 창의적 설계와 감성적 체험, 내용 통합이 있다.

창의적 설계는 학습자들이 주어진 상황에서 지식, 제품, 작품 등과 같은 산출물을 구성하기 위하여 창의성, 효율성, 경제성, 심미성을 발현하여 최적의 방안을 찾아 문제를 해결하는 종합적인 과정이다.

감성적 체험은 학습자들이 학습에 대해 긍정적 감정을 느끼고 성취의 기쁨과 실패의 가치를 경함하게 하는 다양한 활동들이 포함되는데 학습에 대한 흥미, 자신감, 지적 만족감, 성취감 등을 느껴 학습에 대한 동기 유발, 욕구, 열정, 몰입의 의지가 생기고 개인, 타인 및 공동체와의 관계, 자연과 문화 등의 의미를 발견하여 자기주도적 학습을 가능하게 하는 모든 활동과 경험을 의미한다.

내용 통합은 두 개 이상의 교과 내용이 유기적으로 통합하는 것을 의미한다.

교육과학기술부와 한국과학창의재단은 'STEAM 교육 실행방향 정립을 위한 기초연구'를 추진하였으며 이 연구의 결과로 STEAM 교육 학습 준거를 제시하였다. 학습 준거는 융합인재교육의 현장 적용을 위한 가이드라인이자 프로그램의 판단기준으로서 현장의 교사들이 쉽게 STEAM 교육 프로그램의 개발과 적용을 할 수 있는 도구이다. STEAM 교육 준거에서 강조하는 세 가지는 상황 제시, 창의적 설계, 감성적 체험이다(조향숙 외, 2012).

첫째, 상황 제시는 학생들이 주어진 상황의 실생활 문제를 자기 문제로 인식하도록 동기부여하기 위한 장치이다. 이는 교사 주도의 수업에서 학생 중심의 수업으로 전환하는 것을 의미하고, 학생들이 문제에 몰입하도록 해 준다. 학생이 문제를 자기문제로 인식하고 학습 주제에 관해 관련성(relevance)을 확보하기 위해서는 수업을 시작할 때 훨씬 정교한 시나리오 제시나 발문이 필요하다.

둘째, 창의적 설계(Creative Design)는 주어진 상황에서 문제를 해결하기 위하여 창의적으로 설계를 하는 과정을 의미한다. 실생활 문제에서 나타나는 여러 가지 제약 조건 속에서, 문제를 정의하고 최선의 해결책을 만들어 나가는 과정이 바로 창의적 설계 과정이다. 설계 과정에는 여러 학문의 지식이 필요하고, 모둠 활동의 경우 구성원들 간의 협동이 문제 해결에 중요한 열쇠를 제공할 수 있다. 창의적 설계 과정은 기존의 수업과 비교했을 때 학생 개개인의 생각이 구체적으로 표현되고 드러나도록 운영해야 한다.

창의적 설계는 학생들의 문제 정의 능력과 문제 해결 능력을 증진시키는데 그 목적이 있다. 과학적 이론은 대개 이상화된 모형들이다. 실생활의 문제나 직업 세계에서 수행하는 실제 과업은 이상화된 모형과 달리 여러 가지 제약 조건과 한계가 존재한다. 창의적인 설계 과정은 이러한 산업현장에서 일하는 방식으로 실생활 문제를 해결하는 능력을 배양하는 역할을 한다. 이러한 창의적 설계 과정은 과학보다는 공학적인 문제 해결 방식과 더 가깝다고 할 수 있다. 과학이 현상에 대한 원인, “왜?”에 대한 질문을 해결하는 것이라면 반면 공학은 방법에 대한 질문, “어떻게?”에 대한 해답을 구하기 때문이다.

창의적 설계는 문제를 정확히 인식하는 것에서 시작된다. 주어진 제약 조건과 한계 안에서 문제를 해결하는 과정은 과학기술 및 산업 현장에서 실제로 사용되는 방법이다. 대개의 문제에서는 여러 학문 분야의 지식들이 자연스럽게 융합된다. 창의적 설계는 주어진 상황에서 창의성, 효율성, 경제성, 심미성을 발현하여 최적의 방안을 찾는 종합적인 과정이라고 할 수 있다.

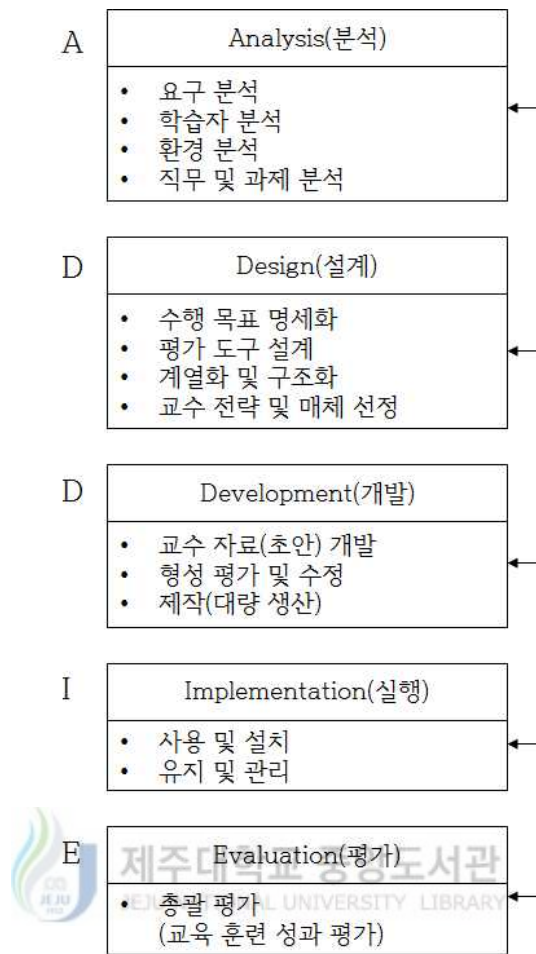
마지막으로 감성적 체험은 학생의 흥미와 동기부여를 위한 요소이다. 학습에 대한 성공의 경험을 통해 새로운 문제에 도전할 수 있는 용기를 자라게 하는 것이 감성적 체험의 목표이다. 성공의 경험을 통해 문제에 몰입하는 능력도 자라나게 된다. 학생들의 활동에 대한 피드백과 성과에 대하여 보상이나 격려를 통해 학생들의 감성적 체험을 강화할 수 있다.

이와 같은 STEAM 학습 준거를 통해 융합인재교육은 학생들이 자신과 연결된 문제를 해결하고, 더 나아가 또 다른 학습을 스스로 하고 싶도록 유도한다. 이러한 선순환적 구조가 완성되면 과학기술에 대한 관심과 흥미를 높이는 것과 더불어, 과학기술 분야로 진출을 유도하는 두 가지 목표를 동시에 달성할 수 있다(조향숙 외, 2012).

3) 프로그램 개발 모형

(1) ADDIE 모형

교육 프로그램 개발의 일반 모형인 ADDIE 모형의 주요 과정을 [그림 II-3]에 제시하였다(정재삼, 2004).



[그림 II-3] ADDIE 모형의 과정

ADDIE 모형에서 ‘분석(A)’은 학습 내용(what)을 정의하는 과정으로서 요구 분석, 학습자 분석, 환경 분석, 직무 및 과제 분석이 포함된다. ‘설계(D)’는 교수 방법(how)을 구체화하는 과정으로서 학습의 영역을 확인하면서 수행 목표의 중요성과 진술 방법에 대한 것, 수행 목표를 바탕으로 한 평가 도구의 설계, 수행 목표와 학습 내용의 조직화 및 계열화 원리, 교수 전략, 교수 매체를 선정하는 처방 등이 포함된다. ‘개발(D)’은 교육 프로그램을 만들어 내는 과정으로서 교육 프로그램의 초안을 개발하여 형성 평가를 실시한 결과를 바탕으로 프로그램을 수정한 후에 제작에 들어가는 상황이 포함된다, ‘실행(I)’은 프로그램을 실제 교육 현장에 사용하고 교육과정 내에 설치하며 계속 유지하고 변화를 관리하는 일이 포함된다. 마지막으로 ‘평가(E)’는 프로그램의 적절성을 결

정하는 과정으로 실행된 프로그램의 유용성과 가치를 판단하는 총괄평가의 성격을 지니고 있다(이상갑, 2001; 이소이, 2011; 이옥화 외, 2008; 이창훈, 2007).

ADDIE 모형은 간 단계가 유기적으로 연관되어 있으며, 각 과정에 따른 역할, 세부 활동 및 그 산출물의 관계는 다음 <표 II-3>와 같다(박성익 외, 2011).

<표 II-3> ADDIE 모형의 단계별 특성

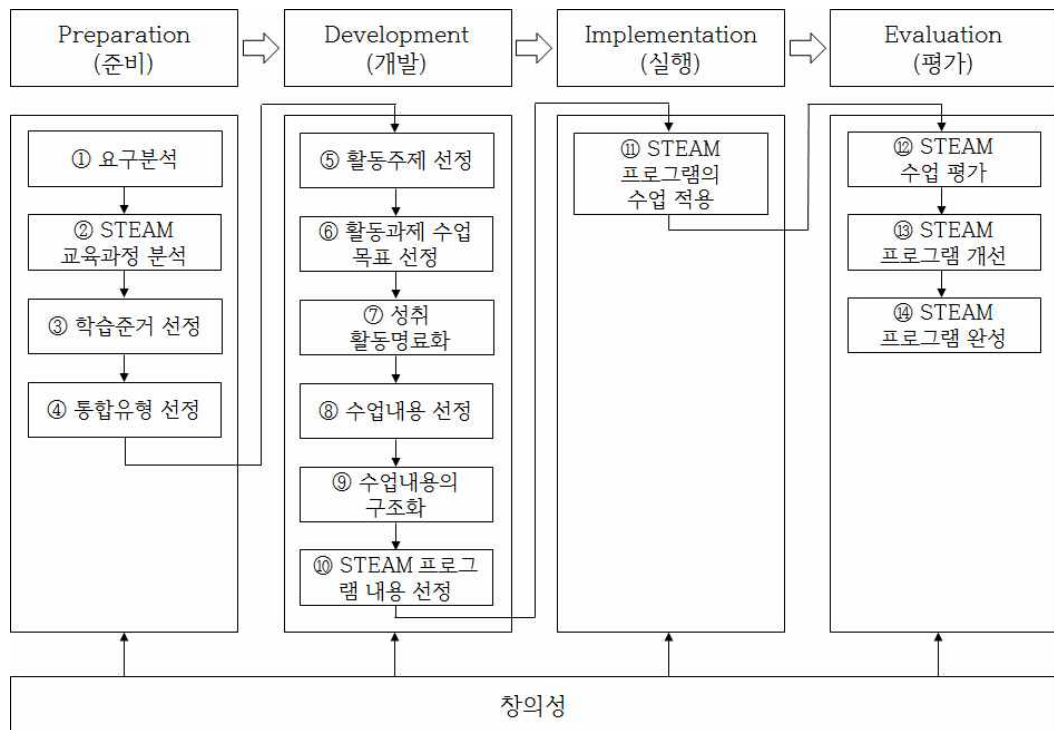
과정	역할(기능)	세부 단계(활동)	산출물
분석	학습내용(what)을 정의하는 과정	요구, 학습자, 환경, 과제 분석	요구, 교육 목적, 제한점, 학습 과제
설계	교수방법(how)을 구체화하는 과정	성취행동 목표 진술, 평가도구 개발, 교수전략 및 매체선정	성취행동 목표, 교수 전략 등을 포함한 설계명세서
개발	교수자료를 제작하는 과정	교수자료의 제작, 형성평가의 실시 및 교수자료의 수정	완성된 프로그램
실행	교수자료를 실제 상황에 적용하는 과정	교수자료의 사용 및 관리	실행된 프로그램
평가	교수자료의 효과성, 효율성을 결정하는 과정	총괄 평가	프로그램의 가치와 평가 보고서

(2) PDIE 모형

김진수(2011)는 STEAM 교육 프로그램의 체계적인 개발을 위한 PDIE 모형을 개발하여 제안하였다. PDIE 모형은 준비(Preparation), 개발(Development), 실행(Implementation), 평가(Evaluation)의 4단계를 통한 개발 모형이다.

PDIE 모형은 기존의 PDI 모형과 ADDIE 모형의 단점을 줄이고 장점을 취하여 구안되었다. ADDIE 모형의 경우 준비단계에 해당하는 Analysis(분석)와 Design(설계)의 단계가 복잡하고 교사가 사용하기에 복잡하다는 단점이 있고 PDI 모형의 경우 수업의 실행 단계가 빠져 있어서 이를 개선한 PDIE 모형을 제시한 것이다.

[그림 II-4]는 PDIE 절차 모형에서 각 단계별로 세부 절차를 나타낸 것이다. 특히 STEAM 교육의 특성에 맞게 각 단계별로 학생들의 창의성 교육이 가능하도록 하는 것이 PDIE 절차 모형의 특징이다(김진수, 2011).



[그림 II-4] PDIE 절차 모형의 단계별 세부 절차



3. 프로그래밍 교육

1) 컴퓨터 과학 교육과정과 프로그래밍

교육 분야에 컴퓨팅 사고력을 도입하기 위한 연구들이 진행되어 왔다. 2011년 미국의 ISTE와 CSTA에서는 컴퓨터과학 교사 및 연구자를 대상으로 설문 을 통해 컴퓨팅 사고력에 대한 조작적 정의를 내렸다. 또한 K-12 컴퓨터 과학 교육 과정에 컴퓨팅 사고력을 하나의 큰 주제로 포함시키고 초중고 단계별로 점차 심화되는 구체적인 교육 목표를 설정하였다. 이러한 컴퓨터 과학 교육의 근본적인 변화와 함께 다양한 형태의 교육 방법을 적용하여 컴퓨팅 사고력을 보다 쉽게 접근하기 위한 노력을 기울이고 있다(이영준 외, 2013).

영국은 2013년 ‘새로운 학교 교육과정’ 방안 발표에서 5세부터 모든 학생들이 컴퓨팅 교육을 받아야 할 것을 강조하면서, 향후 미래 영국 번영의 핵심이 컴

퓨팅이라고 강조하였다. 학습자 연령대에 따라 총 4단계 교육을 제시하면서 2014년부터 컴퓨팅 교육을 실시할 예정이다. 세계적인 대기업들과 캠페인을 전개하여 10만여 학생들에게 디지털 교육 훈련을 제공함으로써 컴퓨팅 사고력의 중요성을 알리고, 흥미를 부여하여 학교 교육 외에 비형식적 교육 환경에서도 스스로 학습할 수 있는 기회를 제공하였다(이영준 외, 2013).

이스라엘은 대학 입학시험에서 컴퓨팅 사고력 능력을 평가할 수 있도록 함으로써 학생들이 자연스럽게 컴퓨팅 사고력과 관련이 깊은 컴퓨터 과학 교과를 이수하도록 유도하고 있으며, 학생의 90% 이상이 대학 입학 시험에서 컴퓨터 과학 교과를 선택하고 있다. 중국은 2000년부터 컴퓨터 과학 교육을 의무화하였으며 전체 교육 시간의 70% 이상을 실습으로 구성하여 컴퓨터가 어떤 역할을 갖고 있는지 경험하고, 이를 활용할 발판을 마련하고 있다. 인도 또한 초등학교 때부터 컴퓨팅 사고력을 도입한 다양한 문제 해결 방법을 교육하고 있으며 컴퓨터 과학 교과 80%를 컴퓨팅 사고력 역량 강화 및 컴퓨팅 사고력을 통한 문제 해결을 강화하고 있다(이영준 외, 2013).

컴퓨팅 사고력 향상을 위한 교육 활동을 위해서는 교육용 프로그래밍 언어를 활용하거나, 컴퓨팅 기기 없이도 컴퓨팅 사고력을 경험하는 방법, 교구를 활용한 방법 등 다양한 방법을 도입하여 교육을 실천하고 있다(이영준 외, 2013).

컴퓨팅 사고력의 추상화와 자동화를 모두 학습할 수 있고, 컴퓨팅 사고력을 활용한 문제 해결에 가장 적절한 교육은 프로그래밍을 통한 문제 해결 학습이라 할 수 있다. 프로그래밍 교육은 컴퓨터 과학적 개념과 원리의 습득과 실생활의 복잡한 문제를 창의적으로 해결하는 능력을 길러주기에 효과적이다. 최근에는 풍부한 미디어 기반 프로그래밍 환경이 주를 이루는 교육에 적합한 교육용 프로그래밍 언어가 등장하면서 학습자의 흥미를 이끌어 내고 있으며 자발적인 학습을 유도하고 있어 컴퓨터 과학을 시작하는 학습자에게 매우 적절한 도구가 되고 있다(이영준 외, 2013)

2) 프로그래밍과 STEAM 교육

STEAM 교육이 학습자의 창의력과 문제해결능력을 강조하며 주목을 받으면

서 프로그래밍 교육을 기반으로 STEAM 교육을 실시하는 연구가 진행되었다.

STEAM 교육에 대한 관심이 높아지면서 컴퓨터과학 요소를 융합하는 연구가 계속 되고 있다. 컴퓨터과학 중심의 융합교육의 근거는 우리나라와 미국의 컴퓨터과학 교육과정에서도 찾을 수 있다. 컴퓨터과학 교육의 목표 중 하나로 다른 학문들과 통합되어 새로운 형태로 확장, 발전시켜 나가는 융합 학문 분야를 개척할 수 있는 역량과 태도를 목표로 설정하였고, 교과별로 정보통신 기술 활동 예시를 제시하여 타 교과 활동에 정보통신기술을 활용할 수 있도록 하였다(교육과학기술부, 2011b; 교육인적자원부, 2005). 또한 미국컴퓨터교육학회(CSTA)는 초등학교 단계의 정보과학 교육은 컴퓨팅 사고에 대한 간단한 아이디어와 기술의 기본적인 개념을 통합함으로써 기본적인 컴퓨터과학 개념을 소개하도록 하였다. 컴퓨터과학 학습 경험은 능동적인 학습, 창의성 및 탐구에 중점을 두어 설계되어야 하고 사회과학, 언어, 수학, 과학 등의 교과영역에 포함될 수 있음을 제시하였다(CSTA, 2011; Barr, V., & Stephenson, C., 2011).

김정아는 융합형 인재 양성을 위한 IT 기반 STEAM 교수·학습 방안을 IT 활용 사례에 따라 IT 학습 콘텐츠를 활용하는 형태, IT 학습 도구로 활용하는 형태, IT를 교수·학습 내용으로 활용하는 형태로 구분하였고 STEAM 교육에서 프로그래밍 교육을 학습 도구로서, 학습 내용으로서 활용할 수 있음을 제시하였다(김정아 외, 2011).

송정범(2010), 신승용(2011) 등의 연구에서는 로봇 프로그래밍을 활용한 STEAM 교육 프로그램을 진행하였다. 로봇은 초등학생들의 발달 단계에 적합하고 실세계와 가장 유사한 경험을 제공할 수 있다는 장점이 있다. 송정범(2010)은 교실친화적 로봇교육 모형 및 프로그램 개발을 통해 학습자의 수학·과학 교과 학업성취도와 교과 태도 수준 향상에 긍정적인 영향을 줌을 밝혔다. 또한 신승용(2011)은 로봇을 활용한 프로그래밍 교육에 참여한 학습자의 요구 분석을 통해 학습자들의 학습의도에 영향을 주는 요인을 파악하고 이를 통해서 밝혀진 요인들을 중심으로 프로그래밍 교육활동에 참여하는 학생들의 학습 지속 의도를 분석하였다.

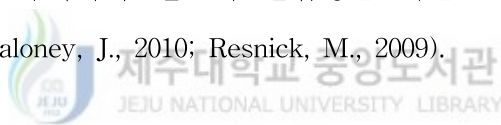
스크래치와 코듀 등의 교육용 프로그래밍 언어를 활용한 프로그래밍 중심의 STEAM 교육 프로그램에 대한 연구도 지속적으로 이루어지고 있다.

이러한 연구들은 과학 주제 학습 과정에서 프로그래밍 활동을 통해 게임이
나 디지털 스토리텔링을 통해 창의적, 논리적 사고력, 과학과 관련된 정의적
영역에 대한 변화를 살펴보고 있다(김태훈, 김병수, 김종훈, 2014; 김태훈, 김종
훈, 2014; 김태훈, 김종훈, 2013; 김태훈, 김종훈, 2012c; 오정철 외, 2012).

3) 교육용 프로그래밍 언어

(1) 스크래치

스크래치는 프로그래밍을 처음 접하는 학생들을 위한 교육용 프로그래밍 언
어이다. 직관적인 블록 쌓기 방식으로 학습자가 적응하기 쉽고 문법에 대한 부
담 없이 프로그래밍을 할 수 있다. 또한 이미지, 음악, 소리와 같은 다양한 미
디어를 손쉽게 조작할 수 있고, 웹을 통한 공유의 장을 지원하기 때문에 쉽게
예제를 찾아볼 수 있다는 장점을 가지고 있다. 그래서 스크래치는 프로그래밍
을 처음 접하는 초등학생을 위해 프로그래밍을 경험하기 위한 도구로 많이 사
용되고 있으며 컴퓨터과학 전공의 신입생을 위한 프로그래밍 학습 도구로도
사용되고 있다(Maloney, J., 2010; Resnick, M., 2009).



<표 II-4> 스크래치 프로그래밍 관련 연구

연구자 (발행연도)	연구 내용	
Romeike, R. (2007)	제목	Applying creativity in CS high school education: criteria, teaching example and evaluation
	내용	고등학교에서의 컴퓨터과학 학습에서 창의성에 대한 평가 준거를 설정한 프로그래밍 입문 학습에서 학습자들의 동기유발과 흥미, 학습 성취도가 증진되었다.
송정범, 조성환, 이태욱 (2008)	제목	스크래치 프로그래밍 학습이 학습자의 동기와 문제해결력에 미치는 영향
	내용	초등학교 6학년 학생들의 재량활동 시간을 활용하여 스크래치 프로그래밍 학습한 결과 학습자의 내재적 동기와 문제해결력 향상에 효과가 있는 것으로 나타났다.
조성환, 송정범, 김성식, 이경화 (2008)	제목	CPS에 기반한 스크래치 EPL이 문제해결력과 프로그래밍 태도에 미치는 효과
	내용	중학교 남녀 학습자에게 CPS 모형 기반의 스크래치 프로그래밍 학습이 문제해결력 향상과 프로그래밍 교육에 대한 태도에 긍정적인 영향을 미쳤다.
김종진,	제목	교육용 프로그래밍 언어인 로고와 스크래치 교재 개발 및 비교 실험

연구자 (발행연도)	연구 내용	
현동림, 김승완, 김중훈, 원유현 (2010)	내용	초등학생을 대상으로 로고와 스크래치 프로그래밍 학습 후, 창의성 검사 결과를 교차 분석하여 비교한 결과, 두 언어 모두 창의성에 긍정적인 영향을 미쳤다. 특히 로고는 창의성 영역의 유창성, 스크래치는 추상성과 저항 영역에 긍정적인 영향을 주었다.
한선관, 김수환, 서정보 (2010)	제목	스크래치 프로그래밍을 활용한 게임중독 치료 프로그램의 개발
	내용	초등학교 게임중독 고위험군을 대상으로 게임중독 치료 대안활동과 함께 스크래치 교육 프로그램을 적용한 결과, 게임중독 치료에 긍정적인 답변을 얻고, 휴식 시간 게임 접속 시간이 감소된 것으로 분석되었다. 관찰과 면접을 통합 분석 결과 자아존중감과 창작의 기쁨, 자아제어 능력에도 긍정적인 영향을 미쳤다.
안경미, 손원성, 최윤철 (2011)	제목	스크래치 프로그래밍 교육이 초등학생의 학습 몰입과 프로그래밍 능력에 미치는 효과
	내용	초등학교 고학년을 대상으로 스크래치 프로그래밍 학습 집단과 순서도 학습 집단을 비교한 결과, 스크래치 프로그래밍 학습이 상대적으로 학습몰입 수준과 프로그래밍 능력에 긍정적인 영향을 미침을 보였다.
박용철, 이수정 (2011)	제목	스크래치 프로그래밍 교육이 초등학생의 자기 주도적 학습 능력에 미치는 효과
	내용	초등학교 6학년 학생들에게 스크래치 프로그래밍을 실시한 결과, ICT 소양 프로그래밍 수업에 비해 자기주도적 학습 능력의 하위요소인 개방성, 내재적 동기, 자율성 영역의 신장에 유의한 차이를 보였다. 특히 ICT 활용 능력이 우수한 학생들에게 그러한 신장이 크게 나타났다.
함성진, 양창모 (2011)	제목	스크래치를 이용한 초등학교 컴퓨터 교육과정 설계
	내용	프로그래밍 교육을 위한 교육과정을 설계하여 전문가 집단으로부터 학습 대상, 교육 요소, 학습 내용, 학습량, 교수학습지도안의 적정성 등의 거의 모든 영역에서 적절하다는 평가를 받았다.
송정미 (2011)	제목	스크래치를 활용한 정보과학적 사고 기반 퍼즐 교육
	내용	문제해결과정으로서 퍼즐을 도입하여 스크래치 교육과 연계하였다. 학생들이 펜토미노 구체물이 아닌 스크래치를 이용하여 문제를 해결하고 이 과정에서 도형의 개념과 퍼즐의 원리를 습득하도록 하였다.
류충구 (2012)	제목	Scratch가 초등 영재들의 창의적 문제해결력에 미치는 효과
	내용	초등학교 영재학생들을 대상으로 스크래치 프로그래밍 학습이 비주얼베이직 프로그래밍 수업보다 창의적 문제해결력에 긍정적인 효과를 미친다.
조준필 (2012)	제목	기술교육에서 중학생의 논리적 사고력 함양을 위한 스크래치 학습 프로그램 개발
	내용	중학생들의 기술교육에 있어서 논리적 사고력을 함양시키기 위한 스크래치 학습 프로그램을 전문가의 검증을 통해 개발하였음.
박정신,	제목	프로그래밍 입문 수업에서 스크래치 활용 효과분석

연구자 (발행연도)	연구 내용	
조석봉 (2012)	내용	전문대학의 컴퓨터 전공 학생들이 프로그래밍언어 수업에서 스크래치를 활용하여 비활용반과의 비교에서 중간고사와 기말고사의 점수를 비교하여 알고리즘 구상과 프로그래밍 작성에 효과가 있음을 보였다.
김병수, 김종훈 (2013)	제목	CPS 모형 기반 스크래치 프로그래밍 학습이 언어 창의성에 미치는 영향
	내용	창의적 문제 해결(CPS) 모형 기반의 스크래치 교육 프로그램이 학습자의 창의성에 미치는 영향을 분석하였다. 기존 연구들과는 달리 TTCT 도형 검사지를 사용하지 않고 스크래치 블록 프로그래밍 활동을 언어 창의성과 연결시켜 언어 창의성 신장에 효과가 있음을 보였다.

<표 II-4>에서 보듯이 스크래치 프로그래밍과 관련된 연구들의 목적들은 논리적 사고력 신장, 몰입도 향상, 자기주도적 학습 능력 향상, 문제해결력 향상, 창의성 향상, 게임 중독 치료 등으로 매우 다양하였다. 본 연구에서 개발하고자 하는 STEAM 교육 프로그램이나 컴퓨팅 사고력의 증진에 대한 상관연구 및 비교 실험 연구는 아직 많지 않은 편이다.

(2) 코듀(Kodu)

코듀는 Logo, Alice, Greenfoot, Scratch 등의 영향을 받아 마이크로소프트 Fuse 연구소(Microsoft Fuse Research)에서 개발한 프로그래밍 언어로서 고급 언어, 비주얼 언어, 인터프리터 언어의 특징을 갖는다. 프로그래밍 모델을 단순화시켜 프로그래밍을 처음 시작하는 학생들도 쉽게 프로그래밍을 접할 수 있게 하여 프로그래밍 학습에 대한 장벽을 낮추고자 하였다. 쉽게 배울 수 있지만 언어에서 지원하는 프로그래밍 표현을 활용하여 학생들이 원하는 다양한 상황의 게임을 만들어 낼 수 있다. 실시간 3D 게임 환경에 통합되어 있고 콘솔 게임 수준의 높은 그래픽 수준을 보여주며 직관적인 사용자 인터페이스를 갖고 있어 어린 학생들의 흥미를 끌기에 적합하다(MacLaurin, M., 2011).

PC 또는 Xbox에서 구동되는 게임을 만드는 용도로 개발되었지만 최근 미국, 영국 등지에서 프로그래밍 학습과 게임 개발을 통한 STEM 교육 도구로서 코듀를 사용하고 있다.

사용자는 코듀 내에 존재하는 캐릭터(물고기, 오토바이 등)들을 프로그래밍하여 각 캐릭터가 자신이 존재하는 세계(world)내에서 다른 캐릭터와 상호작용

하고 높은 수준의 인공지능의 로봇이 되도록 만들 수 있다. 각 캐릭터의 프로그래밍은 페이지 안에서 정사각형 모양의 타일을 추가하여 이루어지며 사용자는 명령을 프로그래밍하여 각 캐릭터를 동작시킬 수 있다. 명령은 “When, Do” 절 형태로 되어 있으며 각 절은 조건과 동작으로 이루어진다. When 다음에는 조건을 체크하는 센서가 오고, Do 절에는 동작을 지시하는 동작기가 온다. [그림 II-5]은 다른 언어를 학습할 때 가장 먼저 실행해 보는 “hello world” 프로그램처럼 코딩에서 실행할 수 있는 가장 간단한 명령의 예로 ”(When) 빨간색(red) 사과(apple)를 보면(see) (Do)그 방향으로(toward) 빠르게(quickly) 이동하라(move)는 명령이다(김태훈, 2013).



[그림 II-5] 코딩의 프로그래밍 예제

코딩을 활용한 프로그래밍 교육은 다음과 같은 측면에서 장점을 갖는다.

첫째, 언어 자체가 갖고 있는 학생들의 흥미를 유발하는 요소이다. 3D 그래픽 환경과 타일 배치 형태의 프로그래밍 형식, 프로그래밍을 통해 명령을 내릴 수 있는 귀여운 캐릭터 등은 프로그래밍을 처음 경험하는 초등학생들이 프로그래밍에 좀 더 쉽게 다가갈 수 있도록 해 준다.

둘째, 학생들을 자연스럽게 문제 상황으로 이끌 수 있다는 장점이 있다. 코딩은 3D 그래픽 환경에서 프로그래밍을 작성하는데 개발 환경 내에서 물리엔진이 이미 작용되기 때문에 실시간의 물리 환경과 매우 유사하다. 실세계와 유사한 물리법칙 안에서 문제를 해결해야 하기 때문에 교사가 문제 상황을 만들어 내고 학생들을 안내할 수 있다(MacLaurin, M., 2011).

셋째, 3D 환경에서 문제에 알맞은 환경(지형, 스토리)을 생성해 내는 것 자체가 STEAM의 공학, 예술과 밀접한 관련을 갖는다. 스크래치에서 그림판을 이용해 무대나 스프라이트를 만들어 내는 것을 넘어서서 문제를 해결하기 위한 지형을 만들어 내고 객체를 추가하는 것이 학습의 일부분이 된다.

넷째, 문법이 간단하여 캐릭터를 동작시키는 프로그램 작성에 대한 부담이 적을 뿐 아니라 간단한 프로그래밍을 통해 변수, 조건, 객체지향 등의 다양한 컴퓨터과학의 요소를 학습할 수 있다(Stolee, K. T. & Fristoe, T., 2011).

마지막으로 프로그램을 작성하거나 수정할 때 그 내용을 바로 확인할 수 있기 때문에 결과물을 쉽게 만들어 낼 수 있고 오류 수정도 용이하다. 프로그래밍 결과를 확인하기 어렵다면 학생들이 지겨움을 느끼기 쉬운데 이를 극복할 수 있다(이찬도, 2010).

프로그래밍 도구로서 코듀를 활용한 국내·외의 연구는 <표 II-5>와 같다.

<표 II-5> 코듀를 활용한 프로그래밍 교육에 대한 연구

연구자 (발행연도)	연구 내용	
Stolee, K. T., Fristoe, T. (2011)	제목	Expressing computer science concepts through Kodu Game Lab
	핵심 내용	코듀가 갖고 있는 컴퓨터 과학 개념을 분석하고 코듀의 이용자들의 패턴분석을 통해 게임 플레이 보다 게임 설계 및 제작에 많은 시간을 할애하고 있음을 제시하였다. 커뮤니티에 올라온 소스를 분석하여 많은 이용자들이 코듀에서 제공하고 있는 정보과학적인 요소를 사용하고 있으며 이상의 결과를 통해 코듀가 단지 프로그래밍을 시작하는 언어로서가 아니라 컴퓨터 과학 개념을 학습하여 주류 프로그래밍 언어로 전환할 수 있는 시작점으로서의 역할을 할 수 있음을 제시하였다.
	시사점	코듀를 활용한 컴퓨터 과학 개념과 프로그래밍 학습에 대한 가능성을 제시하였다.
Fowler, A., Fristoe, T., MacLauren, M. (2012)	제목	Kodu Game Lab: a programming environment
	핵심 내용	어린 학생들을 위한 프로그래밍 도구로 코듀와 코듀가 가지는 장점을 소개하였다. 코듀의 문법과 코듀가 가지는 프로그래밍 언어적인 요소(객체, 변수, 조건문, 부울 로직, 클래스, 페이지, 에러처리)를 설명하였다.
	시사점	코듀를 이용하여 프로그래밍 언어의 요소를 학습할 수 있음을 제시하였다.
Fowler, A., Cusack, B. (2011)	제목	Enhancing introductory programming with Kodu Game Lab an exploratory study
	핵심 내용	중학생을 대상으로 학교 교육과정에 프로그래밍 개념을 소개하기 위한 도구로 코듀를 활용하여 그 효과성을 검증하였다. 전통적인 프로그래밍 언어(C++, Java 등)를 학습할 때와 비교하여 학생의 학습 동기, 즐거움, 강화, 자신감의 측면에서 긍정적인 결과를 얻었다.
	시사점	코듀를 이용한 학습이 프로그래밍 개념을 위한 소개 활동에 적합한 언어임을 제시하였다.

연구자 (발행연도)	연구 내용	
Fristoe, T. et al. (2011)	제목	Say it with systems expanding Kodu's expressive power through gender-inclusive mechanics
	핵심 내용	여학생들이 코듀 프로그래밍 활동을 통해 STEM 교육, 특히 프로그래밍에 관한 흥미를 가질 수 있도록 캐릭터 간에 상호작용이나 동적인 관계를 강화하는 요소들을 추가하여 긍정적인 피드백을 얻었다.
	시사점	코듀를 이용한 프로그래밍 활동이 여학생을 포함한 모든 학생들에게 프로그래밍과 게임설계에 대한 즐거움을 줄 수 있다고 제시하였다.
Kane, L., et al. (2011)	제목	Studio K: a game design curriculum for computational thinking
	핵심 내용	Studio K 커리큘럼은 컴퓨팅 사고를 신장시키기 위하여 코듀를 이용한 게임 설계 기반 학습에 대한 연구를 진행하였다. 학습자에게 게임 디자이너로서의 상호작용, 이야기 등을 만드는 경험을 제공하기 위한 컴퓨팅 사고, 게임 설계 프레임워크를 제안하였다.
	시사점	코듀를 통한 게임 개발 활동을 통한 컴퓨팅 사고 신장의 가능성을 제시하였다.
Touretzky , D. et al. (2013)	제목	Accelerating K-12 computational thinking using scaffolding, staging, and abstraction
	핵심 내용	학생들이 프로그래밍과 컴퓨팅 사고에 대하여 좀 더 깊게 이해할 수 있도록 Alice와 Lego NXT-G를 학습하기 이전 단계로 코듀를 이용하는 교육프로그램을 구성하였다. 변수, 조건문, 병렬화, 반복, 상태 등의 개념을 비교하여 좀 더 쉽게 이해할 수 있음을 확인하였다.
	시사점	프로그래밍을 학습하기 위한 시작점 단계의 언어로서 코듀를 제안하였다.
김병수, 김중훈 (2012)	제목	초등 예비교사의 컴퓨터과학에 대한 인식 변화를 위한 컴퓨팅 사고 기반 알고리즘 학습의 설계 및 적용
	핵심 내용	비전공자인 초등 예비교사를 대상으로 프로그래밍 언어와 컴퓨팅 사고를 접하기 위한 도구로서 코듀를 통한 알고리즘 학습을 진행하였다.
	시사점	프로그래밍과 컴퓨팅 사고 학습 도구로서 코듀가 의미있는 도구가 될 수 있을 것이다.
임화경, 조용남 (2012)	제목	Kodu 비주얼 프로그래밍 언어를 사용한 초등학생의 창의적 3D 게임 프로그래밍 학습
	핵심 내용	6학년 초등학생을 대상으로 30차시에 걸쳐 코듀를 이용한 게임 제작 수업을 진행하여 학습자들의 성찰글과 설문지를 통해 창의적 사고력에 대한 변화와 게임 프로그래밍에 대한 학습효과를 분석하였다.
	시사점	코듀를 활용한 프로그래밍 학습이 학생들의 창의력 변화에 긍정적인 효과를 주었지만 좀더 객관적인 검사지를 통한 검증이 필요하다.

Kathryn T. Stolee(2011), Allan Fowler(2012)는 코듀가 초보자들도 쉽게 프로그래밍을 할 수 있고 컴퓨터 과학적인 요소가 포함되어 있어 초보 학습자들의 프로그래밍 도구로서 가능성을 갖고 있음을 제시하였다.

Fowler(2012)와 Fristoe 외(2011)의 연구에서는 각각 중학생과 여중학생들을 대상으로 코듀를 활용한 프로그래밍 수업을 진행하였고 프로그래밍 학습에 대한 흥미 및 자신감들의 정의적인 요소에서 긍정적인 효과가 있음을 제시하였다.

Kane 등의 연구(2012)와 김병수, 김종훈(2012)의 연구에서는 컴퓨팅 사고를 학습하는 도구로 코듀를 활용하는 방법을 제시하였지만 실제로 학습에 대한 결과를 제시하지는 못했다.

임화경(2012)의 연구에서는 학생들의 성찰글을 이용하여 코듀를 활용한 프로그래밍 학습이 학생들의 창의적 사고력에 긍정적인 영향을 미쳤다는 결과를 제시하였지만 좀 더 객관적인 검사도구를 이용한 검증이 필요할 것으로 보인다.

코듀와 관련된 사전 연구를 분석한 결과 코듀가 컴퓨터과학적인 요소를 갖고 있고 초보자도 쉽게 프로그래밍 할 수 있는 쉬운 언어이기 때문에 초등학생들의 프로그래밍 학습에 적합하다는 사실을 알 수 있었다. 또한 효과성 검증 결과 학습 흥미에서도 긍정적인 결과를 보여주고 있었다. 하지만 코듀를 이용한 프로그래밍 학습의 인지적 효과에 대한 언급이 없어 코듀가 학습자의 인지적인 측면에 어떠한 영향을 줄 것인지에 대한 실험을 진행할 필요가 있었다.

(3) 코듀를 활용한 프로그래밍 학습이 논리적 사고력에 미치는 영향

코듀가 초등학생 또는 프로그래밍을 처음 접하는 학습자들에게 프로그래밍과 컴퓨팅 사고를 학습하기 위한 도구로서 적합한지를 살펴보기 위해 코듀를 활용한 프로그래밍 학습 프로그램을 개발하였다. 개발한 학습 프로그램이 학습자의 논리적 사고에 미치는 영향을 알아보기 위해 프로그램 투입 사전과 사후에 GALT를 이용하여 학습자의 논리적 사고력을 측정하였고 이를 활용하여 연구결과를 분석하였다.

① 연구 대상

제주특별자치도에 위치한 H초등학교 6학년 중 2개 학급을 각각 실험집단과 비교집단으로 선정하였다. 각 반은 23명으로 실제 프로그래밍 경험이 있는 학생은 거의 없었지만 대부분의 학생들이 프로그래밍이란 용어는 알고 있었다. 실험집단에는 코듀를 이용한 프로그래밍 교육을 실시하였으며 비교집단에는

초등학교 현장에서 많이 이루어지고 있는 워드프로세서 및 파워포인트를 학습하는 ICT 활용 수업을 진행하였다.

② 연구 설계

실험집단의 코뮤티를 활용한 프로그래밍 학습을 위하여 총 10차시의 수업을 창의적 체험활동 시간을 이용하여 투입하였다.

학생들의 논리적 사고력을 측정하기 위한 도구로 GALT 검사를 사용하였다. 연구에서 사용한 검사지는 노정원(1997)의 연구에서 사용된 21문항의 원본 검사지를 사용하였고 의미를 손상시키지 않는 범위 내에서 학생들에게 친숙한 이름과 보충 설명을 추가하였다.

코뮤티를 활용한 프로그래밍 학습이 학습자의 논리적 사고에 미치는 영향을 알아보기 위해 교육프로그램 투입 전과 후에 실험집단과 통제집단에 GALT 검사지를 실시하여 그 결과를 분석하였다.

③ 코뮤티를 활용한 프로그래밍 학습 프로그램 개발

코뮤티를 활용한 프로그래밍 학습을 위해 교육 프로그램의 주제 및 활동을 선정하였다. 코뮤티와 기존에 사용된 교육용 프로그래밍 언어를 이용한 프로그래밍 학습에 대하여 연구한 문헌을 검토하였다. 코뮤티를 활용한 프로그래밍의 학습 주제를 추출하였고 이를 투입하기 위해 총 10차시의 투입계획을 수립하였으며 이를 <표 II-6>에 제시하였다.

프로그래밍 경험이 없는 초등학생들에게 프로그래밍을 지도하기 위해서는 언어 자체에 대한 학습이 필요하지만 단순한 프로그래밍 언어의 학습만으로는 학생들의 논리적 사고력을 증진시키기 어렵다. 코뮤티로 제작한 게임이나 코뮤티의 그래픽 환경 등에 대한 흥미가 아니라 프로그래밍을 통한 문제해결에 집중할 수 있도록 하는 장치가 필요하다. 이를 위해 각 차시의 수업을 프로그래밍 문제를 제시한 도입 부분과 이를 해결하기 위한 프로그래밍 요소를 지도하는 부분, 이를 사용해 문제를 해결하는 부분, 마지막으로 학습한 내용을 활용해 개인별로 창의적인 프로그래밍을 진행하는 단계로 구분하였다. 문제해결을 단계별로 제시함으로써 학생들이 문제해결에 집중할 수 있도록 하였다.

<표 II-6> 투입 내용

차시	학습주제	학습활동
1	코딩과 인사하기	프로그래밍이란? 로봇에게 명령 내리기 프로그래밍 입력 방법 익히기
2	World 만들기	지형 메뉴 익히기 제주도 모양의 World 만들기
3	기본 명령어 익히기	로봇을 움직이는 방법 알아보기 다양한 센서와 동작기 다루기
4	Score를 사용하기	센서를 이용한 조건문 만들기 Score를 이용한 점수기능 익히기 변수로서 Score 사용하기
5	Timer 사용하기	Timer 이용하여 동작 제어하기 Random 이용한 프로그래밍하기
6	Creatable 사용하기	Creatable 기능 이해하기 Creatable 속성의 객체 로봇 만들기
7	Page 사용하기	Page를 사용하여 로봇 제어하기 다양한 동작을 수행하는 로봇 프로그래밍하기
8	로봇으로 대화하기	Hear 센서와 Say 동작기를 이용한 대화하기
9	디지털 스토리텔링하기	토끼와 거북이 이야기 프로그래밍하기
10	프로젝트 발표	학생이 직접 만든 프로그램으로 발표하기

④ 연구 결과

실험집단과 비교집단의 논리적 사고력의 변화를 비교 분석하였다. 두 집단 모두 학습자의 수가 23명으로 정규분포를 따르는지 정규성 검정이 필요했지만 GALT 검사지는 각 하위요소별 문항이 적어 정규성 검정이 어렵기 때문에 모든 영역에 대하여 비모수 통계방법을 사용해 검정하였다.

두 집단이 동질집단임을 확인하기 위해 Mann-Whitney 검정 방법을 이용하여 사전 결과를 비교한 결과 모든 논리적 사고력 하위요소 및 논리적 사고력 합계에서 집단 간 차이가 없음을 확인하였다. 두 집단의 사전검사 비교 결과를 <부록 1>에 제시하였다.

코딩을 통한 프로그래밍 학습이 학습자의 논리적 사고력에 미치는 영향을 알아보기 위해 두 집단의 사후검사 결과를 비교하였다. 검증 결과 보존논리에

서 실험집단의 평균이 2.913, 편차가 .765였고 비교집단의 평균이 2.304, 편차가 .765로 나타나 사전검사에 비해 두 집단의 평균이 모두 상승하였지만 실험집단 (.348)의 상승폭 비교집단(.043)에 비해 컸으며 유의확률 $p < .05$ 내에서 유의미한 차이를 보였다. 조합논리에서도 실험집단의 평균이 1.783, 편차가 .850였고 비교집단의 평균이 1.087, 편차가 .949로 나타나 사전검사에 비해 두 집단의 평균이 모두 상승하였으나 실험집단(.783)의 상승폭이 비교집단(.435)에 비해 컸으며 유의확률 $p < .05$ 내에서 유의미한 차이를 보였다. ICT 활용 교육에 비해 코딩을 통한 프로그래밍 학습이 논리적 사고력의 하위요소인 보존논리와 조합논리 향상에 더욱 효과적이라는 결론을 내릴 수 있었다.

집단 내에서의 변화를 살펴보기 위해 각 집단의 사전·사후검사 결과에 대해 Wilcoxon 부호순위 검정을 실시하였고 그 결과는 <부록 2> 같다.

실험집단의 경우 조합논리와 논리적 사고력 합계에 대해서 유의미한 증가를 보였고 비교집단의 경우 조합논리에 대해서 유의미한 결과가 나타났다.

이러한 결과를 해석해 본다면 두 집단 모두 조합 논리 부분에서 유의미하게 상승하긴 했지만 집단 간 비교에서 실험집단의 상관, 조합논리의 평균의 비교집단에 비해 유의미하게 높은 것으로 나타난 점, 그리고 실험집단의 사전·사후비교에서 논리적 사고력 합계가 유의미하게 상승한 점에서 코딩을 활용한 프로그래밍 교육이 학습자의 논리적 사고력 향상에 긍정적인 영향을 주는 것을 확인할 수 있었다.

4) 프로그래밍 교수 전략

(1) 디지털 스토리텔링

사람은 서로 이야기를 나누며 살아가는 사회적 존재이다. 이야기를 통해서 정보를 교환하고, 자신의 감정이나 의견을 표현한다. 잘 짜여진 이야기는 타인에게 강한 정서적 공감을 불러일으키고 감동을 유발하여 교훈을 전달하기도 한다(이하룡, 2013).

어떠한 대상에 대한 또는 그 안에 숨어 있는 이야기는 그에 대한 관심을 불러 일으킨다. 이야기 속에는 그 사람이 가지는 외모나 능력, 어떤 물건의 디자인이나 기능을 통한 이해를 넘어서는 감동을 갖고 온다. 이야기를 통해 누군가

또는 무엇인가에 더욱 관심을 갖게 되고 그러한 관심은 더 많이 알고 싶다는 욕구를 불러일으키게 된다.

사람들에게 이야기가 필요한 이유는 기억을 잡아두기 위함이다. 대부분의 사람이 어떤 사실이나 수치, 지식에 관련한 말을 기억해야 할 때 단적으로 머릿속에 입력하는 것보다 이야기를 들을 때 더 잘 몰입하고 감동받는다. 두 번째 이유는 이야기가 사람의 마음을 변화시키기 때문이다. 이야기에는 힘이 있기 때문에 그것이 누군가를 향해 전달되었을 때 듣는 사람으로 하여금 어떠한 변화를 이끌어 낼 수 있다. 마지막으로 이야기는 인간이 세상을 이해하기 위해 필요하다. 인간은 추상적인 설명보다 구체적인 이야기를 통해 훨씬 더 잘 이해한다(EBS 다큐프라임 ‘이야기의 힘’ 제작팀, 2011). 즉, 이야기를 통해 좀 더 잘 기억하고 감정의 변화와 대상에 대한 깊은 이해를 이끌어 낼 수 있다는 의미이다.

스토리텔링(storytelling)은 이야기를 들려주는 활동이라는 사전적 의미를 가지며 우리나라에서는 구연이라는 영어로 많이 쓰인다(이진희, 2009).

연구자들이 스토리텔링에 대한 다양한 정의를 내리고 있다.

나보라(2011)는 스토리텔링을 이야기 안에 담겨있는 지식과 교훈을 학습자에게 전달하여 효과적인 학습이 이루어지도록 도와주는 교육 방법의 하나로 정의하였고 McWilliams(2011)는 눈앞에 있는 청중에게 스토리의 내용을 묘사하기 위해 언어, 발성, 신체적인 움직임 사용하는 것을 스토리텔링이라고 정의하였다.

스토리텔링을 교육에 접목하여 그 효과에 대한 연구도 다양하게 이뤄지고 있으며 스토리텔링이 교육적 수단으로 사용할 수 있는 효율적인 전략임이 제시되었다.

Egan(1989)은 아이들은 정서적으로 이야기 형태에 더 잘 끌리며, 낯설고 신비스러운 이야기 일수록 아이들의 지적활동을 촉진시킨다고 하였고, Schank(1999)는 스스로 말하는 이야기에 대해 더 많이 생각할수록 더 많은 것을 기억하게 된다고 하였다.

또 Galletts(2005)는 스토리텔링의 경험을 통해 정보회상능력이 향상된다고 하였고, Wells(1986)는 학습자의 스토리텔링에 대한 경험을 교과 학업 성취도를

가장 잘 예측할 수 있는 변인으로 꼽았다.

스토리텔링은 이야기 안에 담겨있는 지식과 교훈을 학습자에게 전달하는 점에서 교육적으로 중요한 수단으로 인식되고 있다. 또한 스토리텔링은 수업 내용을 구체적 맥락과 연관 속에서 파악할 수 있도록 지원하며 학습자들이 수업 내용의 의미를 깊이 있게 이해하는데 도움을 준다. 또한 스토리텔링의 풍부한 맥락은 학습자의 감성과 상상력을 자극하고 학습에 대한 흥미를 유발하며 바람직한 학습 태도를 형성하는데 효과가 있다(권혁일, 2008).

최근에는 스토리텔링을 사전적 의미의 '구연'에만 국한되지 않은 좀 더 넓은 형태로 확장되고 있다.

한아름(2009)은 교육적인 의미로서의 스토리텔링을 보다 확장하여 단순한 이야기를 제시하는 것이 아니라 내용을 이해시키는 모든 활동을 스토리텔링으로 보았다. 또한 염혜진(2006)은 스토리텔링은 상대방에게 알리거나, 설득하거나, 자기 자신을 즐기거나, 자기를 표현하기 위한 효과적이고 창의적인 이야기 행위라고 정의하였다. 스토리텔링을 '구연'에만 국한하지 않은 이유는 최근 들어 음성 언어 뿐만 아니라 문자, 이미지, 동영상, 애니메이션 등 멀티미디어에서도 스토리텔링을 폭넓게 사용하고 있기 때문이다.

스토리텔링 학습 전략을 프로그래밍 수업에 적용하는 시도도 이뤄지고 있다. 이러한 수업은 다음과 같은 효과를 기대할 수 있다(박정호, 2008).

첫째, 교육과 재미의 결합 즉 에듀테인먼트를 실현할 수 있다. 학습자들의 동기를 유지하기 위해서는 흥미로운 수업이 요구되는데 스토리는 흥미를 끌고 주어진 상황에서 문제해결 능력 등을 제시해 줌으로써 학습자 스스로가 흥미를 느끼고 능동적으로 참여할 수 있게 도와준다.

둘째, 몰입을 통한 주체적 학습을 기대할 수 있다. 이야기는 강력한 몰입과 공감 유발의 능력을 가진 일종의 정보전달 방식이기 때문에 쉽게 이해할 수 있고 효과적으로 전달되는 장점을 기대할 수 있다.

셋째, 스토리는 본질적으로 순서적이기 때문에 학습자는 일련의 명령을 실행함으로써 스토리를 시작하고 점차 자신감과 경험을 쌓아감에 따라 상위의 프로그래밍 개념을 습득할 수 있다.

(2) 피지컬 컴퓨팅

STEAM 교육을 위한 프로그래밍 도구로서 스크래치가 갖는 장점 중 하나는 센서를 활용할 수 있다는 것이다. 센서는 최근 스마트폰, 로봇 등에서 사용되는 기술로서 그 자체로 기술 교과와 연관된 학습이 가능할 뿐만 아니라 학생들의 흥미를 유발할 수 있다는 장점이 있다. 센서를 이용하면 컴퓨터 외부의 현상, 즉 주변의 과학 현상(빛, 소리, 전기저항)을 컴퓨터로 입력받아 데이터를 프로그래밍 할 수 있는데 이를 통해 과학 주제와 자연스럽게 연결할 수 있고 이를 다양한 프로그램으로 설계하고 작성하는 과정에서 컴퓨팅 사고를 체험할 수 있다(Resnick, M., 2009).

센서 기반의 프로그래밍 교육에 대한 연구는 주로 로봇을 활용하는 경우가 많았다. 로봇 프로그래밍과 PBL, 스토리텔링 등을 연계하여 문제를 해결하는 학습 프로그램에 대한 연구가 진행되었다(박정호, 2012; 한정혜, 2011; 허경, 2011). 권대용(2013)은 텐지블 프로그래밍 도구를 활용하여 초등학생 저학년에 게 프로그래밍에 대한 흥미를 갖도록 하는 연구를 진행하였다(권대용, 2013).

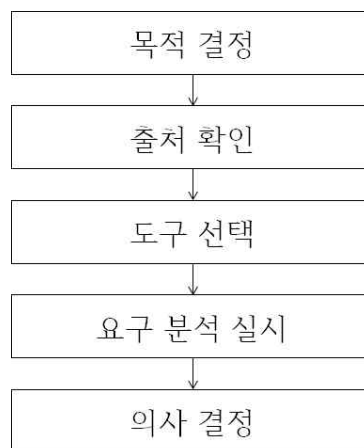
김석희와 유현창(2013)의 연구에서는 체험활동을 위한 센서 기반 4C-STEAM 교육 프로그램을 개발하였다. 초등학교 4학년 교육과정을 재구성한 내용 요소를 스크래치에서 센서의 입력값을 이용한 프로그래밍 통하여 융합교육을 진행하였고, 그 결과 학생의 창의력과 문제해결력 등에서 긍정적인 효과를 얻었다.

III. 프로그래밍 중심 STEAM 교육 프로그램

1. STEAM 교육 프로그램을 위한 요구 분석

교육 요구 분석은 교육 목표를 추출하고, 교육내용을 선정하는 데 있어서 반드시 필요한 활동이다. 교수 학습자의 요구 상황을 제대로 반영한다면 교수 학습자들에게 더욱 유용하게 활용될 수 있을 것이다. 따라서 이 연구는 교수 학습자의 요구 사항을 반영하기 위한 방법을 구체화하기 위해 요구 분석을 수행한다.

본 연구에서는 대표적인 교육 요구 분석 모형 중 하나인 Rossett 요구 분석 모형을 사용하였다. Rossett 모형을 적용한 이유는 기업 교육에서 널리 활용되고 있는 대표적인 교육 요구 분석 모형으로 요구분석의 실행과정에 초점을 두기 때문에 실제 요구 분석 실행자들이 적용하기 쉬운 안내를 제공하기 때문이다. Rossett 모형의 절차는 [그림 III-1]과 같다(Rossett, A., 1987; 이재무, 2013).



[그림 III-1] Rossett 모형

Rossett 모형은 유발된 문제에 관한 요구 분석 목적 결정에서부터 문제 해결

을 위한 의사결정까지 단계별로 제시하고 있다. Rossett의 요구 분석 절차에 따라 본 연구에서 수행한 요구 분석 절차는 다음과 같다.

첫째, 요구 분석의 목적을 선택한다. 본 요구 분석은 컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램을 개발하기 위하여 학습자들의 학습 흥미나 사전 지식 등을 조사하여 교육 프로그램 개발의 기반을 제공하는 데 있다.

둘째, 요구 분석을 위한 출처를 확인한다. 문제 유발 상황이 무엇인지를 확인하고 요구 분석 정보 및 자료를 어디에서 얻을 것인지를 결정한다.

셋째, 요구 분석을 위한 도구 선택으로서 요구 분석 대상들을 대상으로 설문 조사 방법을 이용하여 요구 분석을 실시한다.

넷째, 설문조사 결과를 바탕으로 요구분석을 실시한다.

다섯째, 요구 분석 결과를 분석하여 STEAM 교육 프로그램 개발 시 반영할 요소를 결정한다.

1) 요구 분석 목적 결정

요구 분석 목적 결정 단계에서는 요구 분석을 통해 찾아내고자 하는 정보가 무엇인지를 결정한다. 이 연구에서는 프로그래밍 중심 STEAM 교육 프로그램을 개발하기 위하여 교육 프로그램에 대한 요구 분석을 실시한다. 특히 본 요구 분석에서는 학습자가 프로그래밍에 대하여 얼마나 이해하고 있는가? 그리고 프로그래밍, 과학 학습에 대한 자신감과 흥미가 얼마나 있는가? 마지막으로 STEAM 교육에 대한 경험, 자신감, 흥미가 얼마나 있는가? 에 대한 요구를 분석한다.

2) 출처 확인

출처 확인은 누가 어떤 정보를 가지고 있는가에 대한 정보 출처를 확인하는 단계이다. 프로그래밍 중심 STEAM 교육 프로그램에 대한 요구 정보와 관련하여 정보를 가지고 있는 대상 및 내용을 <표 III-1>에 정리하여 제시하였다.

<표 III-1> 정보원과 정보 내용

정보원	추출 가능한 정보
학습자	프로그래밍 중심 STEAM 교육 프로그램 설계 및 개발에 필요한 학습자의 실태 및 요구되는 정보
교수자	프로그래밍 중심 STEAM 교육 프로그램 설계 및 개발에 필요한 학습 환경이나 교수 전략 등에 대한 지식 및 정보

본 요구 조사에서는 프로그래밍 중심 STEAM 교육 프로그램의 학습자들을 중심으로 요구 정보를 추출한다.

3) 도구 선택

본 연구에서 요구 분석을 위한 도구는 연구자가 직접 설문지를 개발하여 사용하였다. 설문 내용은 프로그래밍 중심 STEAM 교육 프로그래밍 설계 및 개발을 위한 학습자의 실태와 학습요소에 대한 흥미와 자신감을 묻는 내용이 중심이다.

설문 형태는 학습자들의 의견을 효율적으로 알아보기 위한 폐쇄형 질문과 질문에 대한 선택 이유를 묻는 개방형 질문이 혼합되어 있다. 폐쇄형 질문은 5단계 리커트 척도(Likert scale)로 되어 있다.

4) 요구 분석 실시

프로그래밍 중심 STEAM 교육 프로그램에 대한 학습자 분석을 실시하기 위해 설문지를 통하여 요구분석을 실시하였다.

학습자를 분석하기 위한 질문 내용은 다음과 같다.

- 프로그래밍을 학습한 적이 있는가? 학습한 경험이 있다면 어떤 언어를 학습하였는가?
- 프로그래밍 학습에 흥미가 있는가?
- 프로그래밍 학습에 자신이 있는가?
- 과학 학습에 흥미가 있는가?
- 과학 학습에 자신이 있는가?

- STEAM 교육을 경험한 적이 있는가?

요구 분석을 위하여 교육기부 형태로 운영되는 ○대학교 창의 컴퓨터교실 (2012년 2기)에 지원한 초등학교 4~5학년 31명의 학생들을 대상으로 교육 프로그램 학습 전에 실시하였다.

(1) 프로그래밍(프로그래밍 학습)에 대한 경험

학습자들의 프로그래밍 또는 프로그래밍 학습에 대한 경험을 알아보기 위하여 설문을 실시하였고 이 결과를 <표 III-2>에 제시하였다.

<표 III-2> 프로그래밍 경험

내용	있다	없다
(2달 이상 지속적으로)프로그래밍(학습)에 대한 경험이 있나요?	11 (35%)	20 (65%)
프로그래밍에 사용한 언어는 무엇인가요?	스크래치(7), 코듀(3) 기타(1)	

학습자들의 절반 이상(65%)이 프로그래밍에 대한 경험이 없었고 경험이 있는 35%의 학생이 프로그래밍에 사용한 프로그래밍 언어 역시 스크래치와 코듀 등의 교육용 프로그래밍 언어로 나타났다. 학습자들의 프로그래밍 교육에 관심을 갖고 교육기부 강의에 신청한 학생들이 감안하면 실제 일반 학생들의 경우에는 프로그래밍 경험이 훨씬 낮을 것으로 판단된다. 따라서 프로그래밍 중심 STEAM 교육 프로그램 개발을 위해 프로그래밍 도구 선정 시 프로그래밍의 경험이 없는 학습자들을 감안해야 한다는 것을 주의해야 한다. 또한 프로그래밍을 통한 STEAM 교육을 위하여 프로그래밍 소양에 대한 교육이 별도의 교육도 필요하고 판단할 수 있다.

(2) 프로그래밍 학습에 대한 흥미와 자신감

학습자들이 프로그래밍 학습에 대해 갖고 있는 흥미와 자신감에 대한 설문을 실시하였고 그 결과를 <표 III-3>에 제시하였다.


<표 III-3> 프로그래밍 학습에 대한 흥미와 자신감

내용	평균	표준 편차	전혀 그렇지 않다 (1점)	그렇지 않다 (2점)	보통이다 (3점)	그렇다 (4점)	매우 그렇다 (5점)
프로그래밍 학습에 흥미가 있나요?	4.17	0.79	0	1	4	14	11
프로그래밍 학습에 자신감이 있나요?	3.8	0.89	0	1	12	9	8

프로그래밍 학습에 대한 흥미와 자신감에 대해 학습자들은 전체적으로 긍정적인 생각을 갖고 있었다. 다만, 프로그래밍 학습의 흥미에 비해 프로그래밍 학습의 자신감에 중립적인 의견이 많고 평균이 낮게 나타난 부분은 프로그래밍 흥미가 자신감으로 이어질 수 있도록 주제 선정이나 난이도 조절에 주의가 필요하다고 판단할 수 있다.

(3) 과학 학습에 대한 흥미와 자신감

학습자들이 과학 학습에 대해 갖고 있는 흥미와 자신감에 대한 설문을 실시하였고 그 결과를 <표 III-4>에 제시하였다.

 제주대학교 중앙도서관
<표 III-4> 과학 학습에 대한 흥미와 자신감

내용	평균	표준 편차	전혀 그렇지 않다 (1점)	그렇지 않다 (2점)	보통이다 (3점)	그렇다 (4점)	매우 그렇다 (5점)
과학 학습에 흥미가 있나요?	4.23	0.84	0	1	5	11	14
과학 학습에 자신감이 있나요?	4.1	0.71	0	0	6	15	9

학습자들은 과학 학습에 대해 흥미와 자신감에 측면에서 전체적으로 긍정적인 생각을 갖고 있었다. 학습에 흥미가 있는 이유로는 ‘실험이 재미있다’, ‘과학의 원리가 재미있다’, ‘유용한 사실을 알 수 있다’, ‘질문에 답해주고 새롭다’ 등으로 나타났고 흥미가 낮은 이유는 ‘재미가 없다’, ‘복잡하다’ 등으로 나타났다. 학생들이 과학의 원리를 학습하고 실험을 하는 것을 재미있고 있다는 점은 교육 프로그램을 설계 및 개발 시에 단순한 지식을 학습하는 것이 아니라 다양한 교과와의 융합을 통해 학습하는 것이 학습에 도움을 줄 수 있을 것이라 판단

된다.

(4) STEAM 교육에 대한 경험과 흥미

학습자들이 STEAM 교육에 대한 경험이 있는지 묻고 그 결과를 <표 III-5>에 제시하였다. STEAM 교육 경험이 있는 학습자에게 STEAM 교육에 대한 만족도를, STEAM 교육 경험이 없는 학습자에게 STEAM 교육에 대한 만족도를 추가로 설문하였고 그 결과를 <표 III-6>에 제시하였다.

<표 III-5> STEAM 교육 경험

내용	있다	없다
STEAM 교육에 대한 경험이 있나요?	8(26%)	23(74%)

<표 III-6> STEAM 교육에 대한 흥미

내용	평균	표준 편차	전혀 그렇지 않다 (1점)	그렇지 않다 (2점)	보통이다 (3점)	그렇다 (4점)	매우 그렇다 (5점)
(경험이 있는 학생(N=8)) STEAM 교육이 재미있었나요?	4.63	0.52	0	0	0	3	5
(경험이 없는 학생(N=23)) STEAM 교육에 흥미가 있나요?	4.19	0.93	0	2	1	9	9

STEAM 교육을 경험이 있는 학습자는 8명으로 전체의 26%에 해당하고 있었다. 아직 STEAM 교육이 활성화되지 않은 시점이기 때문에 STEAM 교육에 대한 경험이 적은 것으로 판단된다. STEAM 교육을 경험했던 학생들이 STEAM 교육은 재미있다고 느끼고 있었고 경험하지 못한 학생들도 흥미가 높게 나타난 것으로 볼 때 STEAM 교육이 학습자들에게 긍정적인 영향을 줄 것으로 판단된다.

5) 요구 분석 의사 결정

요구 분석 결과를 바탕으로 다음과 같은 요구를 추출 및 정리하였다.

첫째, 학습자들은 프로그래밍 경험이 전혀 없는 학생이 많고 경험이 있는 학생들도 초등학생의 수준과 흥미를 고려한 교육용 프로그래밍 언어를 사용한

것으로 나타났다. 프로그래밍 중심 STEAM 교육 프로그램을 설계하고 개발할 때 프로그래밍 도구로서 학생들의 수준에 적합한 교육용 프로그래밍 언어 선정과 프로그래밍 언어에 대한 학습이 필요함을 알 수 있었다. 또한 여러 명의 학습자가 직접 프로그래밍을 학습하는 것을 지원하기 위한 장치도 필요할 것으로 보인다.

둘째, 학생들이 과학 학습에 흥미가 있고 과학 학습 중에서도 실험이나 발견을 통한 학습을 선호하는 것으로 나타났다. 따라서 지식이나 개념 위주로 구성하기 보다는 학생들이 직접 조작하고 체험할 수 있도록 교육 프로그램을 구성하는 것이 필요하다.

셋째, 학생들이 STEAM 교육에 경험이 적고 초등학생이기 때문에 융합의 수준을 적절하게 조절하는 것이 필요하다. 주제를 선정하고 프로그램을 개발함에 있어 학습자의 흥미나 자신감을 떨어뜨리지 않도록 어려운 문제 보다는 학생들의 학습 태도나 흥미를 신장시킬 수 있는 방향으로 융합하는 것이 바람직하다.



2. STEAM 교육 프로그램의 설계

1) 교육과정 분석

본 연구에서 개발한 교육 프로그램은 초등학교 4~6학년의 학생들을 주된 학습자로 계획되었다. 이에 프로그래밍 중심 STEAM 교육 프로그램의 설계 단계에서 교육 프로그램의 주제 선정을 위하여 2009 개정 과학과 교육과정을 중심으로 초등학교 4~6학년의 내용을 분석하였다. <표 III-7>는 2009 개정 과학과 교육과정 중 4~6학년에 해당하는 내용이다(교육과학기술부, 2011a).

<표 III-7> 2009 개정 과학과 교육과정 내용(4~6학년)

영역 \ 학년	4	5	6
운동과 에너지	<ul style="list-style-type: none"> • 무게 • 열전달 	<ul style="list-style-type: none"> • 물체의 속력 • 전기 회로 	<ul style="list-style-type: none"> • 빛 • 에너지 • 자기장
물질	<ul style="list-style-type: none"> • 물의 상태 변화 	<ul style="list-style-type: none"> • 용해와 용액 	<ul style="list-style-type: none"> • 산과 염기 • 여러 가지 기체 • 연소와 소화
생명	<ul style="list-style-type: none"> • 식물의 한 살이 • 식물의 세계 	<ul style="list-style-type: none"> • 식물의 구조와 기능 • 작은 생물의 세계 • 우리의 몸 	<ul style="list-style-type: none"> • 생태계와 환경
지구와 우주	<ul style="list-style-type: none"> • 지층과 화석 • 화산과 지진 • 지표의 변화 	<ul style="list-style-type: none"> • 지구와 달 • 태양계와 별 	<ul style="list-style-type: none"> • 날씨의 변화 • 계절의 변화

국민 공통 기본 교육과정의 과학교과는 3학년부터 10학년까지 과학의 기본 개념을 이해하고, 과학적 탐구 능력과 태도를 함양하여 일상생활의 문제를 창의적이고 합리적으로 해결하는데 필요한 과학적 소양을 기르는 교과로서 2009 개정 과학과의 영역은 운동과 에너지, 물질, 생명, 지구와 우주의 네 가지 영역으로 구성되어 있다.

2009 개정 과학과 교육과정 중 본 연구에서 개발한 교육 프로그램의 주된 학습자로 선정한 초등학교 4~6학년의 과학과 교육과정 내용을 분석하여 각 학년 학기별 단원에 따라 프로그래밍 중심 STEAM 교육 프로그램에 적당한 내용 요소들을 추출하였다.

4학년 과학과 교육과정 중 프로그래밍 중심 STEAM 교육 프로그램의 주제로 가능성이 있는 내용 요소를 추출하여 <표 III-8>에 제시하였다. 4학년 과학과에서 지구와 우주 영역에 해당하는 단원은 지층과 화석의 생성 과정, 화산과 지진이 발생 원인, 지표의 변화 등 지구의 변화 과정을 살펴보는 내용으로 이루어져 있다. 이러한 변화 과정을 프로그래밍 하여 스토리텔링이나 시뮬레이션 하는 활동으로 재구성 할 수 있다. 생명 영역에서 식물의 한살이나 사는 곳에 따른 식물의 모습과 생활을 스토리텔링 하는 활동도 생각해 볼 수 있다.

<표 III-8> 2009 개정 과학과 4학년 교육과정 분석 결과

학년	학기	단원	추출 요소
4	1	무계재기	양팔저울 시뮬레이션 하기 나만의 저울 디자인하기
		지표의 변화	흙이 만들어지는 과정 스토리텔링하기 산사태 시뮬레이션 하기 강의 상류에서 하류로 가면서 지표가 달라지는 모습 표현하기 바다 속 탐사선 디자인하기
		식물의 한 살이	식물의 한살이를 프로그래밍하고 다른 친구의 것과 비교하기
		모습을 바꾸는 물	물의 상태 변화 표현하기 일상생활에서 물이 사용되는 모습 스토리텔링 하기 물의 순환 과정 스토리텔링하기
	2	식물의 세계	뿌리, 꽃, 열매의 생김새의 모습 표현하기 벌레잡이 식물 스토리텔링 하기 사는 곳에 따른 식물의 모습과 생활 스토리텔링하기
		지층과 화석	지층의 생성과정 표현하기 운석의 생성과정 표현하기 화석이 만들어 지는 과정을 표현하기
		열전달과 우리 생활	액체에서의 대류의 모습 시뮬레이션 하기
화산과 지진	화산 분출하는 모습 표현하기(화산 모형 만들기) 화산탐사 로봇 디자인하기 지진의 생성 원인 (습곡, 단층) 시뮬레이션 하기 진도계의 디자인하기 지진 발생 지역 표시하는 프로그램 만들기		

5학년 과학과 교육과정 중 프로그래밍 중심 STEAM 교육 프로그램의 주제로 가능성이 있는 내용 요소를 추출하여 <표 III-9>에 제시하였다. 5학년 과학과는 다른 학년에 비해 우주나 우리의 몸 등 직접 실험할 수 없는 내용이거나 전기나 작은 생물처럼 눈에 보이지 않는 내용이 많다. 프로그래밍을 통해 스토리텔링하거나 시뮬레이션하여 학습할 수 있도록 내용을 추출하였다.

<표 III-9> 2009 개정 과학과 5학년 교육과정 분석 결과

학년	학기	단원	추출 요소
5	1	지구와 달	달의 공전과 위상 변화 시뮬레이션 하기 낮과 밤이 생기는 까닭을 스토리텔링 하기
		전기회로	전기가 통하는 물체를 구분하는 프로그램 만들기 전기를 안전하게 이용하는 방법 스토리텔링 하기 나만의 전지 디자인하기
		식물의 구조와	식물의 물과 양분의 이동 경로 시뮬레이션 하기 광합성과 증산작용 시뮬레이션 하기

		기능	꽃과 열매의 구조와 하는 일 시뮬레이션 하기 식물의 씨가 번식하는 과정 스토리텔링하기
		작은 생물의 세계	우리 주변의 작은 생물들 표현하기 습지 속 작은 생물, 모기의 동작 표현하기
	2	우리 몸	소화과정 표현하기 위와 창자 속 여행을 스토리텔링하기 순환과 호흡 기관의 동작 시뮬레이션 하기
		용해와 용액	용해되는 장면 표현하기 다양한 용액의 특징 표현하기
		물체의 속력	속도를 알려주는 속도측정기 디자인하기
		태양계와 별	행성과 별자리 찾기 게임 만들기 태양계 시뮬레이션 하기 계절에 따른 별자리 관찰 프로그램 만들기 별자리에 대한 스토리텔링하기

6학년 과학과 교육과정 중 프로그래밍 중심 STEAM 교육 프로그램의 주제로 가능성이 있는 내용 요소를 추출하여 <표 III-10>에 제시하였다. 6학년 과학과의 내용은 학생들의 일상생활에서 쉽게 접할 수 있는 과학 현상에 대한 내용이 많은데 이러한 과학 현상을 프로그래밍을 통해 스토리텔링이나 시뮬레이션, 게임의 형태로 만들어보도록 내용 요소를 추출하였다.



<표 III-10> 2009 개정 과학과 6학년 교육과정 분석 결과

학년	학기	단원	추출 요소
6	1	빛	바늘구멍 사진기 시뮬레이션 하기 거울을 이용해 빛을 이동시키는 게임 만들기 공기와 물이 만나는 면에서의 빛의 굴절 시뮬레이션 하기 렌즈의 특성을 이용한 동작하는 게임 만들기
		산과 염기	산과 염기의 중화작용 시뮬레이션 하기 산도를 측정하는 도구 설계 또는 지시약 시뮬레이션 하기
		계절의 변화	하루 동안의 태양의 고도 변화 표현하기 계절에 따라 달라지는 모습 스토리텔링 하기 계절에 따른 태양의 남중고도의 변화 표현하기 공전에 따른 태양의 고도 변화 시뮬레이션 하기
		생태계와 환경	생태계 표현하기 생물의 상호작용(먹이사슬, 먹이피라미드, 기생과 공생) 스토리텔링 하기 생물이 환경에 적응하는 모습(보호색, 부리 등) 스토리텔링 하기 환경오염에 대한 스토리텔링 하기
		자기장	전자석 표현하기 전자석을 이용한 게임 만들기
	2	날씨의 변화	다양한 날씨를 스크래치로 표현하기 습도계 디자인하기 안개와 구름의 생성을 애니메이션으로 표현하기

		바람이 부는 이유 애니메이션으로 표현하기 일기도에 날씨의 변화 표현하기
	여러 가지 기체	열기구 디자인하기 여러 가지 기체의 성질 스토리텔링 하기 우리 생활 주변의 기체의 활용에 대한 스토리텔링 하기 지구온난화 예방을 위한 게임 또는 애니메이션 만들기
	에너지와 도구	에너지의 종류 카드 게임 만들기 태양을 이용하는 태양광/열 발전 표현하기 에너지 전환의 모습 스토리텔링 하기 에너지를 절약하는 모습 스토리텔링 하기 지레, 도르레, 경사면의 작동원리를 이용한 도구 디자인하기
	연소와 소화	촛불이 탈 때의 모습 시뮬레이션 하기 소화의 조건 활용한 게임 만들기

2) 학습준거 선정

교육과학기술부와 한국과학창의재단은 현장의 교사들이 쉽게 STEAM 교육 프로그램의 개발과 적용을 할 수 있는 도구 STEAM 교육 학습준거(틀)를 제시하였다(조향숙 외, 2012). 학습 준거는 STEAM 교육 프로그램 개발의 가이드라인이 되는 동시에 본 연구의 산출물로서 프로그램의 판단 기준으로 활용될 수 있다.

본 연구에서 개발한 교육 프로그램의 학습준거를 상황 제시, 창의적 설계, 감성적 체험으로 구분하여 <표 III-11>에 제시하였다.

<표 III-11> 학습준거 선정

구분	내용
상황 제시	환경, 교통 등 해결이 시급한 문제를 이야기 형태로 제시
창의적 설계	조사, 토론학습을 통한 과학적 탐구 및 해결방안 모색 프로그래밍을 통한 디지털 스토리 작성
감성적 체험	자신의 프로그램을 이용하여 디지털 스토리텔링

STEAM 교과 요소가 교육 프로그램에 기여할 내용 요소를 <표 III-12>과 같이 선정하였다.

<표 III-12> 내용 요소 선정

교과	내용 요소
과학	생활 속 과학적 사실, 문제의 발견 조사, 발표학습을 통한 문제해결방안 모색
기술	가상 로봇, 센서, 정보통신 기술을 활용한 프로그래밍 활동
공학	무대와 객체(스프라이트, 캐릭터) 디자인 자기만의 프로그램 제작
예술	스토리텔링을 위한 예술적이고 창의적 표현 심미적인 배경과 객체 표현
수학	프로그래밍 속 계산과 논리식 이해

과학 교과 요소는 본 연구에서 핵심 주제가 되는 부분으로 일상생활 속에서 접할 수 있는 다양한 과학적 사실과 문제를 선정하였다. 이러한 과학 주제는 조사, 발표, 토론학습을 통해 진행하였다.

기술 교과 요소로는 가상 로봇, 센서, 정보통신 기술 등 프로그래밍에 활용되는 최신 기술 분야를 선정하였다. 초등학교 수준에서 직접 다루지는 부분은 아니지만 프로그래밍 활동을 통해 자연스럽게 연결되도록 계획하였다.

공학 교과 요소로는 실제적인 프로그래밍 활동에서 교육용 프로그래밍 언어의 무대와 객체를 설계하고 자기만의 프로그램을 제작하는 것을 선정하였다.

스토리텔링을 위한 예술적이고 창의적인 표현, 그리고 이야기와 프로그램의 배과 객체들을 심미적으로 표현하는 활동을 예술 요소로 선정하였고 마지막으로 프로그래밍 속 좌표, 연산 등의 계산과 논리식을 이해하는 것을 수학 요소로 선정하였다.

융합인재교육은 실생활과 관련된 문제(real world problem)를 해결하는 경험을 제공한다. 즉, 학생들의 문제 해결 능력을 기르고, 빠르게 변하는 미래의 상황에 대처할 수 있는 적응력을 길러주게 된다(조향숙 외, 2012). 본 연구에서 개발한 STEAM 교육 프로그램에서는 우리 주변에서 해결이 시급한 문제, 즉 환경, 교통, 에너지 등 시급한 문제를 이야기 형태로 제시하여 학생들이 문제 상황을 인식하도록 제시하였다.

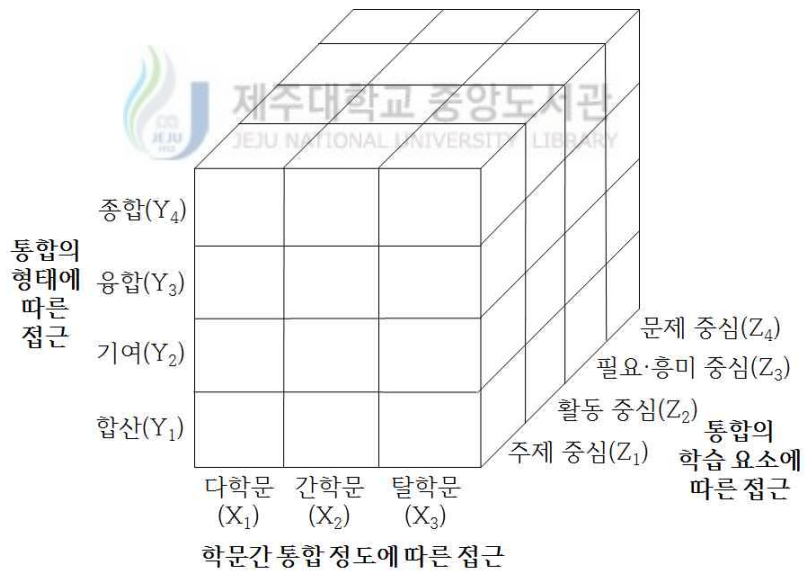
주어진 상황에서 문제를 해결하기 위해서 학생들은 조사, 토론 등의 방법을

통해 해결방안을 모색하고 이것을 프로그래밍을 통한 다양한 산출물(애니메이션, 게임, 스토리, 프리젠테이션 등)을 디자인한다.

창의적 설계를 통해 디자인한 프로그램을 작성하고 프로그래밍에 그치는 것이 아니라 이를 자신만의 이야기로 풀어내는 디지털 스토리텔링을 하게 된다. 이런 활동을 통해 학생들은 과학 학습과 프로그래밍, 스토리텔링에 대한 흥미를 갖게 되는 감성적 체험을 하게 된다.

3) 통합유형 선정

통합교육과정이나 교과목의 통합에 대한 관점은 학자에 따라 다양한데 본 연구에서는 배선아가 제시한 통합유형 분석 모형을 토대로 프로그래밍 중심의 STEAM 교육 프로그램에 적합한 통합 유형을 선정하였다(배건, 1997; 배선아, 2009))



[그림 III-2] 통합 유형 분석 모형

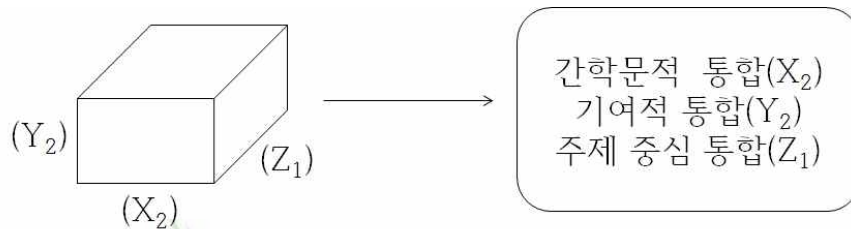
먼저, 통합의 형태에 따른 접근에서는 기여적 통합유형을 선택하였다. STEAM의 각 교과가 선정한 주제 학습에 서로 기여할 수 있는 요소에 의하여 학습에 도움을 줄 수 있도록 조직하여 연계하도록 하였다.

통합의 학습 요소에 따른 접근에서는 주제 중심 통합유형을 선택하였다. 과

학교과의 학습내용을 중심으로 주제를 선정하되 해당 주제에 대한 학습을 위해 기술, 공학, 예술, 수학의 학습요소를 융합하도록 하였다.

학문간 통합정도에 따른 접근에서는 간학문적 통합 유형을 선정하였다. 과학 분야에 밀접한 주제가 선정되었지만 프로그래밍, 기술, 공학, 수학 분야의 탐구 방법이나 모형을 적용하여 해결함으로써 STEAM에 해당하는 각 학문이 서로 공존하면서 새롭게 의미 있는 융합이 이루어지게 하였다.

이 연구에서 개발하고자 하는 교육 프로그램의 통합 유형을 [그림 III-3]에 제시하였다. X축의 간학문적 통합, Y축의 기여적 통합, Z축의 주제 중심 통합에 해당한다.



[그림 III-3] 통합 유형 선정
 JEJU NATIONAL UNIVERSITY LIBRARY

4) 수업모형 선정

STEAM 교육 프로그램을 지도하기 위한 수업 모형으로 일리노이주 주립대 CeMaST(Center for Mathematics, Science, and Technology)의 IMaST(Integrated Mathematics, Science, and Technology) 프로젝트에서 사용되었던 순환학습 모형을 수업 모형으로 사용하였다(Satchwell, R. E., & Loepp, F. L., 2002).

본 연구에서 사용한 학습 모형을 <표 III-13> 순환 학습 모형에 제시하였다.

<표 III-13> 순환 학습 모형

단계	활동
탐색하기	<ul style="list-style-type: none"> · 동기유발 · 이야기책, 생활 속의 과학적 문제 제시
아이디어 얻기	<ul style="list-style-type: none"> · 과학적 사실의 조사, 발표, 토의
아이디어 적용하기	<ul style="list-style-type: none"> · 주제에 대한 프로그래밍 · 디지털 스토리텔링 밑그림
아이디어 확장하기	<ul style="list-style-type: none"> · 자신만의 프로그램 만들기 · 창의적인 스토리 만들어 내기
평가하기	<ul style="list-style-type: none"> · 디지털 스토리텔링하기 · 상호평가, 학습 소감발표, 정리

탐색하기 단계에서는 과학 주제와 관련된 이야기책이나 동영상 등을 통해 문제를 제시하여 학생들이 학습할 문제를 파악하는 단계이다. STEAM 학습 준거에서 상황제시에 해당하는 부분이다.

아이디어 얻기는 과학주제에 대한 학습이 이루어지는 단계이다. 학생들은 발표, 조사, 토의하는 활동을 통해 주어진 문제를 학습한다. 이 과정에서 생성된 아이디어는 적용하기와 확장하기 단계에서 이루어지는 프로그래밍 활동과 디지털 스토리텔링 활동의 토대가 된다.

아이디어 적용하기 단계에서는 아이디어 얻기에서 학습한 과학 주제를 실제 프로그래밍으로 작성한다. 이 단계에서 교사는 예시 프로그램을 제시하여 학생들에게 자신만의 프로그램 구상에 도움을 준다. 또한 어떠한 이야기를 만들지 스토리텔링의 밑그림을 그린다.

아이디어 확장하기에서는 아이디어 얻기에서 학습한 내용과 아이디어 적용하기에서 제시한 교사의 예시 프로그램을 토대로 자신만의 프로그램으로 새롭게 창조하거나 업그레이드하는 활동을 한다. 학습자에 따라 수정이나 창조하는 능력에 차이가 있어 산출물에도 많은 차이가 있을 수 있지만 여기서 중요한 점은 자신만의 무엇인가를 새롭게 추가한다는 것이다.

마지막으로 평가하기 단계에서는 각자 만든 프로그램을 바탕으로 이야기를 구성하고 디지털 스토리텔링한다. 친구들과 발표하는 과정에서 상호평가 및 자

기 반성 등이 이루어진다.

3. STEAM 교육 프로그램의 개발

1) 학습 주제 및 내용 선정

STEAM 교육은 인류가 당면한 새로운 문제를 해결하고 유망한 분야에 대한 경쟁력을 확보하기 위해 시작되었다. Bybee(2010)는 PISA 2006의 scientific literacy를 바탕으로 STEM 교육의 주제를 제시하였다. 제시된 주제는 건강, 에너지 효율성, 천연자원, 환경(보호), 재난(방지), STEM의 첨단 분야 등이다 (Bybee, R. W., 2010). 이러한 문제들은 사회가 발전함에 따라 점점 복잡해지면서 어느 한 분야의 지식이나 노력으로만 해결하기가 힘든 문제이거나 전 세계가 경쟁력을 확보하기 위해 노력하는 분야이다. 이에 따라 본 연구에서는 환경보호, 우주 등 인류가 해결하려는 문제와 경쟁력을 확보하려는 과학 분야의 주제를 STEAM 교육 프로그램의 대주제로 선정하였다. 이는 과학 분야의 주제를 통해 융합하는 것이 STEM 또는 STEAM 교육이 추구하는 목적에 적합하다고 판단하였기 때문이다. 주제에 따른 소주제를 선정하기 위하여 2009 개정 초등학교 과학과 4-6학년 교육과정의 분석 내용을 바탕으로 교육 프로그램의 학습 주제로 선정하였다.

주제의 선정 기준은 교육용 프로그래밍 언어를 통한 프로그래밍 활동에 적합한지에 대한 프로그래밍 활동 적합성, 과학 중심의 주제 활동 안에 타 교과 의 학습내용을 융합하여 과목 간의 연계성을 높일 수 있는지에 대한 STEAM 교육 적합성, 교육과정에 포함된 내용과 연관이 되어있으면서도 학생들의 흥미를 유발할 수 있는지에 대한 학생 흥미 유발 여부이다.

본 연구에서는 센서 기반 스크래치 프로그래밍을 통한 STEAM 교육 프로그램, 코딩 프로그래밍을 통한 STEAM 교육 프로그램, 스크래치 프로그래밍 중심의 STEAM 교육 프로그램, 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램으로 나누어 총 4개의 교육 프로그램으

로 나누어 운영하였다.

초등학교 4-6학년의 과학과 교육과정 분석을 거쳐 15개의 예비주제를 선정하였고 컴퓨터과학 교육, 초등교육 전문가 집단과 협의하여, 코딩 프로그래밍을 통한 STEAM 교육 프로그램과 센스 기반 스크래치 프로그래밍에 대한 주제를 각 5개씩 선정하였다. 선정된 주제는 <표 III-14>에 제시하였다.

<표 III-14> 교육 프로그램 선정 주제(1차)

코딩 프로그래밍 중심 STEAM 교육 프로그램		교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램	
대주제	소주제	대주제	소주제
환경	지구가 뜨거워져요	환경	분리수거를 실천해요
생물	생태보호종을 지켜라	생물	반딧불이를 지켜라
에너지	에너지를 절약해요	에너지	에너지를 절약해요
우주	우주 생명의 흔적을 찾아라	로봇	미래의 로봇 설계하기
교통	교통 안전 도시 만들기	전기	과일을 이용한 악기 만들기

<표 III-15> 교육 프로그램 선정 주제(2차)

방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램		정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램	
대주제	소주제	대주제	소주제
생물	생물의 보호색	기체	여러 가지 기체의 쓰임
물질	물의 순환		
빛	빛	에너지	신재생 에너지
에너지	신재생 에너지	에너지	지구 온난화
우주	태양계 시뮬레이션		

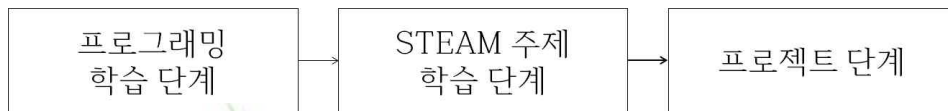
<표 III-14>에서 제시한 교육 프로그램 투입을 통해 얻은 피드백을 통해 수정 및 보완을 거쳐 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램의 주제를 선정하였다. 또한 6학년 정규 교육과정에서의 STEAM

교육 프로그램은 6학년 과학 교육과정에 준하여 해당 학기인 6학년 2학기의 내용을 선정하였다. 해당 프로그램의 주제를 <표 III-15>에 제시하였다.

2) 교육 프로그램 진행 단계에 따른 개발

본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램의 효율적인 실행을 위해 프로그램의 단계를 구분하여 개발하였다.

프로그래밍에 대한 기본 소양을 학습하는 단계, 선정된 과학 주제에 대하여 조사하고 토의한 내용을 바탕으로 이야기를 구성하고 이를 프로그래밍을 통해 구성하여 디지털 스토리텔링을 하는 STEAM 주제 학습 단계, 자신이 직접 선정한 과학 주제에 대한 프로그램을 디자인하고 작성하여 발표하는 프로젝트 단계이다. 교육 프로그램의 단계별 구성은 [그림 III-4]에 제시하였다.



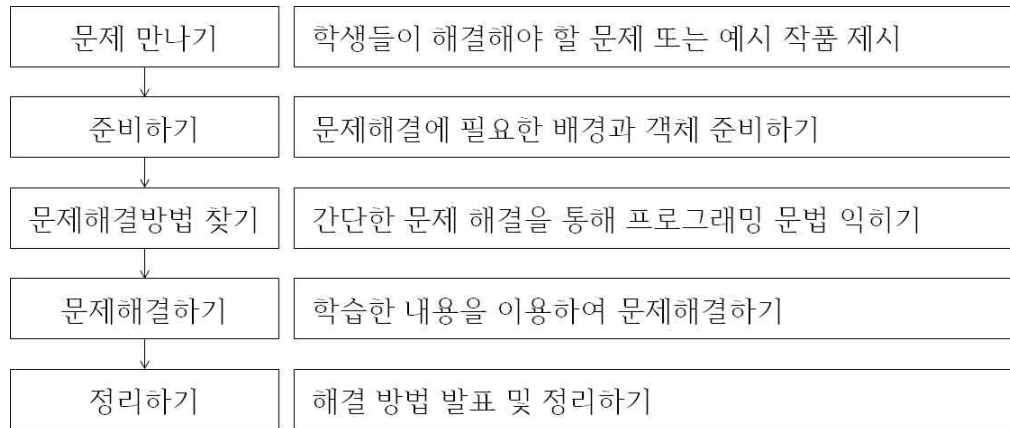
[그림 III-4] 교육 프로그램 단계별 구성

(1) 프로그래밍 학습 단계

요구 분석에서 대부분의 초등학생들이 프로그래밍 언어를 다뤄보지 않은 것으로 나타났다. 초등학교에서 정규 교과에서 프로그래밍을 다루는 시간이 없을 뿐만 아니라 방과후학교, 학원 등에서 이루어지는 컴퓨터교육 역시 컴퓨터 활용에 치중되었기 때문이다. STEAM 주제 학습 단계에서 학습한 과학 주제의 내용을 프로그램으로 작성하기 위해서 프로그래밍과 프로그래밍 언어 사용에 대한 이해가 필요하다고 판단하여 이에 알맞은 프로그래밍 학습 단계를 구성하였다.

프로그래밍 언어의 문법을 익히는 것만으로는 프로그래밍에 대한 흥미를 느끼기 어렵고 자기만의 프로그램을 작성하는데 어려움을 느낄 수 있기 때문에 단순히 교사의 코딩을 따라 하는 활동을 지양하고 학생 스스로 해결해야 할 문제를 제시하고 교사와 함께 학습한 예제 프로그램을 확장시켜 문제를 해결하는 방식으로 진행하였다. STEAM 주제 학습 단계에서 학습한 과학 주제에

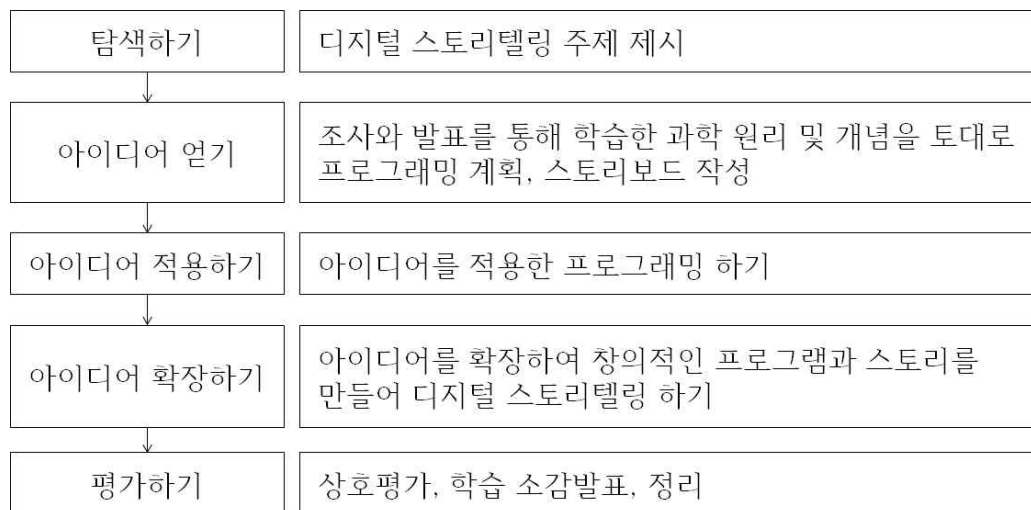
대해 자신의 프로그램을 직접 설계하여 프로그래밍하기 위해서는 문법 기능의 학습보다는 주어진 문제를 해결하기 위한 프로그래밍 언어의 활용이 중요하기 때문이다.



[그림 III-5] 프로그래밍 학습 단계

(2) STEAM 주제 학습 단계

STEAM 주제 학습 단계는 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램이 핵심이 되는 부분이다.



[그림 III-6] STEAM 주제 학습 단계

STEAM 주제 학습 단계는 [그림 III-6] 에 제시한 순환학습 모형의 단계에 따라 개발하였다.

가장 먼저 탐색하기 단계에서는 과학 교과를 중심으로 선정한 주제에 대하여 우리 주변에서 접할 수 있는 문제 상황을 제시한다. 문제 상황과 관련된 과학 원리나 개념을 찾는 단계로 학습자의 흥미 유발을 극대화하기 위하여 동영상 또는 이야기책을 이용하여 이루어진다.

아이디어 얻기 단계에서는 탐색하기 단계에서 찾은 문제와 그와 관련된 과학 원리나 개념을 학습하는 단계이다. 여기서 학습하는 과학 원리나 개념은 아이디어 적용하기와 확장하기 단계에서 이루어지는 프로그래밍 활동의 아이디어가 된다.

아이디어 적용하기 단계에서는 교사가 제시한 프로그램을 함께 작성해 본다. 자신만의 프로그래밍 전의 준비단계로 교사가 제시하는 것을 그대로 따라하는 것을 예방하기 위해 전체 코드를 제시하지 않고 학생들이 추측하여 활동할 수 있게 구성하였다.

아이디어 확장하기 단계에서는 자기만의 이야기를 구성하기 위해 적용하기 단계에서 작성한 프로그램을 업그레이드하거나 완전한 자신의 프로그램을 작성한다. 완성한 프로그램에 대해서 소개서와 프로그램 속에 숨은 이야기를 만드는 활동을 통해 학생들이 감성적인 체험이 이루어지도록 하였고 프로그램 학습 단계 중에서 창의적인 부분이 가장 강조되는 부분이다.

마지막으로 평가 단계에서는 작성한 프로그램과 이야기를 발표하고 이에 대한 상호평가를 실시한다. 좀 더 효율적인 프로그램의 공유 및 상호평가를 위해서 학습자간 아이디어를 공유할 수 있는 웹사이트를 제작하였다.

(3) 프로젝트 단계

학습자가 주제의 선정부터 완성까지 스스로 계획하고 결정하는 경험을 제공하고 그 과정을 통해 컴퓨팅 사고력과 융합적 사고력을 신장시키기 위해 개별 프로젝트 단계를 구성하였다. 프로젝트의 주제는 학생 스스로 선정하되 STEAM 주제 학습 단계에서 학습한 내용을 선정하거나 완전히 새로운 과학 주제를 선정하도록 하였다. 이 때 교수자의 역할은 학습자가 선정한 주제에 대한 프로그램을 완성할 수 있도록 과학적인 지식, 원리 및 프로그래밍 방법 등

에 대한 지원이다.

4. STEAM 교육 프로그램의 적용

컴퓨팅 사고력 신장을 위해 설계하고 개발한 프로그래밍 중심 STEAM 교육 프로그램을 초등학생들에게 적용하여 프로그램의 교육적 효과를 검증하였다.

본 연구에서는 실제 4개의 프로그래밍 중심 STEAM 교육 프로그램을 개발하였으며 프로그래밍 교육에 사용된 프로그래밍 언어에 따라 코듀 프로그래밍 중심의 STEAM 교육 프로그램과 스크래치 프로그래밍 중심의 STEAM 교육 프로그램으로 나누어 제시하였다. 교육 현장에서 가장 많이 보급되어 일반적으로 사용되고 있는 스크래치의 경우 투입 방식에 따라 각각 교육기부 프로그램, 방과후학교 프로그램, 정규교육과정 내의 프로그램으로 구분하였다.

1) 코듀 프로그래밍 중심 STEAM 교육 프로그램

코듀 프로그래밍 중심 STEAM 교육 프로그램은 초등학교 4~6학년의 학생들을 대상으로 하는 교육기부 프로그램을 적용하였다. 코듀는 최근 미국, 영국 등을 중심으로 프로그래밍 교육과 STEM(STEAM) 교육의 도구로 사용되고 있다. 교육용 프로그래밍 언어로서 코듀를 활용한 프로그래밍 교육과 STEAM 교육의 효과를 살펴보았다. 또한 가장 대표적인 프로그래밍 언어인 스크래치를 활용한 프로그래밍 중심 STEAM 교육 프로그램과의 비교를 통해 교육용 프로그래밍 언어에 따른 학습자의 창의력, 논리적 사고력, 과학과 관련된 정의적 영역에 어떠한 변화가 있는지를 분석하였다.

2) 스크래치 프로그래밍 중심 STEAM 교육 프로그램

스크래치 프로그래밍 중심 STEAM 교육 프로그램은 투입 방식과 수업전략에 따라 3개의 세부 프로그램으로 구분하였다.

(1) 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램

초등학교 4~6학년 학생을 대상으로 운영한 교육기부 프로그램으로 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램을 적용하였다. 센서는 스마트폰, 로봇 등에서도 사용되어 최근 많은 관심을 받고 있는 기술이다. 스크래치는 소리, 빛, 전기저항 등을 입력받는 센서보드를 지원하며 이를 활용한 프로그래밍을 통해 과학 교과와 자연스럽게 연결시킬 수 있다는 장점이 있다. 센서를 활용한 스크래치 프로그래밍 중심의 STEAM 교육 프로그램의 효과와 함께 센서의 활용 여부에 따라 STEAM 교육 프로그램의 효과에는 차이가 있는지를 분석하였다.

(2) 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램

두 번째는 초등학교 4~6학년 방과후학교 프로그램으로 스크래치 프로그래밍 중심의 STEAM 교육 프로그램을 적용하였다. 초등학교에는 컴퓨터 과학 또는 프로그래밍을 학습할 수 있는 교과가 없고 이에 따라 투입 시수를 확보할 수 없기 때문에 방과후학교 또는 동아리 활동을 통한 교육이 대안이 될 수 있다. 스크래치는 교육 현장에서 프로그래밍 교육을 위해 가장 많이 사용되고 있는 교육용 프로그래밍 언어이다. STEAM 교육과 프로그래밍 교육을 위한 도구로서 스크래치를 선정하고 이를 중심으로 구성된 프로그래밍 중심 STEAM 교육의 교육적 효과를 살펴보았다.

(3) 정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램

마지막으로 세 번째는 정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램으로 초등학교 6학년 정규 교육과정 내에서 스크래치 프로그래밍을 활용한 STEAM 교육 프로그램을 적용하였다. 스크래치 프로그래밍 중심의 STEAM 교육 프로그램이 초등학교의 정규 교육과정에서 어떻게 투입이 될 수 있는지, 정규 교육과정의 과학 등의 교과와 어떻게 융합될 수 있는지를 살펴보려고 하였다.

5. STEAM 교육 프로그램의 평가

본 연구에서는 프로그래밍 중심 STEAM 교육 프로그램을 평가하기 위해 교육 프로그램을 투입한 학습자의 변화를 비교·분석하여 교육적 효과를 검증하였다.

프로그래밍 중심의 STEAM 교육 프로그램의 교육적 효과를 평가하기 위해 평가의 관점을 두 부분으로 나누었다.

먼저 프로그래밍 교육에 대한 평가이다. 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램은 학습자의 컴퓨팅 사고력의 신장을 목표로 하였다. 주어진 문제를 고민하고 해결하는 과정에서 컴퓨팅 사고력이 얼마나 신장되었는지를 측정하기 위하여 컴퓨팅 사고력의 하위요소인 창의력과 논리적 사고력의 변화를 측정하였다.

두 번째는 STEAM 교육에서 요구하는 융합적 소양에 대한 평가이다. STEAM 교육은 21세기 학습자로 하여금 문제해결 과정에서 어느 하나의 학문에 구애받지 않고 문제해결을 할 수 있는 융합적 소양을 기르는 것을 목표로 한다. 학습에 대한 흥미는 이러한 융합적 소양을 위한 기본 요소이다. 특히 우리나라의 STEAM 교육의 도입은 과학·수학교과에 비해 높은 성취도에 비해 낮은 흥미도와 자신감 등 정의적 영역에 대한 문제를 해결하기 위한 고민에서 비롯되었다. 즉, 각각 개별적인 교과로써 학습했던 전통적인 방법을 벗어나 예술이나 인문학을 포함한 다양한 교과 요소의 융합적 창의적인 접근을 통해 과학, 수학에 대한 흥미를 높이고 이것이 학습자의 융합적 소양을 기르는데 도움이 된다는 것이다. 이에 본 연구에서는 융합적 소양이 기초가 되는 과학에 대한 흥미와 태도의 정의적 영역의 변화를 통해 STEAM 교육 프로그램의 효과를 검증하고자 하였다.

1) 검사 도구

(1) 창의력 검사 도구

창의성의 신장 여부를 확인하기 위한 검사 도구로 Torrance가 개발한 TTCT(Torrance Tests of Creative Thinking) 검사지 중 도형 A형을 사용하였다.

TTCT 창의력 검사에서는 '창의력'이란 창의적인 성취를 수행할 때 작용한

다고 생각되는 ‘일반화된 정신 능력들의 집합’이라고 정의한다. TTCT 검사는 그림 구성하기, 그림 완성하기, 쌍의 두직선- 선 그리기의 세 가지 활동으로 이루어져 있다. 이들 각기는 창의적 사고의 측면들 가운데서 각기 상이하면서도 독특한 측면을 다루고 있다.

수검자의 창의력을 판단하기 위해 TTCT 검사는 창의력의 하위요소를 유창성, 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 저항으로 구분하였다.

유창성은 주어진 자극을 유의미하게 사용하여 해석 가능한 반응으로 표현해 내는 능력을 말한다.

독창성은 수검자의 반응이 얼마나 드물게 일어나며 특별한 것인지를 판단한다.

제목의 추상성은 수검자가 종합하고 조직화할 줄 아는 사고과정과 관련되어 있다. 가장 높은 수준에서는 관련정보들의 핵심을 포착해 내고, 무엇이 중요한 것인지를 알고, 그리고 그림을 보다 깊게 풍부하게 볼 줄 아는 능력이 작용한다.

정교성은 자극도형에 대한 최소의 일차적인 반응은 하나의 단일 반응이고 내용을 자세하게 상상하고 설명할 수 있는 것은 정교성이라는 창의력의 함수라고 보는 것이다.

성급한 종결에 대한 저항은 독창적인 아이디어를 가능하게 하는 정신적 비약을 할 수 있을 만큼 충분히 긴 시간동안 마음을 열고 있으며 그래서 성급하게 반응을 폐쇄하고 종결시키는 것을 지연시킬 줄 아는 능력이다.

앞에서 다룬 5가지 척도들은 모두가 표준자료를 이용하여 해석하는 표준관련 척도이다. 창의적 강점 체크리스트는 절대 기준을 사용하는 준거관련 척도들로 정서적 표현, 이야기의 명료성, 운동 또는 행위, 제목의 표현성, 불완전도형들의 종합, 선들의 종합, 독특한 시각화, 내적인 시각화, 경계의 확대 또는 파괴, 유머, 심상의 풍부함, 심상의 다채로움, 환상의 13가지 항목으로 구성되어 있다. 이러한 창의적 강점을 교육과정과 수업방법을 개발하는데 사용할 수 있다.

창의력 점수는 유창성, 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 저항과 각 하위요소의 평균점수, 창의적 강점이 포함된 창의성 지수로 구분되

며 각각 표준점수와 백분위 점수를 사용할 수 있다. 본 연구에서는 표준점수를 사용하였다(Torrance, E. P. 김영채 역, 2010).

(2) 논리적 사고력 검사도구

본 연구에서는 학습자들의 논리적 사고력 변화를 측정하기 위하여 GALT(Group Assessment of Logical Thinking) 검사를 실시하였다.

Piaget의 인지발달 이론은 모든 학습자에게서 공통적으로 나타나는 어떤 독특한 양상을 인지구조라고 보며, 이러한 지적 구조는 학습자의 발달 과정에 따라 단계적으로 질적 분화하여 발달해 간다고 가정하는 이론이다. Piaget는 인지구조가 발달해 가는 단계를 감각운동기, 전조작기, 구체적 조작기, 형식적 조작기의 4단계로 구분하였다. Piaget의 연구에 따르면 형식적 조작기에 도달한 학생은 보존논리, 비례논리, 변인통제논리, 확률논리, 상관논리, 조합논리의 사고가 가능하다. 각 논리의 의미는 다음과 같다.

첫째, 보존논리는 어떤 수, 길이, 물질, 면적, 부피 등은 그 순서나 형태를 바꾸어 제시한다 하더라도 항상 변하지 않는다는 것을 아는 능력을 의미한다.

둘째, 비례논리는 어떤 두 비에 있어서 그 비의 값이 같다는 논리를 바탕으로 비례와 관련된 규칙을 이해하는 능력을 의미한다.

셋째, 변인통제논리는 상황의 모든 변인들을 인식하고 변인들의 역할에 관한 가설을 설정한 다음, 그 가설을 검증하기 위해 체계적으로 변인들을 통제하여 결론을 도출해 내는 능력을 의미한다.

넷째, 확률논리는 학습과정에 우연히 일어나는 사상 중에서 어떤 사상이 일어날 확률을 계산할 수 있는 능력을 의미한다.

다섯째, 상관논리는 두 개 변인 또는 현상사이에 어떤 상관적인 관계가 있을 것이라고 예상하는 능력을 의미한다.

마지막으로 조합논리는 문제를 해결해 나가는 과정에서 있을 수 있는 여러 가지 경우를 빠짐없이 중복되지 않게 셀 수 있는 능력을 의미한다.

Roadrangka 외(1983)는 학습자의 인지발달 수준을 측정하기 위한 방법으로 지필평가 형식의 논리적 사고력을 검사지를 개발하였다. 이 검사지는 6개의 하위논리를 포함하고 있는 총 21개 문항으로 개발되었으며 현재까지 학습자의 인지수준을 파악하기에 상당히 신뢰도와 타당도가 높고 사용이 편리한 검사도

구로 보고되었다. 뿐만 아니라 현재 많은 논문들의 학습자가 가지고 있는 오개념 조사나 인지발달 수준에 맞는 학습전략의 개발, 교과서 인지요구도 조사 등에 GALT 검사지를 적용 또는 인용하고 있고, 탐구 기능이나 인지 유형 등 다른 요인들과 인지수준과의 상관관계를 알아볼 때에도 GALT 검사가 사용되고 있다.

Roadrangka 외(1983)의 연구에서 GALT 검사지를 개발할 당시 시간이나 그 외의 제약이 있을 경우는 21문항 중에서 논리별로 균형을 맞추어 12문항으로 줄인 축소본을 사용하도록 권유하였다. 본 연구에서는 노정원(1997)의 연구에서 사용된 검사지에서 원본의 의미를 손상시키지 않으면서 학생들에게 친숙한 이름과 보충 설명들을 첨가하여 사용하였다. 축소본의 하위논리별 문항번호는 <표 III-16>과 같다.

<표 III-16> GALT 검사지 축소본의 문항 구성

하위 논리	보존 논리	비례 논리	변인통제 논리	확률 논리	상관 논리	조합 논리
문항번호	1, 2	3, 4	5, 6	7, 8	9, 10	11, 12

각 문항은 2개의 문항으로 구성되어 있으며 정답으로 생각하는 것과 그렇게 생각한 이유를 2문제 모두 맞추어야 정답으로 처리된다. GALT 축소본의 정답수에 따라 학습자의 인지수준을 분류하였는데 정답이 4개 이하이면 구체적 조작기, 5-7개까지는 과도기, 8개 이상이면 형식적 조작기로 분류할 수 있다.

(3) 과학과 관련된 정의적 영역 검사도구

STEAM 교육은 학생들이 갖고 있는 과학에 대한 흥미와 태도가 긍정적으로 변화하기를 기대한다. 본 연구에서 개발한 STEAM 교육 프로그램이 과학에 대한 흥미, 태도에 미치는 영향을 알아보기 위한 검사도구로 김효남 외(1998)가 개발한 과학에 관련된 정의적 특성 검사를 사용하였다. 이 검사지는 <표 III-17>에 제시한 것처럼 과학에 대한 정의적 특성을 과학에 대한 인식, 흥미, 과학적 태도의 3가지 범주로 구분하였으며 각 범주를 다시 하위 영역으로 구분하여 제시하였다. 하위 영역 별로 3개의 문항으로 구성되어 총 48개의 문항

이며 각 문항은 5단계의 Likert 척도로 구성되었다.(김효남 외, 1998).

<표 III-17> 과학과 관련된 정의적 영역 평가 세부요소

인식(C)	1.1 과학에 대한 인식(CS) 1.2 과학 교육에 대한 인식(CL) 1.3 과학과 관련된 직업에 대한 인식(CC) 1.4 과학·기술·사회의 상호 관련성에 대한 인식(CT)
흥미(I)	2.1 과학에 대한 흥미(IS) 2.2 과학 학습에 대한 흥미(IL) 2.3 과학과 관련된 활동에 대한 흥미(IA) 2.4 과학과 관련된 직업에 대한 흥미(IC) 2.5 과학 불안(IX)
태도(A)	3.1 호기심(AU) 3.2 개방성(AP) 3.3 비판성(AR) 3.4 협동성(AO) 3.5 자진성(AV) 3.6 끈기성(AE) 3.7 창의성(AC)



IV. STEAM 교육 프로그램 개발의 실제

1. 코딩 프로그래밍 중심의 STEAM 교육 프로그램

1) 요구분석

초등학교의 프로그래밍 교육에서 가장 많이 사용되고 있는 언어는 스크래치이다. 스크래치는 쉽고 장점이 많은 언어이지만 저학년의 학생들에게는 다소 어려울 수 있다. 스크래치가 스프라이트를 생성하고 모든 블록을 입력해야 동작하는 방식이라면 코딩은 각 캐릭터에 기본적인 동작이 입력되었기 때문에 실제로 캐릭터가 동작하는데 집중하여 프로그래밍을 학습할 수 있다는 장점이 있다. 게다가 로봇의 발전으로 최근 과학 교과에 등장하는 로봇 요소와 연결하여 학습하기 쉽다는 장점도 있다. 이에 본 연구에서는 스크래치와 비교하여 코딩이 갖는 교육적 가치와 융합교육 측면의 가능성을 살펴보기 위하여 코딩 프로그래밍 중심의 STEAM 교육 프로그램을 설계·개발하여 그 결과를 분석하였다.

2) 설계 및 개발

(1) 주제의 선정

STEAM 교육은 인류가 당면한 새로운 문제를 해결하고 유망한 분야에 대한 경쟁력을 확보하기 위해 시작되었다. Bybee(2010)는 PISA 2006의 scientific literacy를 바탕으로 STEM 교육의 주제를 제시하였다. 제시된 주제는 건강, 에너지 효율성, 천연자원, 환경(보호), 재난(방지), STEM의 첨단 분야 등이다. 본 연구에서는 STEAM 교육 프로그램의 주제를 선정하기 위해 초등학교 과학 4~6학년 교육과정을 분석하였다. 과학적인 주제로서 융합하는 것이 STEM 또는 STEAM 교육이 추구하는 목적에 적합하다고 판단하였기 때문이다.

과학 중심의 주제 활동 안에 타 교과의 학습내용을 융합하여 과목 간의 연

계성을 높일 수 있고 코딩으로 프로그래밍하여 디지털 스토리텔링하기에 적합한 내용을 주제로 선정하였다. 교육과정에 포함된 내용과 연관이 되어있으면서도 학생들의 흥미를 유발할 수 있는 소재를 추출하여 주제 중심의 학습이 될 수 있도록 구성하였으며 그 내용을 <표 IV-1>에 제시하였다.

<표 IV-1> 선정 주제

선정 주제	관련 내용 요소
환경	지구가 뜨거워져요(지구온난화)
생물	생태보호종을 지켜라(생물보호)
에너지	에너지를 절약해요(에너지 절약)
우주	우주 생명의 흔적을 찾아라(우주탐사)
교통	교통안전 도시 만들기(미래 교통도시)

(2) 세부 학습내용
 교육 프로그램은 <표 IV-2>처럼 크게 3개의 단계로 나누어서 진행되었다.

<표 IV-2> 교육 프로그램의 단계별 구성

단계(시수)	내용
스크래치 학습 (6)	<ul style="list-style-type: none"> · 프로그래밍이란?, 코딩 소개 · 나만의 세계 디자인하기 · 타이머와 스크어를 이용한 제어 · 객체 만들기 · 페이지를 이용한 제어 흐름 · 스토리텔링을 위한 다양한 기능
STEAM 주제 학습 (15)	<ul style="list-style-type: none"> · [환경] 지구가 뜨거워져요 · [생물] 생태보호종을 지켜라 · [에너지] 에너지를 절약해요 · [우주] 우주 생명의 흔적을 찾아라 · [교통] 교통안전 도시 만들기
프로젝트 (4)	<ul style="list-style-type: none"> · 개인 프로젝트 준비 · 프로젝트 발표

프로그래밍 경험이 없는 학생들이 코딩 프로그래밍을 통해 프로그래밍 언어를 접하고 문제해결을 하는 활동(프로그래밍 학습), 제시한 다섯 가지 STEAM 문제에 대해 조사·토론 등을 통해 과학적 사실을 탐구하고 이에 대한 프로그래밍을 해서 디지털 스토리텔링을 하는 단계(STEAM 주제 학습), 마지막으로 전 단계에서 학습한 문제 또는 자신이 관심 있는 주제에 대하여 프로그래밍을 재구성 또는 심화하여 자신만의 프로그램을 작성하는 단계(프로젝트)이다.

STEAM 주제 학습 단계는 일리노이주 주립대의 CeMaST(Center for Mathematics, Science, and Technology)의 IMaST(Integrated Mathematics, Science, and Technology) 프로젝트에서 사용되었던 순환학습 모형을 본 연구에 맞게 수정하여 진행하였다. 학생들은 발표, 조사, 토의하는 활동을 통해 주어진 문제를 학습한다. 그 과정에서 생성된 아이디어를 반영하여 이야기나 게임 등의 형태로 프로그래밍을 작성하고 이것을 다른 사람에게 디지털 스토리텔링하는 활동을 진행하였다.

초등학생들은 TV, 영화, 애니메이션을 통해 로봇을 접하고 로봇에 대한 흥미가 많은 편이다. 아이들은 숙제와 심부름을 대신해 주는 로봇, 같이 놀아 줄 수 있는 로봇을 상상한다. 실험 수업에서는 가상 로봇을 프로그래밍하는 언어인 코딩을 이용하여 프로그래밍 중심의 STEAM 교육을 실시하였다. 본 연구에서 사용한 STEAM 교육 프로그램의 학습준거와 교과 내용 요소를 <표 IV-3>, <표 IV-4>에 제시하였다.

<표 IV-3> 학습준거 선정

구분	내용
상황 제시	환경, 교통 등 해결이 시급한 문제를 이야기 형태로 제시
창의적 설계	조사, 토론학습을 통한 과학적 탐구 및 해결 방법 모색 프로그래밍을 통한 디지털 스토리 작성
감성적 체험	자신의 프로그램을 이용하여 디지털 스토리텔링

상황제시에서는 초등학생들 수준에 적합하고 쉽게 접할 수 있는 우리 사회의 다양한 문제들을 동영상이나 이야기책을 이용하여 제시하여 학습에 대한

흥미를 유발하고 과학적 사실과 문제를 유추해 낼 수 있도록 하였다. 최종 단계에서 이루어지는 학생들의 디지털 스토리텔링과 자연스럽게 이어질 수 있도록 이야기 형식으로 제시하였다.

창의적 설계에서는 제시한 문제를 바탕으로 과학적 사실과 해결방안에 대한 조사, 탐구학습을 통해 주제에 대한 심도있는 학습이 이루어지도록 하였다. 또한 그 내용을 바탕으로 객체와 환경을 디자인하여 코딩 프로그래밍을 통해 디지털 스토리텔링을 위한 프로그램을 작성하였다. 이 과정에서 STEAM 교과별 요소의 학습이 이루어진다.

감성적 체험에서는 각 학습자가 만든 프로그램을 통해 디지털 스토리텔링을 진행하였다. 디지털 스토리텔링을 통해 당면한 문제의 심각성을 알고 그에 대한 과학적 사실을 실제 생활과 연관되어 이해할 수 있으며 해결을 위한 노력과 실천의지를 다질 수 있는 기회를 갖도록 하였다.

<표 IV-4> 내용 요소 선정

교과	내용 요소
과학	생활 속 과학적 사실, 문제의 발견 조사, 발표학습을 통한 해결방안 모색
기술	가상 로봇 기술
공학	세계(World) 디자인, 프로그래밍 활동
예술	스토리텔링을 위한 예술적이고 창의적 표현 심미적인 세계(World) 디자인
수학	프로그래밍 속 계산과 논리식 이해

3) 적용 및 효과분석

(1) 연구 가설

① 가설 1

- 연구가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 없다.
- 대립가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한

학습자의 창의력에는 차이가 있다.

② 가설 2

- 연구가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 없다.
- 대립가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 있다.

③ 가설 3

- 연구가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 없다.
- 대립가설: 코딩 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 있다.

(2) 연구 대상

실험집단은 J대학교 창의컴퓨터교실에 지원한 14명의 학생을 선정하였다. 창의컴퓨터교실은 지역 내 4, 5학년 초등학생 중 프로그래밍 강좌 수업을 희망하는 학생들을 모집하여 교육기부 형태로 운영되었다.

비교집단 역시 제주지역 J대학교의 초등창의컴퓨터 교실에 지원한 4~5학년 학생 13명을 선정하였다. 비교집단에는 코딩 프로그래밍을 통한 게임 개발을 학습주제로 수업을 진행하였다. 연구 대상에 대한 구체적인 사항을 <표 IV-5>에 제시하였다.

<표 IV-5> 연구 대상

구분	실험집단(명)			비교집단(명)		
	남	여	소계	남	여	소계
4학년	6	5	11	4	4	8
5학년	3	0	3	3	2	5
소계	9	5	14	7	7	13

실험집단의 참여자 중 1명은 GALT와 TTCT의 검사를 실시하지 못하여 이를 결측값으로 처리하여 통계를 실시하였다.

(3) 연구 절차

본 연구에서 개발한 코뮤티를 이용한 프로그래밍 중심의 STEAM 교육 프로그램을 투입하기 전 실험집단과 비교집단의 창의력과 논리적 사고력을 비교하기 위한 사전검사를 실시하였다. 실험집단에 대해서는 과학과 관련된 정의적 영역 검사를 추가로 실시하였다.

교육 프로그램은 6일간 총 24차시의 집합 강의 및 실습으로 구성하였고 교육 프로그램이 시작되는 첫 날에는 사전검사와 오리엔테이션 마지막 날에는 프로젝트 발표 및 사후검사를 실시하였다. 실험집단에 투입한 교육 프로그램의 투입 일정을 <표 IV-6>에 제시하였다.

<표 IV-6> 프로그램 일정

일자	프로그램 일정
1	사전검사, 오리엔테이션(3차시)
2	코뮤티 소양 학습(6차시)
3	1, 2주제 학습(각 3차시)
4	3, 4주제 학습(각 3차시)
5	5주제 학습(3차시), 개인별 프로젝트 준비(3차시)
6	프로젝트 발표 및 사후검사(3차시)

<표 IV-7> 실험 설계

	사전검사	처치	사후검사
실험집단	O ₁ , O ₃	X ₁	O ₂ , O ₄
비교집단	O ₁	X ₂	O ₂

- O₁ : 논리적 사고력, 창의력 사전검사
- O₂ : 논리적 사고력, 창의력 사후검사
- O₃ : 과학과 관련된 정의적 영역 사전검사
- O₄ : 과학과 관련된 정의적 영역 사후검사
- X₁ : 코뮤티 프로그래밍 중심 STEAM 교육 프로그램
- X₂ : 코뮤티 프로그래밍을 통한 게임 개발 프로그램

비교집단에는 7주간 총 21차시의 집합 강의 및 실습으로 구성된 코뮤티를 활

용한 게임 개발을 주제로 강의를 진행하였으며 마찬가지로 첫 주차와 마지막 주차에는 창의력과 논리적 사고력에 대한 사전검사와 사후검사를 실시하였다. 본 실험의 연구 설계를 도식화 하여 <표 IV-7>에 제시하였다.

(4) 연구 결과

① 창의력 변화

- 정규성 검정

창의력 검사 결과 비교 전 두 집단의 창의력 사전 점수가 정규성을 확보하고 있는지 확인할 필요가 있었다. 먼저 실험·비교집단의 사전 창의력 검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 <표 IV-8>, <표 IV-9>에 그 결과를 제시하였다.

<표 IV-8> 실험집단 창의력 검사 정규성 검정

하위 요소	기술통계(N=13)				통계량	유의도
	M	SD	Max	Min		
유창성	98.154	15.513	125.0	71.0	.966	.847
독창성	101.000	17.272	130.0	70.0	.955	.672
제목의 추상성	108.154	19.065	67.0	130.0	.909	.180
정교성	135.154	22.479	150.0	92.0	.667	.000**
성급한 종결에 대한 저항	104.385	10.153	112.0	73.0	.655	.000**
창의력 평균	109.308	13.263	129.0	76.0	.886	.086
창의력 지수	123.769	16.971	148.0	82.0	.879	.068

**p<.01

정규성 검사 결과 실험집단의 창의력 검사 결과에서 정교성과 성급한 종결에 대한 저항 영역의 유의도가 .000으로 나타나 귀무가설을 기각하여 정규성이 만족되지 않았다. 나머지 영역에서는 유의도가 유의수준인 .05보다 크게 나타나 귀무가설이 채택되어 정교분포임이 확인되었다.

<표 IV-9> 비교집단 창의력 검사 정규성 검정

하위 요소	기술통계(N=13)				통계량	유의도
	M	SD	Max	Min		
유창성	90.308	14.279	113	65	.962	.779
독창성	84.462	17.295	127	54	.927	.307
제목의 추상성	109.231	36.431	154	40	.934	.380
정교성	108.846	10.197	128	91	.966	.840
성급한 종결에 대한 저항	90.308	11.636	110	67	.963	.793
창의력 평균	96.154	13.397	113	67	.936	.402
창의력 지수	108.846	15.889	133	73	.953	.648

비교집단은 모두 정규성을 확보한 것으로 나타났다. 실험집단과 비교집단의 정규성 확보 여부에 따라 집단 간·집단 내 비교 시 모수적 통계방법과 비모수적 통계방법을 혼합하여 검사결과를 비교하였다.

• 집단 간 비교

본 연구에서 개발한 교육 프로그램이 학습자의 창의력에 미친 영향을 살펴 보기 위해 실험집단과 비교집단의 창의력 사전·사후검사 결과를 비교하였다.

먼저 실험 투입 전 두 집단이 동질집단임을 확인하기 위해 사전검사를 비교하였는데 정규성을 확보한 영역에 대해서는 독립표본 t검정을 실시하여 <표 IV-10>에 제시하였고 제목의 추상성과 정교성의 경우 실험집단의 사전검사 결과에서 정규성을 확보하지 못하여 Mann-Whitney U검정을 실시하여 <표 IV-11>에 제시하였다.

두 집단의 창의력 사전검사를 비교한 결과 유창성과 제목의 독창성에서는 두 집단이 동질 집단으로 나타났지만 나머지 영역에 대하여서는 이질집단인 것으로 나타났다. 모집으로 이루어진 집단이기 때문에 실험집단의 평균이 일반적인 집단에 비해 창의력 수준이 높은 것이 원인인 것으로 판단된다.

<표 IV-10> 창의력 사전검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
유창성	실험	13	98.154	15.513	1.342	.192
	비교	13	90.308	14.279		
독창성	실험	13	101.000	17.272	2.440	.022*
	비교	13	84.462	17.295		
제목의 추상성	실험	13	108.154	19.065	-.094	.926
	비교	13	109.231	36.431		
창의력 평균	실험	13	109.308	13.263	2.516	.019*
	비교	13	96.154	13.397		
창의력 지수	실험	13	123.769	16.971	2.314	.030*
	비교	13	108.846	15.889		

*p<.05

<표 IV-11> 창의력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
정교성	실험	13	135.154	22.479	17.46	33.000	.008**
	비교	13	108.846	10.197	9.54		
성급한 종결에 대한 저항	실험	13	104.385	10.153	18.23	23.000	.002**
	비교	13	90.308	11.636	8.77		

**p<.01

프로그램 투입 후 사후검사를 비교하기 위해서 정규성을 확보하고 사전검사 에서 두 집단이 동질집단임이 확인된 유창성과 제목의 추상성에는 독립표본 t 검정을 이용하여 비교하였고 그 결과를 <표 IV-12>에 제시하였다.

<표 IV-12> 창의력 사후검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
유창성	실험	13	119.385	17.086	2.308	.030*
	비교	13	104.154	16.552		
제목의 추상성	실험	13	108.846	20.780	.006	.995
	비교	13	108.769	40.225		

*p<.05

창의력 사후검사 중 유창성과 제목의 추상성의 결과를 비교한 결과 실험집

단이 유창성이 비교집단에 비해 유의미하게 상승한 것으로 나타났다.

정규성을 확보하였으나 사전검사 분석 결과 이미 이질집단으로 판단된 독창성, 창의력 평균, 창의력 지수 하위요소에 대해서는 사전점수를 통제하기 위해 공분산 분석을 실시하였다.

실험집단과 비교집단의 독창성, 창의력 평균, 창의력 지수에 대한 사전평가, 사후평가, 사전평가를 통제한 교정 점수의 평균을 <표 IV-13>에 제시하였다.

<표 IV-13> 집단에 따른 독창성, 창의력 평균, 창의력 지수에 대한 서술통계

하위요소	구분		실험집단	비교집단	합계
독창성	사전	M	101.000	84.462	92.731
		SD	17.272	17.295	18.918
	사후	M	121.615	99.308	110.462
		SD	16.761	20.176	21.439
	교정 사후	M	115.655	105.268	
		SD	4.127	4.127	
	N		13	13	26
창의력 평균	사전	M	109.308	96.154	102.731
		SD	13.263	13.397	14.682
	사후	M	118.2308	101.077	109.654
		SD	8.36813	10.7196	12.856
	교정 사후	M	114.886	104.422	
		SD	2.055	2.055	
	N		13	13	26
창의력 지수	사전	M	123.769	108.846	116.308
		SD	16.971	15.889	17.814
	사후	M	134.6154	114.308	124.462
		SD	10.96615	13.332	15.820
	교정 사후	M	130.471	118.452	
		SD	2.420	2.420	
	N		13	13	26

독창성의 교정평균은 실험집단이 115.082, 비교집단이 105.268이고, 창의력 평균의 교정평균은 실험집단이 114.886, 비교집단이 104.422로 나타났으며 창의력지수의 교정평균은 실험집단이 130.471, 비교집단이 118.452로 나타났다. 교정된 사후 창의력 하위요소별 결과가 집단에 따라 어떠한 차이가 있는지에 대한 공분산분석 결과는 <표 IV-14>와 같다.

<표 IV-14> 집단에 따른 교정된 하위요소에 대한 공분산분석 결과

하위요소	분산원	제곱합	자유도	평균제곱	F	p
독창성	공분산(사전독창성)	6396.779	1	6396.779	32.466	.000
	집단	562.013	1	562.013	2.852	.105
	오차	4531.670	23	197.029		
	합계	328736.000	26			
창의력 평균	공분산(사전창의력평균)	2452.614	1	2452.614	50.542	.000
	집단	563.163	1	563.163	11.605	.002**
	오차	1116.107	23	48.526		
	합계	316755.000	26			
창의력 지수	공분산(사전창의력지수)	3913.842	1	3913.842	57.156	.000
	집단	767.657	1	767.657	11.210	.003**
	오차	1574.963	23	68.477		
	합계	409014.000	26			

사전 유창성, 사전 창의력 평균, 사전 창의력 지수의 F통계값은 각각, 32.466, 50.542, 57.156이고 유의확률은 3요소 모두 .000으로 사전 창의력 평가 결과가 사후 점수에 유의한 영향을 미치고 있는 것으로 나타났다. 사전 결과가 사후 결과에 미치는 영향을 통제된 후의 사후평균의 F값은 각각 .2852, .11.605, .11.210이고 유의 확률이 .105, .002, .003으로 두 집단의 교정된 창의력 평균과 창의력 지수에 유의한 차이가 있는 것으로 나타났다.

창의력에 대한 두 집단 간 비교 결과 비교집단에 비해 코듀 프로그래밍 중심의 STEAM 교육 프로그램을 학습한 실험집단의 유창성, 창의력 평균, 창의력 지수에 유의미한 영향을 미침을 알 수 있었다.

• 집단 내 비교

각 집단 내에서는 어떠한 변화가 있었는지를 확인하기 위해 각 집단별로 사전·사후 결과를 비교해 보았다. 실험집단의 경우 정규성을 확보하지 못한 정규성과 성급한 종결에 대한 저항을 제외한 영역에 대해 대응표본 t검정을 실시하였으며 그 결과는 <표 IV-15>와 같다.

<표 IV-15> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	98.154	15.513	-4.948	.000**
	사후	119.385	17.086		
독창성	사전	101.000	17.272	-4.457	.001**
	사후	121.615	16.761		
제목의 추상성	사전	108.154	19.065	-.143	.888
	사후	108.846	20.780		
창의력 평균	사전	109.308	13.263	-2.942	.012*
	사후	118.231	8.368		
창의력 지수	사전	123.769	16.971	-3.133	.009**
	사후	134.615	10.966		

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

유창성에서 t 통계값이 -4.948, 유의확률이 .000, 독창성에서 t 통계값이 -4.457, 유의확률이 .001, 창의력 평균의 t 통계값이 -2.942, 유의확률이 .012, 창의력 지수에서 t 통계값이 -3.133, 유의확률이 .009로 나타나 각각 유의수준 .05에서 사전점수에 비해 사후점수에서 유의미한 상승이 있는 것으로 나타났다.

정규분포를 따르지 않는 정교성과 성급한 종결에 대한 저항에 대해서는 Wilcoxon 부호순위 검정을 실시하였고 그 결과를 <표 IV-16>에 제시하였다.

<표 IV-16> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
정교성	실험	14	사전	135.154	22.479	-1.173	.241
			사후	140.846	13.741		
성급한 종결에 대한 저항	실험	14	사전	104.385	10.153	-1.876	.061
			사후	108.231	8.308		

검증 결과 정교성과 성급한 종결에 대한 저항에 대해서는 사전, 사후검사 간에 유의미한 차이가 없는 것으로 나타났다.

비교집단에 대해서 창의력 사전·사후검사에 대해 대응표본 t검정을 통해 프로그래밍 투입 간에 유의미한 차이가 발생하였는지를 확인하였다.

<표 IV-17> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	90.308	14.279	-5.486	.000**
	사후	104.154	16.552		
독창성	사전	84.462	17.295	-4.428	.001**
	사후	99.308	20.176		
제목의 추상성	사전	109.231	36.431	.090	.930
	사후	108.769	40.225		
정교성	사전	108.846	10.197	-.723	.484
	사후	110.462	10.268		
성급한 종결에 대한 저항	사전	90.308	11.636	.944	.364
	사후	84.769	18.244		
창의력 평균	사전	96.154	13.397	-2.306	.040*
	사후	101.077	10.720		
창의력 지수	사전	108.846	15.889	-2.171	.051
	사후	114.308	13.332		

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

<표 IV-17>에서 제시하였듯이 비교집단의 유창성, 독창성, 창의력 평균이 유의미하게 상승하였으며 **제주특별자치도교육청** 코딩을 통한 게임 프로그래밍 활동 역시 학습자의 창의력 신장에 도움을 주는 것으로 나타났다.

하지만 집단 간, 집단 내 결과를 비교하였을 때 코딩을 통합 게임 프로그래밍을 학습한 비교집단에 비해 코딩 프로그래밍 중심의 STEAM 교육 프로그램이 학습자의 창의력 신장에 더 효과적이라는 결과를 도출할 수 있었다.

② 논리적 사고력 변화

• 정규성 검정

논리적 사고력 검사지 역시 각 집단별 사전검사 결과에 대해 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-18>, <표 IV-19>에 제시하였다.

정규성 검정의 결과 두 집단 모두 논리적 사고력 합계 점수에서는 정규분포를 따르고 있지만 각 하위요소에 대해서는 정규분포를 나타내지 않았다. 따라서 정규성 검증 결과에 따라 집단 간, 집단 내 비교 방법을 달리하여 비교하였다.

<표 IV-18> 실험집단 논리적 사고력 검사 정규성 검정

하위 요소	기술통계(N=13)				통계량	유의도
	M	SD	Max	Min		
보존	1.231	0.599	2	0	.766	.003**
비례	0.385	0.650	2	0	.650	.000**
변인통제	0.846	0.899	2	0	.766	.003**
확률	0.308	0.630	2	0	.567	.000**
상관	0.077	0.277	1	0	.311	.000**
조합	1.769	0.439	2	1	.533	.000**
합계	4.615	1.557	7	2	.924	.286

**p<.01

<표 IV-19> 비교집단 논리적 사고력 검사 정규성 검정

하위 요소	기술통계(N=13)				통계량	유의도
	M	SD	Max	Min		
보존	1.077	0.760	2	0	.825	.014*
비례	0.077	0.277	1	0	.311	.000**
변인통제	0.615	0.650	2	0	.772	.003**
확률	0.077	0.277	1	0	.311	.000**
상관	0.231	0.439	1	0	.533	.000**
조합	1.231	0.832	2	0	.785	.005**
합계	3.308	1.932	7	0	.970	.889

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

• 집단 간 비교

본 연구에서 개발한 교육 프로그램이 학습자의 창의력 논리적 사고력에 미친 영향을 살펴보기 위해 실험집단과 비교집단의 논리적 사고력 사전·사후검사 결과를 비교하였다.

먼저 실험 투입 전 두 집단이 동질집단임을 확인하기 위해 정규성을 확보한 논리적 사고력 합계의 비교에는 독립표본 t검정을 실시하였고 나머지 하위 영역에 대해서는 Mann-Whitney U검정을 사용하여 사전검사를 비교하였다. 두 집단 간 사전검사 결과를 비교하여 <표 IV-20>, <표 IV-21>에 제시하였다. 비교 결과 두 집단의 논리적 사고력 모든 하위요소 및 합계에서 동질집단인 것으로 나타났다.

<표 IV-20> 논리적 사고력 사전검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
합계	실험	13	4.615	1.557	1.901	.069
	비교	13	3.308	1.932		

<표 IV-21> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	13	1.231	0.599	14.19	75.500	.609
	비교	13	1.077	0.760	12.81		
비례	실험	13	0.385	0.650	15.04	64.500	.135
	비교	13	0.077	0.277	11.96		
변인통제	실험	13	0.846	0.899	14.31	74.000	.560
	비교	13	0.615	0.650	12.69		
확률	실험	13	0.308	0.630	14.54	71.000	.270
	비교	13	0.077	0.277	12.46		
상관	실험	13	0.077	0.277	12.50	71.500	.286
	비교	13	0.231	0.439	14.50		
조합	실험	13	1.769	0.439	15.85	54.000	.070
	비교	13	1.231	0.832	11.15		

프로그램 투입 후 집단 간에 차이가 발생했는지를 알아보기 위해 사후검사를 비교하였고 그 결과를 <표 IV-22>, <표 IV-23>에 제시하였다.

<표 IV-22> 논리적 사고력 사후검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
합계	실험	13	5.000	1.472	1.620	.118
	비교	13	3.923	1.891		

<표 IV-23> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	13	1.231	0.599	13.50	84.500	1.000
	비교	13	1.231	0.599	13.50		
비례	실험	13	0.385	0.650	14.42	72.500	.403
	비교	13	0.231	0.599	12.58		
변인통제	실험	13	1.077	0.954	14.46	72.000	.497
	비교	13	0.846	0.689	12.54		
확률	실험	13	0.462	0.660	15.35	60.500	.094
	비교	13	0.154	0.555	11.65		
상관	실험	13	0.000	0.000	13.00	78.000	.317
	비교	13	0.077	0.277	14.00		
조합	실험	13	1.846	0.376	16.08	51.000	.039*
	비교	13	1.385	0.650	10.92		

*p<.05

실험집단과 비교집단의 사후검사를 비교한 결과 논리적 사고력 하위요소 중 조합논리에 대하여 Mann-Whitney의 U 값이 51.000으로 나타났으며 유의 확률이 .039로 나타나 유의수준 .05에서 통계적으로 유의미한 결과를 얻을 수 있었으며 실험집단에 실시한 교육 프로그램이 학습자의 조합논리 향상에 긍정적인 영향을 줄 수 있었다.

• 집단 내 비교

집단 내 비교를 통해 본 연구에서 개발한 교육 프로그램이 학습자의 논리적 사고력에 미친 영향에 대해 확인하고자 하였다. 두 집단 모두 정규분포를 따르는 논리적 사고력 합계에 대해서는 대응표본 t검정을 실시하였고 나머지 논리적 사고력의 하위 요소에 대해서는 Wilcoxon 부호순위 검정을 사용하여 분석하였다.

먼저 실험집단의 사전·사후검사 비교 결과를 <표 IV-24>, <표 IV-25>에 제시하였다.

<표 IV-24> 실험집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 T 검정)

하위 요소	시기	N	M	SD	t	유의도
합계	사전	13	4.615	1.557	-2.739	.018*
	사후	13	5.000	1.472		

*p<.05

<표 IV-25> 실험집단 논리적 사고력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
보존	13	사전	1.231	0.599	.000	1.000
		사후	1.231	0.599		
비례	13	사전	0.385	0.650	.000	1.000
		사후	0.385	0.650		
변인통제	13	사전	0.846	0.899	-1.732	.083
		사후	1.077	0.954		
확률	13	사전	0.308	0.630	-1.414	.157
		사후	0.462	0.660		
상관	13	사전	0.077	0.277	-1.000	.317
		사후	0.000	0.000		
조합	13	사전	1.769	0.439	-.577	.564
		사후	1.846	0.376		

실험집단의 논리적 사고력 사전·사후검사 분석 결과 논리적 사고력 합계의 t 통계값은 -2.739이고 유의확률이 .018로 나타나 유의수준인 .05보다 작으므로 사전검사와 사후검사가 통계적으로 유의하며 교육 프로그램이 학생들의 논리적 사고력을 향상시킨 것으로 나타났다.

비교집단 역시 정규분포를 따르는 논리적 사고력 합계는 사전·사후검사의 결과를 대응표본 t검정으로 분석하여 <표 IV-26>에 제시하였다. 정규분포를 따르지 않는 나머지 논리적 사고력 하위요소에 대해서는 Wilcoxon 부호순위 검정으로 분석하였고 그 결과를 <표 IV-27>에 제시하였다.

<표 IV-26> 비교집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정)

하위 요소	시기	N	M	SD	t	유의도
합계	사전	13	3.308	1.932	-2.125	.055
	사후	13	3.923	1.891		

<표 IV-27> 비교집단의 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
보존	13	사전	1.077	0.760	-.632	.527
		사후	1.231	0.599		
비례	13	사전	0.077	0.277	-1.000	.317
		사후	0.231	0.599		
변인통제	13	사전	0.615	0.650	-1.134	.257
		사후	0.846	0.689		
확률	13	사전	0.077	0.277	-1.000	.317
		사후	0.154	0.555		
상관	13	사전	0.231	0.439	-1.414	.157
		사후	0.077	0.277		
조합	13	사전	1.231	0.832	-1.000	.317
		사후	1.385	0.650		

비교집단의 논리적 사고력 사전·사후검사 분석 결과 상관논리를 제외한 모든 영역에서 평균점수 상승은 있었지만 논리적 사고력의 모든 영역에서 통계적으로 유의미한 차이가 나타나지 않았다.

③ 과학과 관련된 정의적 영역

실험집단의 과학과 관련된 정의적 영역에 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-28>에 제시하였다.

<표 IV-28> 과학과 관련된 정의적 영역 검사 정규성 검정

하위 요소	기술통계(N=14)				통계량	유의도
	M	SD	Max	Min		
인식	3.739	.455	4.50	3.08	.954	.619
흥미	3.700	.709	4.93	1.93	.899	.109
태도	3.744	.405	4.24	2.90	.940	.417
평균	3.729	.437	4.25	2.65	.876	.051

정규성 검정 결과 과학과 관련된 인식, 흥미, 태도와 평균 점수가 모두 유의수준인 .05보다 크기 때문에 귀무가설이 채택되어 정규분포임을 확인하였다.

본 연구에서 개발한 프로그램이 과학과 관련된 정의적 특성에 미친 영향을 알아보기 위해 사전·사후검사 결과를 분석하였다. <표 IV-29>에서 제시하였듯이 과학에 대한 인식, 흥미, 태도, 평균의 모든 측면에서 점수가 상승하였지만 모든 영역의 유의확률이 유의수준인 .05보다 크기 때문에 통계적으로는 유의미하지 않은 것으로 나타났다.

<표 IV-29> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교

하위 요소	시기	N	M	SD	t	유의도
인식	사전	13	3.739	0.455	-.612	.551
	사후	13	3.816	0.589		
흥미	사전	13	3.700	0.709	-.817	.429
	사후	13	3.790	0.713		
태도	사전	13	3.744	0.405	-1.663	.120
	사후	13	3.846	0.436		
평균	사전	13	3.729	0.437	-1.478	.163
	사후	13	3.821	0.513		

(5) 연구 결과 분석

본 연구에서 개발한 코듀 프로그래밍 중심의 STEAM 교육 프로그램의 교육적 효과를 살펴보기 위해 창의력, 논리적 사고력, 과학의 정의적 영역에 대한 검사 결과를 <표 IV-30>에 제시하였다.

<표 IV-30> 실험의 결과 분석

	집단	상승 하위요소	
집단 간 비교	실험집단 우위	창의력	유창성, 창의력 평균, 창의력 지수
		논리적 사고력	조합논리
	비교집단 우위	창의력	-
		논리적 사고력	-
집단 내 비교	실험집단	창의력	유창성, 독창성, 창의력 평균, 창의력 지수
		논리적 사고력	논리적 사고력 합계
		과학의 정의적 영역	-
	비교집단	창의력	유창성, 독창성, 창의력 평균
		논리적 사고력	-

창의력 검사에서는 유창성과 독창성의 영역, 창의력 평균과 창의력 지수가 통계적으로 유의미한 상승한 것으로 나타났다. 이러한 결과는 본 연구에서 개발한 교육 STEAM 교육 프로그램이 STEAM 교육에서 강조하는 창의력 향상에 기여하고 있음을 보여준다.

논리적 사고력 합계가 유의미한 향상이 나타난 것은 STEAM 형태의 프로그래밍 교육이 논리적 사고력을 길러주며 이를 통해 프로그래밍 교육 자체에도 효과가 있음을 보여준다.

다만 과학과 관련된 정의적 영역에 대한 유의미한 상승을 이끌어 내지 못한 점은 다소 아쉬운 부분이다.

2. 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램

1) 요구 분석

스마트폰, 로봇이 대중화되면서 센서에 대한 관심이 높아지고 있다. 최근 아두이노 등의 센서보드를 활용한 프로그래밍 교육에 대한 다양한 연구가 진행되고 있다. 이번 연구에서는 STEAM 교과 융합을 위한 도구로 센서를 선정하여 교육 프로그램을 진행하였다. 센서를 이용한 스크래치 프로그래밍 활동을 통해 과학 주제를 학습하고 융합적인 사고를 경험함으로써 프로그래밍과 과학 학습에 흥미를 느낄 수 있도록 STEAM 교육 프로그램을 구성하였다. 교육 프로그램에서 중요한 요소인 센서를 활용하기 위한 도구로 센서보드를 사용하였다. 센서보드는 소리 센서, 빛 센서, 전기 저항 센서, 슬라이드와 버튼이 있어 소리, 빛, 전기신호, 슬라이드와 버튼의 입력을 받을 수 있다는 장점이 있다. 과학 현상인 빛, 소리 등을 직접 입력 받을 수 있어 이를 이용한 다양한 프로그래밍이 가능하고 이를 통해 과학 교과에서 배우던 개념을 직접 다루고 그것을 이용하여 창의적인 프로그램을 만들 수 있다.

2) 설계 및 개발

(1) 주제의 선정

센서 기반의 스크래치 프로그래밍과 과학 주제를 융합하기 위하여 2009 개정 과학과 교육과정 중 초등학교 4~6학년의 내용을 분석하였다. 과학 교과를 중심으로 내용을 선정한 이유는 STEAM 교육이 과학에 대한 흥미와 태도를 향상시키기 위한 것이고 궁극적으로 인류가 직면한 문제의 해결과 과학에 대한 경쟁력을 기르기 위한 목적을 갖고 있기 때문이다. 과학 분야의 학습 주제와 스크래치와 센서가 갖고 있는 기술, 공학, 예술, 수학적인 부분을 융합하는 것이 본 연구의 핵심이다.

교육과정을 분석하여 15개의 예비 주제를 선정하였는데, 센서 기반의 프로그래밍 교육의 적합 여부, STEAM 교육 주제로서의 효과성 여부, 학생의 흥미를 유발 여부를 기준으로 컴퓨터과학 교육, 초등교육 전문가 집단과 협의하여 5개의 주제를 선정하였다. 선정된 주제는 <표 IV-31>과 같다.

<표 IV-31> 선정 주제

선정 주제	관련 내용 요소	주요 센서
생물 보호	생태계와 환경, 빛	빛 센서
에너지 절약	에너지	빛, 소리 센서
환경 보호	생태계와 환경	빛, 소리 센서
미래의 로봇	지층과 화석, 태양계와 별	슬라이드, 버튼 센서
전기 저항	전기 회로	전기 저항 센서

(5) 세부 학습내용

교육 프로그램의 진행은 크게 3개의 단계로 나눌 수 있다. 스크래치에 대한 기본 소양을 학습하는 단계, 선정된 과학 주제에 대하여 조사하고 토의하여 센서를 활용한 스크래치 프로그래밍을 통해 STEAM 소양을 기르는 단계, 자신이 선정한 과학 주제에 대한 프로그램을 디자인하고 작성하는 프로젝트 단계이다. 교육 프로그램의 단계별 구성은 <표 IV-32>에 제시하였다.

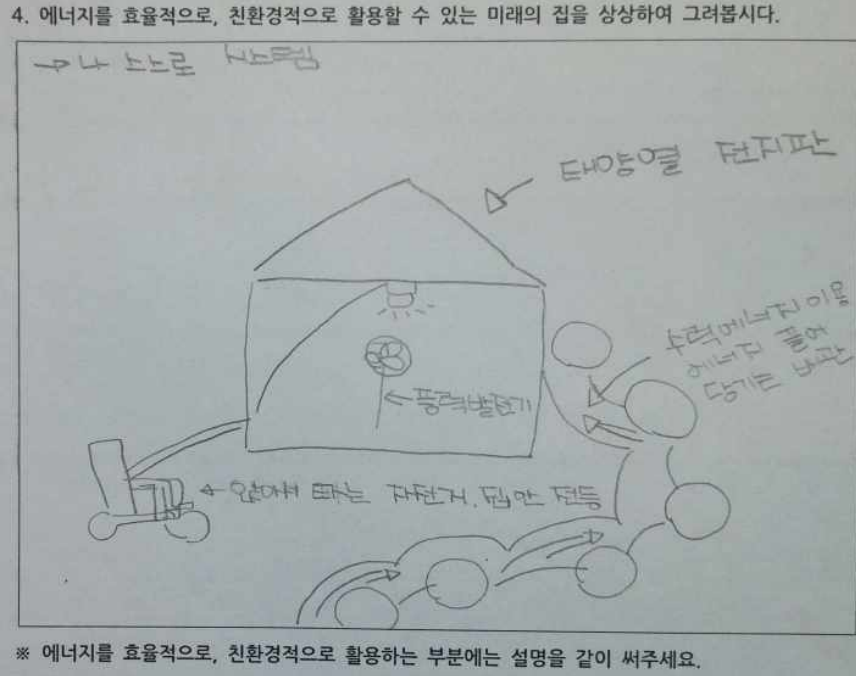
<표 IV-32> 교육 프로그램의 단계별 구성

단계(시수)	내용
스크래치 학습 (6)	<ul style="list-style-type: none"> • 스크래치 메뉴 익히기 • 조건문, 반복문 익히기 • 다양한 블록 명령어 익히기 • 방송하기 기능 활용하기 • 센서 알기 • 간단한 게임 만들기
STEAM 주제 학습 (15)	<ul style="list-style-type: none"> • 1주제 - 야행성 동물(반딧불이) • 2주제 - 에너지 절약 • 3주제 - 분리수거 • 4주제 - 미래의 로봇 설계 • 5주제 - 과일을 이용한 입력 장치
프로젝트 (4)	<ul style="list-style-type: none"> • 개인 프로젝트 준비 • 프로젝트 발표

STEAM 주제 학습 단계는 일리노이주 주립대의 CeMaST(Center for Mathematics, Science, and Technology)의 IMaST(Integrated Mathematics, Science, and Technology) 프로젝트에서 사용되었던 순환학습 모형을 본 연구에 맞게 수정하여 진행하였다. 학생들은 발표, 조사, 토의하는 활동을 통해 주어진 문제를 학습한다. 그 과정에서 생성된 아이디어를 반영하여 이야기나 게임 등의 형태로 프로그래밍을 작성하고 이것을 다른 사람에게 디지털 스토리텔링하는 활동을 진행하였다. 학습 모형에 따른 활동 내용을 <표 IV-33>에 제시하였다.

<표 IV-33> 학습 모형

단계	활동
탐색하기	<ul style="list-style-type: none"> • 동기유발 • 이야기책, 생활 속의 과학적 문제 제시
아이디어 얻기	<ul style="list-style-type: none"> • 과학적 사실의 조사, 발표, 토의
아이디어 적용하기	<ul style="list-style-type: none"> • 주제에 대한 프로그래밍 • 디지털 스토리텔링 밑그림
아이디어 확장하기	<ul style="list-style-type: none"> • 자신만의 프로그램 만들기 • 창의적인 스토리 만들어 내기
평가하기	<ul style="list-style-type: none"> • 디지털 스토리텔링하기 • 상호평가, 학습 소감발표, 정리



[그림 IV-1] 프로그램 구상 및 학생 작품 예시

아이디어 적용하기 단계에서 교사가 미리 준비한 예제를 통해 센서를 활용하는 방법을 따라하기 식의 프로그래밍 했다면 아이디어 확장하기 단계에서는 학습한 스크래치 블록과 센서를 활용한 자신만의 프로그램을 만드는 활동을 진행하였다. 1주제 야행성 동물(빛 센서)와 5주제 과일을 이용한 입력 장치(전기저

항 센서)처럼 특정 센서가 필요한 경우를 제외하고는 센서의 활용에 제한을 두지 않았기 때문에 처음 활용하는 시점에서는 주로 그 차시에 배운 내용을 활용하는 경향이 강했으나 시간이 지날수록 자기만의 창의적인 스토리를 만들어 내기 위해 여러 개의 센서를 활용하는 등 각자의 아이디어로 센서를 활용하는 모습을 관찰할 수 있었다. 교재를 자체 제작하여 활용하였는데 학생들의 아이디어를 만들어 내고 프로그램을 설계할 수 있도록 스케치를 통한 구상활동을 하도록 하였다. 구상활동이 끝나면 구상한 프로그램을 완성하고 이것을 이야기로 풀어 쓰고 자신만의 이야기를 발표하는 활동을 가졌다. [그림 IV-1]에 한 학생이 수업시간에 작성한 구상활동 부분과 이를 통해 완성된 프로그램의 예제를 제시하였다.

3) 적용 및 효과 분석

(1) 연구 가설

① 가설 1

- 연구가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 없다.
- 대립가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 있다.

② 가설 2

- 연구가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 없다.
- 대립가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 있다.

③ 가설 3

- 연구가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 없다.
- 대립가설: 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 있다.

(2) 연구 대상

본 연구의 실험집단은 제주지역 J대학교 창의컴퓨터교실에 지원한 14명의 학생을 선정하였다. 창의컴퓨터교실은 지역 내 4, 5학년 초등학생 중 프로그래밍 강좌의 수강을 희망하는 학생들을 모집하여 교육기부 형태로 운영되었다. 실험 집단에는 센서 기반 스크래치 프로그래밍을 통한 STEAM 교육 프로그램을 투입하였다.

비교집단은 제주지역 D초등학교의 25명의 5학년 학생들을 선정하였다. 비교 집단에는 개별적인 과학, 수학, 그리고 창의적 체험활동 시간을 활용한 ICT 활용 수업을 진행하였다. 연구 대상에 대한 구체적인 사항을 <표 IV-34>에 제시하였다.

<표 IV-34> 연구 대상

구분	실험집단(명)			비교집단(명)		
	남	여	소계	남	여	소계
4학년	4	5	10	0	0	0
5학년	4	10	14	13	12	25
소계	8	6	14	13	12	25

(3) 연구의 절차

본 연구에서 개발한 센서 기반 스크래치 프로그래밍을 통한 STEAM 교육 프로그램은 실험집단을 대상으로 방학기간 6일간 총 24차시의 집합 강의 및 실습으로 구성하였다. 6일 중 교육프로그램이 시작되는 첫 날에는 사전검사와 교육 프로그램에 대한 오리엔테이션을 실시하였으며 마지막 날에는 교육프로그램 기간 동안 학습한 내용을 바탕으로 본인이 직접 센서와 주제를 선택하여 프로그래밍을 완성하고 발표하는 시간과 함께 사후검사를 실시하였다. 실험집단에 투입한 교육 프로그램의 투입 일정은 <표 IV-35>와 같다.

<표 IV-35> 프로그램 일정

일차	프로그램 일정
1	사전검사, 오리엔테이션(3차시)
2	스크래치 소양 학습(6차시)
3	1, 2주제 학습(각 3차시)
4	3, 4주제 학습(각 3차시)
5	5주제 학습(3차시), 개인별 프로젝트 준비(3차시)
6	프로젝트 발표 및 사후검사(3차시)

교육 프로그램이 실험집단의 창의력과 논리적 사고력, 과학과 관련된 정의적 영역에 미치는 영향을 살펴보기 위하여 이에 대한 사전, 사후검사를 실시하였으며 본 실험 연계를 도식화 하여 <표 IV-36>에 제시하였다.



집단	사전검사	처치	사후검사
실험집단	O ₁ , O ₃	X ₁	O ₂ , O ₄
비교집단	O ₁	X ₂	O ₂

- O₁ : 논리적 사고력, 창의력 사전검사
- O₂ : 논리적 사고력, 창의력 사후검사
- O₃ : 과학과 관련된 정의적 영역 사전검사
- O₄ : 과학과 관련된 정의적 영역 사후검사
- X₁ : 스크래치 프로그래밍 중심 STEAM 교육 프로그램
- X₂ : 일반적인 교과수업, 방과후교육 프로그램

(4) 연구 결과

① 창의력 변화

- 정규성 검정

창의력 검사 결과의 집단 간, 각 집단 별 비교 전에 두 집단 모두 30명 이하의 집단이기 때문에 정규분포임을 확인하기 위하여 사전검사에 대하여

Shapiro-Wilks 정규성 검정을 실시하였다. 실험, 비교집단의 정규성 검정 결과를 각각 <표 IV-37>, <표 IV-38>에 제시하였다.

<표 IV-37> 실험집단 창의력 검사 정규성 검정

하위요소	기술통계(N=14)				Sig.	p
	M	SD	Max	Min		
유창성	103.571	11.614	123	85	.959	.714
독창성	104.929	15.163	127	82	.925	.264
제목의 추상성	101.571	32.837	115	40	.926	.270
정교성	121.500	18.325	150	79	.938	.397
성급한 종결에 대한 저항	93.571	24.200	112	40	.701	.000**
창의력 평균	104.857	14.330	121	67	.878	.054
창의력 지수	115.000	16.984	139	7-3	.911	.161

**p<.01

실험집단의 창의력 검사에 대한 정규성 검정 결과 성급한 종결에 대한 저항 요소를 제외한 하위요소에서 정규성이 만족된 것으로 나타났다.

<표 IV-38> 비교집단 창의력 검사 정규성 검정

하위요소	기술통계(N=25)				Sig.	p
	M	SD	Max	Min		
유창성	81.360	11.593	108	52	.965	.512
독창성	80.040	17.326	135	51	.916	.041*
제목의 추상성	110.280	21.157	148	40	.891	.012*
정교성	86.480	24.227	145	57	.912	.034*
성급한 종결에 대한 저항	89.840	20.174	115	40	.918	.045*
창의력 평균	89.600	15.313	113	53	.961	.426
창의력 지수	97.960	17.820	124	53	.959	.391

*p<.05

비교집단의 창의력 검사에 대한 정규성 검정 결과 유창성, 창의력 평균, 창의력 지수에 대해서만 정규성이 만족되었으며 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 저항의 요소에서는 정규성을 만족하지 않았다.

실험, 비교집단의 창의력 검사 정규성 검정 결과에 따라 집단 간, 집단 별 비교 시에 정규성이 확보된 하위요소에 대하여 모수검정을, 정규성을 만족하지 못한 하위요소에 대해서는 비모수 검정 방법을 사용하였고 연구결과를 분석하였다.

• 집단 간 비교

<표 IV-39>, <표 IV-40>에서 제시한 것처럼 두 집단 모두 정규성을 확보한 창의력의 하위요소인 유창성, 창의력 평균, 창의력 지수에 대한 사전검사에 독립표본 t검정을 실시하였고 두 집단 중 하나의 집단에서라도 정규성을 확보하지 못한 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 저항 하위요소에 대해서는 Mann-Whitney U 검증을 실시하였다.

<표 IV-39> 창의력 사전검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
유창성	실험	14	103.571	11.614	5.736	.000**
	비교	25	81.360	11.593		
창의력 평균	실험	14	104.857	14.330	3.052	.004**
	비교	25	89.600	15.313		
창의력 지수	실험	14	115.000	16.984	2.912	.006**
	비교	25	97.960	17.820		

**p<.01

<표 IV-40> 창의력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
독창성	실험	14	104.929	15.163	29.64	40.000	.000**
	비교	25	80.040	17.326	14.60		
제목의 추상성	실험	14	101.571	32.837	17.82	144.500	.369
	비교	25	110.280	21.157	21.22		
정교성	실험	14	121.500	18.325	29.25	45.500	.000**
	비교	25	86.480	24.227	14.82		
성급한 종결에 대한 저항	실험	14	93.571	24.200	22.43	141.000	.317
	비교	25	89.840	20.174	18.64		

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

모수통계와 비모수통계의 방법을 사용하여 창의력 사전검사의 결과를 분석한 결과 제목의 추상성과 성급한 종결에 대한 저항의 요소를 제외하고는 실험집단이 비교집단에 비해 평균이 매우 높아 두 집단이 이질집단으로 나타났다. 창의력 검사 결과 비교에 사용된 데이터는 6,452명의 TTCT 검사 표본에 따른 데이터를 정규분포곡선을 이용하여 평균치 100, 표준편차가 20이 되도록 만든 표준점수를 사용하였다(김영채, 2010). 창의력 평균에 창의점 감점 요소를 추가한 창의력 지수 점수(해당연령 평균이 111.38)를 제외한 모든 하위요소의 평균이 100이라는 의미인데 비교집단의 데이터는 창의력 평균이 89.600으로, 가장 낮은 하위요소인 독창성의 경우에는 80.040으로 나타났다. 정확한 원인을 찾을 수는 없지만 평균점수가 모집단에 비해 너무 낮게 나타나고 정규성 분포를 따르지 않는 것으로 나타나 창의력 검사의 모든 하위 요소에 대해서 검증하기에는 한계가 있었다. 따라서 두 집단 모두 정규성을 확보하였지만 사전검사 결과 통계적으로 유의미한 차이를 보인 유창성, 창의력 평균, 창의력 지수에 대해서 공분산 분석을 실시하였고 정규성을 확보하지 못했지만 제목의 추상성과 성급한 종결에 대한 저항에 대해서는 Mann-Whitney U 검증을 이용하여 사후검사 결과를 비교하였다.

먼저 유창성, 창의력 평균, 창의력 지수에 대하여 공분산분석을 실시하여 결과를 <표 IV-41>와 <표 IV-42>에 제시하였다.

실험집단의 하위요소별 교정평균은 유창성이 107.082, 창의력 평균이 101.369, 창의력지수가 112.158이고, 비교집단의 하위요소별 교정평균은 유창성이 96.354, 창의력 평균이 97.873, 창의력 지수가 105.592이다. 교정된 사후 창의력 하위요소별 결과가 집단에 따라 어떠한 차이가 있는지에 대한 공분산분석 결과는 <표 IV-42>와 같다. 사전 유창성, 사전 창의력 평균, 사전 창의력 지수의 F통계값은 각각, 52.490, 81,750, 86,854이고 유의확률은 3요소 모두 .000으로 사전 창의력 평가 결과가 사후 점수에 유의한 영향을 미치고 있는 것으로 나타났다. 사전 결과가 사후 결과에 미치는 영향을 통제된 후의 사후평균의 F값은 각각 2.995, .958, 2.633이고 유의 확률이 .092, .334, .113으로 두 집단의 교정된 사후 결과에 유의한 차이는 없었다.

<표 IV-41> 집단에 따른 유창성, 창의력 평균, 창의력 지수에 대한 서술통계

하위요소	구분		실험집단	비교집단	합계
유창성	사전	M	103.571	81.360	89.333
		SD	11.614	11.593	15.734
	사후	M	118.214	90.120	100.205
		SD	11.116	18.267	20.963
	교정 사후	M	107.082	96.354	
		SD	4.524	3.103	
	N		14	25	39
창의력 평균	사전	M	104.857	89.600	95.077
		SD	14.330	15.313	16.533
	사후	M	109.214	93.480	99.128
		SD	13.389	16.202	16.899
	교정 사후	M	101.369	97.873	
		SD	2.754	1.997	
	N		14	25	39
창의력 지수	사전	M	115.000	97.960	104.077
		SD	16.984	17.820	19.178
	사후	M	120.786	100.760	107.949
		SD	17.012	17.838	19.867
	교정 사후	M	112.158	105.592	
		SD	3.130	2.275	
	N		14	25	39

<표 IV-42> 집단에 따른 교정된 하위요소에 대한 공분산분석 결과

하위요소	분산원	제곱합	자유도	평균제곱	F	p
유창성	공분산(사전유창성)	9580.798	1	9580.798	52.490	.000
	집단	546.644	1	546.644	2.995	.092
	오차	6570.917	36	182.525		
	합계	408300.000	39			
창의력 평균	공분산(사전창의력평균)	7473.637	1	7473.637	81.750	.000
	집단	87.593	1	87.593	.958	.334
	오차	3291.129	36	91.420		
	합계	394082.000	39			
창의력 지수	공분산(사전창의력지수)	10378.768	1	10378.768	86.804	.000
	집단	314.777	1	314.777	2.633	.113
	오차	4304.352	36	119.565		
	합계	469462.000	39			

다음으로 정규성을 확보하지 못한 제목의 추상성과 성급한 종결에 대한 저항에 대해서도 사후검사 결과를 비교하여 <표 IV-43>에 제시하였는데 두 요소 모두 유의확률이 유의수준보다 낮아 두 집단 간 유의미한 차이가 나타나지 않았다.

<표 IV-43> 창의력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
제목의 추상성	실험	14	97.786	33.889	17.43	139.000	.291
	비교	25	109.240	21.873	21.44		
성급한 종결에 대한 저항	실험	14	90.786	26.385	19.14	163.000	.724
	비교	25	96.720	21.573	20.48		

이상 유창성, 창의력 평균, 창의력 지수에 대해서 실시한 공분산 분석과 제목의 추상성과 성급한 종결에 대한 저항에 대해서는 실시한 Mann-Whitney U 검증결과 두 집단에 유의미한 차이는 없는 것으로 나타났다.

• 집단 내 비교

이번에는 각 집단별로 사전, 사후 결과를 비교해 보았다. 실험집단의 경우 정규성을 확보하지 못한 성급한 종결에 대한 저항을 제외한 영역에 대해 대응 표본 t검정을 실시하였으며 그 결과는 <표 IV-44>과 같다.

<표 IV-44> 실험집단 창의력 사전·사후검사 결과 비교(독립표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	103.571	11.614	-.345	.001**
	사후	118.214	11.116		
독창성	사전	104.929	15.163	-2.597	.041*
	사후	115.000	10.727		
제목의 추상성	사전	101.571	32.837	-2.143	.524
	사후	97.786	33.889		
정교성	사전	121.500	18.325	-.981	.420
	사후	124.571	20.213		
창의력 평균	사전	104.857	14.330	-2.747	.071
	사후	109.214	13.389		
창의력 지수	사전	115.000	16.984	-3.188	.034*
	사후	120.786	17.012		

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

유창성에서 t 통계값은 -.345, 유의확률은 .001, 독창성에서 t 통계값은 -2.597, 유의확률은 .041, 창의력 지수에서 t 통계값 , -3.188, 유의확률 .034로서 각각 유의수준 .05에서 사전점수에 비해 사후점수에서 유의미한 상승이 있는 것으로 나타났다.

정규분포를 따르지 않는 성급한 종결에 대한 저항 하위요소에는 Wilcoxon 부호순위 검정을 실시하였고 그 결과를 <표 IV-45>에 제시하였다. 검정 사전·사후검사에 유의한 차이가 없는 것으로 나타났다.

<표 IV-45> 실험집단의 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
성급한 종결에 대한 저항	실험	14	사전	93.571	24.200	-.415	.678
			사후	90.786	26.385		

비교집단의 사전·사후 결과에 대해서 정규성을 만족한 유창성, 창의력 평균, 창의력 지수에는 대응표본 t검정을, 정규성을 만족하지 않은 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 저항의 요소에서는 Wilcoxon 부호순위 검정을 실시하였다.

먼저 유창성과 창의력 평균, 창의력 지수에 대한 검정 결과를 <표 IV-46>에 제시하였다.

<표 IV-46> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	81.360	11.593	-3.231	.004**
	사후	90.120	18.267		
창의력 평균	사전	89.600	15.313	-1.821	.081
	사후	93.480	16.202		
창의력 지수	사전	97.960	17.820	-1.125	.272
	사후	100.760	17.838		

**p<.01

비교집단의 사전 점수와 사후 점수의 차이에 대한 통계적 유의성을 검정한

결과 유창성 요소에서 t 통계값이 -3.231, 유의확률이 .004로서 유의수준 .05에서 사전과 사후의 유창성 사이에 차이가 있는 것으로 나타났다.

정규성을 만족하지 못한 독창성, 제목의 추상성, 정교성, 성급한 종결에 대한 Wilcoxon 부호순위 검정 결과를 <표 IV-47>에 제시하였다. 검정결과 사전·사후 결과 사이에 유의한 차이가 발생하지 않았다.

<표 IV-47> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
독창성	비교	25	사전	80.040	17.326	-1.400	.162
			사후	83.720	17.662		
제목의 추상성	비교	25	사전	110.280	21.157	-.130	.897
			사후	109.240	21.873		
정교성	비교	25	사전	86.480	24.227	-.057	.954
			사후	87.360	25.179		
성급한 종결에 대한 저항	비교	25	사전	89.840	20.174	-1.836	.066
			사후	96.720	21.573		

창의력 검사 분석 결과 본 연구에서 개발한 교육 프로그램 투입 후에 학습자의 창의력 하위요소 중 유창성, 독창성에서 통계적으로 유의한 상승을 나타냈고, 창의력 지수에 유의한 영향을 준 것으로 볼 때 전체적인 창의력 신장에 기여한 것으로 나타났다.

② 논리적 사고력 변화

• 정규성 검정

두 집단의 논리적 사고력 변화를 살펴보기 위해 두 집단의 사전검사에 대한 Shapiro-Wilks 정규성 검정을 실시해서 그 결과를 <표 IV-48>, <표 IV-49>에 제시하였다.

<표 IV-48> 실험집단 논리적 사고력 검사 정규성 검정

하위요소	기술통계(N=14)				Sig.	p
	M	SD	Max	Min		
보존	1.214	0.426	2	1	.516	.000**
비례	0.286	0.469	1	0	.576	.000**
변인통제	0.714	0.825	2	0	.767	.002**
확률	0.429	0.852	2	0	.516	.000**
상관	0.000	0.000	0	0	-	-
조합	1.643	0.745	2	0	.532	.000**
합계	4.286	2.128	8	1	.965	.801

**p<.01

실험집단의 논리적 사고력 사전검사의 결과에 대하여 정규성 검정을 실시한 결과 합계를 제외한 모든 하위요소에서 정규분포를 따르지 않는 것으로 나타났다.



<표 IV-49> 비교집단 논리적 사고력 검사 정규성 검정

하위요소	기술통계(N=25)				Sig.	p
	M	SD	Max	Min		
보존	1.240	0.523	2	0	.691	.000**
비례	0.280	0.614	2	0	.515	.000**
변인통제	0.520	0.586	2	0	.721	.000**
확률	0.360	0.638	2	0	.610	.000**
상관	0.160	0.374	1	0	.445	.000**
조합	1.280	0.737	2	0	.785	.000**
합계	3.840	1.599	7	1	.918	.046*

**p<.01

비교집단의 논리적 사고력 사전 결과에 대한 정규성 검정 결과 모든 하위 요소와 논리적 사고력 합계에서 정규분포를 따르지 않는 것으로 나타났다. 따라서 논리적 사고력 검사 결과에 대한 두 집단 간, 집단 내 비교를 위하여 비모수적 통계방법을 사용할 필요가 있었다.

- 집단 간 비교

본 연구에서 개발한 교육 프로그램이 실험집단의 논리적 사고력에 미친 영향을 알아보고자 실험집단과 비교집단의 논리적 사고력 검사 결과를 실시하였다. 먼저 두 집단의 사전검사 결과를 Mann-Whitney U검정을 실시하여 비교하였고 <표 IV-50>에 제시하였다.

<표 IV-50> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	14	1.214	0.426	19.57	169.000	.823
	비교	25	1.240	0.523	20.24		
비례	실험	14	0.286	0.469	20.79	164.000	.661
	비교	25	0.280	0.614	19.56		
변인통제	실험	14	0.714	0.825	21.29	157.000	.558
	비교	25	0.520	0.586	19.28		
확률	실험	14	0.429	0.852	19.71	171.000	.878
	비교	25	0.360	0.638	20.16		
상관	실험	14	0.000	0.000	18.00	147.000	.119
	비교	25	0.160	0.374	21.12		
조합	실험	14	1.643	0.745	23.75	122.500	.085
	비교	25	1.280	0.737	17.90		
합계	실험	14	4.286	2.128	21.82	149.500	.448
	비교	25	3.840	1.599	18.98		

<표 IV-50>에서 제시한 것처럼 실험·비교 집단의 논리적 사고력 사전검사 결과를 비교한 결과 모든 영역의 유의 확률이 유의수준인 .05보다 크게 나타나 두 집단이 동질집단임이 확인되었다.

교육 프로그램을 투입한 후 논리적 사고력의 변화를 알아보기 위해 사후검사를 실시하였고 그 결과를 Mann-Whitney U검정을 통해 분석하여 <표 IV-51>에 제시하였다.

<표 IV-51> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	14	1.357	0.497	21.25	157.500	.566
	비교	25	1.200	0.707	19.30		
비례	실험	14	0.500	0.650	23.29	129.000	.077
	비교	25	0.200	0.500	18.16		
변인통제	실험	14	0.929	0.917	24.36	114.000	.036*
	비교	25	0.360	0.700	17.56		
확률	실험	14	0.500	0.855	20.54	167.500	.781
	비교	25	0.360	0.638	19.70		
상관	실험	14	0.071	0.267	19.39	166.500	.636
	비교	25	0.120	0.332	20.34		
조합	실험	14	1.786	0.426	25.14	103.000	.017*
	비교	25	1.280	0.678	17.12		
합계	실험	14	5.143	2.282	25.61	96.500	.020*
	비교	25	3.520	1.960	16.86		

*p<.05

사후검사에 대한 실험·비교집단의 비교 결과 변인통제 논리의 유의확률이 .036, 조합논리의 유의확률이 .017, 논리적 사고력의 유의확률이 .020로 유의수준인 .05 하에서 유의한 것으로 나타났다.

• 집단 내 비교

본 연구에서 개발한 교육 프로그램이 논리적 사고력에 미친 영향을 좀 더 살펴보기 위해서 각 집단별로 사전·사후검사를 비교하였다. 두 집단 모두 대부분의 영역에서 정규성을 만족하지 못해 비모수적 통계방법인 Wilcoxon 부호순위 검정을 사용하였다.

<표 IV-52> 집단별 논리적 사고력 검사결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
보존	실험	14	사전	1.214	0.426	-1.000	.317
			사후	1.357	0.497		
	비교	25	사전	1.240	0.523	-.378	.705
			사후	1.200	0.707		
비례	실험	14	사전	0.286	0.469	-1.134	.257
			사후	0.500	0.650		
	비교	25	사전	0.280	0.614	-1.000	.317
			사후	0.200	0.500		
변인통제	실험	14	사전	0.714	0.825	-1.342	.180
			사후	0.929	0.917		
	비교	25	사전	0.520	0.586	-1.414	.157
			사후	0.360	0.700		
확률	실험	14	사전	0.429	0.852	-1.000	.317
			사후	0.500	0.855		
	비교	25	사전	0.360	0.638	.000	1.000
			사후	0.360	0.638		
상관	실험	14	사전	0.000	0.000	-1.000	.317
			사후	0.071	0.267		
	비교	25	사전	0.160	0.374	-.577	.564
			사후	0.120	0.332		
조합	실험	14	사전	1.643	0.745	-1.414	.157
			사후	1.786	0.426		
	비교	25	사전	1.280	0.737	.000	1.000
			사후	1.280	0.678		
합계	실험	14	사전	4.286	2.128	-2.521	.012*
			사후	5.143	2.282		
	비교	25	사전	3.840	1.599	-1.507	.132
			사후	3.520	1.960		

*p<.05

<표 IV-52>에서 각 집단별 논리적 사고력 사전·사후검사를 비교한 결과 실험집단의 논리적 사고력 합계의 Z 통계값이 -2.521이고 유의확률이 .012로 유의수준 .05에서 본 연구에서 개발한 STEAM 교육 프로그램에 의한 사전과 사후 논리적 사고력에 차이가 있는 것으로 나타났다.

③ 과학과 관련된 정의적 영역

실험집단의 과학과 관련된 정의적 영역에 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-53>에 제시하였다.

<표 IV-53> 과학과 관련된 정의적 영역 검사 정규성 검정

하위 요소	기술통계(N=14)				통계량	유의도
	M	SD	Max	Min		
인식	3.951	.441	4.75	3.33	.958	.690
흥미	4.029	.646	4.87	2.93	.902	.120
태도	3.918	.632	4.95	2.05	.958	.685
평균	3.962	.546	4.77	3.13	.927	.272

정규성 검정 결과 과학과 관련된 인식, 흥미, 태도와 평균 점수가 모두 유의수준인 .05보다 크기 때문에 귀무가설이 채택되어 정규분포임을 확인하였다.

본 연구에서 개발한 프로그램이 과학과 관련된 정의적 특성에 미친 영향을 알아보기 위해 사전·사후검사 결과를 분석하였다. <표 IV-54>에서 제시하였듯이 과학에 대한 인식, 흥미, 태도, 평균의 모든 측면에서 점수가 상승한 것을 알 수 있다. 특히 과학적 태도의 경우 그 변화가 통계적으로도 유의미한 것으로 나타났다. 센서 기반 스크래치 프로그래밍 중심의 STEAM 교육 프로그램이 실험집단의 과학적 태도가 향상되었으며 통계적으로 유의미하지는 않았지만 인식과 흥미, 평균 점수가 향상된 부분도 과학과 관련된 정의적 특성에 긍정적인 영향을 주었음을 보여주고 있다.

<표 IV-54> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교

하위 요소	시기	N	M	SD	t	유의도
인식	사전	14	3.951	0.441	-.203	.842
	사후	14	3.976	0.466		
흥미	사전	14	4.029	0.646	-1.611	.131
	사후	14	4.229	0.659		
태도	사전	14	3.918	0.632	-2.378	.033*
	사후	14	4.091	0.656		
평균	사전	14	3.962	0.546	-2.135	.052
	사후	14	4.106	0.534		

*p<.05

④ 연구 결과 분석

본 연구에서 개발한 센서 기반 스크래치 프로그래밍 중심의 STEAM 교육 프로그램의 교육적 효과를 살펴보기 위해 창의력, 논리적 사고력, 과학의 정의적 영역에 대한 검사 결과를 <표 IV-55>에 제시하였다.

<표 IV-55> 실험의 결과 분석

	집단	상승 하위요소	
집단 간 비교	실험집단 우위	창의력	-
		논리적 사고력	변인통제 논리, 조합논리, 합계논리
	비교집단 우위	창의력	-
		논리적 사고력	-
집단 내 비교	실험집단	창의력	유창성, 독창성, 창의력 지수
		논리적 사고력	논리적 사고력 합계
		과학의 정의적 영역	과학에 대한 태도
	비교집단	창의력	유창성
논리적 사고력		-	

창의력 검사에서는 유창성과 독창성의 영역, 창의력 지수가 통계적으로 유의미한 상승한 것으로 나타났다. 이러한 결과는 본 연구에서 개발한 교육 프로그램이 STEAM 교육에서 강조하는 창의력 향상에 기여하고 있음을 보여준다.

논리적 사고력 합계가 유의미한 향상이 나타난 것은 STEAM 형태의 프로그램

래밍 교육이 논리적 사고력을 길러주며 이를 통해 프로그래밍 교육 자체에도 효과가 있음을 보여준다.

마지막으로 과학과 관련된 정의적 영역에 대한 검사에서 각 영역의 점수가 향상되었고 과학적 태도에서 유의미한 상승을 나타낸 것은 프로그래밍과 과학 교과를 융합한 STEAM 교육의 교육적인 가능성을 보여주는 것이다.

3. 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램

1) 요구 분석

센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램 개발 및 적용을 통해 피지컬 컴퓨팅과 프로그래밍을 활용한 STEAM 교육 프로그램의 효과를 살펴보았다. 스크래치는 교육용 프로그래밍 언어로서 STEAM 교육을 위한 다양한 장점을 갖고 있는데 이러한 STEAM 교육 도구로서의 가능성을 확인하기 위해 스크래치 프로그래밍 중심 STEAM 교육 프로그램을 개발하여 적용하였다.

2) 설계 및 개발

(1) 주제의 선정

본 연구에서는 STEAM 교육 프로그램의 주제를 선정하기 위해 2009 개정 초등학교 4~6학년 교육과정을 분석하여 학습 주제를 선정하였다. STEAM 교과 중에서 과학 주제를 중심으로 타 교과의 학습 내용을 융합하여 과목 간의 연계가 용이하고 스크래치를 통한 프로그래밍과 디지털 스토리텔링에 적합한 주제를 선정하고자 하였다. 2009 개정 과학과 교육과정의 네 가지 영역인 운동과 에너지, 물질, 생명, 지구와 우주를 골고루 반영하여 <표 IV-56>처럼 주제를 선정하였다.

<표 IV-56> 선정 주제

선정 주제	과학 교과 관련 영역	관련 내용 요소
생물	생명	생물의 보호책
물질	물질	물의 순환
빛	운동과 에너지	빛
에너지	운동과 에너지	신재생 에너지
우주	지구와 우주	태양계 시뮬레이션

(2) 세부 학습 내용

스크래치는 프로그래밍을 처음 접하는 초등학생도 쉽게 프로그래밍을 배울 수 있을 뿐 아니라 다양한 미디어를 제공하여 프로그래밍 과정에서 학습자의 융합적 소양을 이끌어 낼 수 있다. 선정된 주제에 대한 과학적 지식이나 원리를 학습하고 이러한 지식이나 원리가 반영된 프로그램을 만들면서 과학 뿐 만 아니라 예술(미술, 음악)적 소양도 함께 기를 수 있다. 또한 프로그래밍을 설계하고 작성하는 과정에서 공학적, 기술적인 체험도 함께 할 수 있어 자연스럽게 STEAM 교과가 융합되는 경험을 할 수 있다.

<표 IV-57> 교육 프로그램의 단계별 구성

단계(시수)	내용
스크래치 학습 (3)	<ul style="list-style-type: none"> · 프로그래밍이란? 스크래치 소개 · 블록을 연결하여 첫 프로그램 만들기 · 스프라이트 움직이기 · 반복블록, 조건블록 이해하기 · 무대의 좌표 이해하기 · 자동차 게임 만들기
STEAM 주제 학습 (15)	<ul style="list-style-type: none"> · [생물] 생물의 보호책 · [물질] 물의 순환 · [빛] 빛 · [에너지] 신재생에너지 · [우주] 태양계 시뮬레이션
프로젝트 (3)	<ul style="list-style-type: none"> · 개인 프로젝트 준비 · 프로젝트 발표

교육 프로그램은 크게 3단계로 나누어서 진행하였다. <표 IV-57>는 교육 프로그램의 단계별 구성이다. 프로그래밍 경험이 없는 학생들이 스크래치 언어를 학습할 수 있는 프로그래밍 학습 단계, 교육과정 분석을 통해 선정된 주제를 스크래치 프로그래밍을 통해 구현하고 디지털 스토리텔링 하는 STEAM 주제 학습 단계, 교육 프로그램 전 단계를 거쳐 학습한 내용을 바탕으로 자신이 관심 있는 주제를 선정하여 직접 프로그램을 만드는 프로젝트 단계이다.

스크래치 학습 단계에서는 다양한 블록 카테고리를 별도로 학습하는 것이 아니라 간단한 게임 만들기 프로젝트를 진행하면서 프로젝트에 사용되는 블록을 익혀 이후의 STEAM 주제 학습 단계에서 프로그래밍 활동에 도움을 줄 수 있도록 하였다.

STEAM 주제 학습 단계에서는 선정된 주제를 일리노이주 주립대의 CeMaST(Center for Mathematics, Science, and Technology)의 IMaST(Integrated Mathematics, Science, and Technology) 프로젝트에서 사용되었던 순환학습 모형에 맞게 수업을 진행하였다.

프로젝트 단계에서는 주제 학습 단계에서 작성한 프로그램을 업그레이드 하거나 관심 있는 과학 주제를 직접 선정하고 프로그래밍 하는 활동을 통해 전체 교육 프로그램을 마무리하는 시간을 가졌다. 스크래치 학습 단계와 STEAM 주제 학습 단계와는 달리 직접 관심 있는 주제를 선정하고 설계, 제작하는 과정을 거치며 학습자들의 컴퓨팅 사고력과 융합적 소양을 기를 수 있다.

3) 적용 및 효과분석

(1) 연구 가설

① 가설 1

- 연구가설: 스크래치 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 없다.
- 대립가설: 스크래치 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 있다.

② 가설 2

- 연구가설: 스크래치 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 없다.
- 대립가설: 스크래치 프로그래밍 중심의 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 있다.

(2) 연구 대상

본 연구의 실험집단은 제주지역 S초등학교의 스크래치 프로그래밍 교실에 지원한 15명의 학생을 선정하였다. 스크래치 프로그래밍 교실은 방과후학교로 운영되었고 교내 4~6학년 학생 중 수강을 희망하는 학생들을 모집하였다. 실험집단에게는 스크래치를 활용한 프로그래밍 중심의 STEAM 교육 프로그램을 투입하였다.

비교집단은 제주지역 J대학교의 초등창의컴퓨터 교실에 지원한 4~5학년 학생 16명을 선정하였다. 초등창의컴퓨터 교실은 제주지역 초등학생 중 프로그래밍 강의 수강을 희망하는 학생들의 지원을 받아 운영하였고 스크래치 프로그래밍을 통한 게임개발을 학습주제로 수업을 진행하였다. 연구 대상에 대한 구체적인 사항을 <표 IV-58>에 제시하였다.

<표 IV-58> 연구 대상

구분	실험집단(명)			비교집단(명)		
	남	여	소계	남	여	소계
4학년	2	8	10	6	5	11
5학년	0	0	0	2	3	5
6학년	4	0	4	0	0	0
소계	6	8	14	8	8	16

실험집단, 비교집단의 참여자 중 각 1명은 GALT 검사는 실시하였지만 TTCT 검사를 실시하지 못하여 결측값으로 처리하여 통계를 실시하였다.

(3) 연구 절차

실험 연구의 절차를 <표 IV-59>에 제시하였다. 본 실험 연구에서의 스크래

치 중심의 교육 프로그램은 실험집단을 대상으로 하는 7주 간 매주 토요일 3차시씩 총 21차시의 집합 강의 및 실습으로 구성하였는데 실제의 방과후학교는 창의력과 논리적 사고력, 과학과 관련된 정의적 영역에 대한 사전·사후검사를 포함하여 9주로 운영되었다. 교육프로그램의 첫 주차에는 사전검사와 교육 프로그램에 대한 오리엔테이션을 실시하였으며 마지막 주차에는 학생들이 직접 디자인하고 작성한 프로젝트를 발표하고 사후검사하는 시간을 가졌다.

<표 IV-59> 프로그램의 투입 기간

구분	시작주	1주차	2주차	3주차	4주차	5주차	6주차	7주차	최종일
1차시	소개								발표회
2차시									
3차시	사전검사								사후평가

비교집단에는 7주간 총 21차시의 집합 강의 및 실습으로 구성된 스크래치를 활용한 게임 개발을 주제로 강의를 진행하였으며 마찬가지로 첫 주차와 마지막 주차에는 창의력과 논리적 사고력에 대한 사전검사와 사후검사를 실시하였다.

실험 처치 후 실험집단과 비교집단에 교육효과를 살펴보기 위하여 논리적 사고력, 창의력 사후검사를 실시하였으며 본 연구의 설계를 도식화하여 <표 IV-60>에 제시하였다.

<표 IV-60> 실험 설계

	사전검사	처치	사후검사
실험집단	O ₁ , O ₃	X ₁	O ₂ , O ₄
비교집단	O ₁	X ₂	O ₂

O₁ : 논리적 사고력, 창의력 사전검사

O₂ : 논리적 사고력, 창의력 사후검사

O₃ : 과학과 관련된 정의적 영역 사전검사

O₄ : 과학과 관련된 정의적 영역 사후검사

X₁ : 스크래치 프로그래밍 중심 STEAM 교육 프로그램

X₂ : 스크래치를 활용한 게임 개발 교육 프로그램

(4) 연구 결과

① 창의력 변화

- 정규성 검정

창의력 검사 결과의 집단 간, 각 집단 별 비교 전에 두 집단 모두 30명 이하의 집단이기 때문에 정규분포임을 확인하기 위하여 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였다. 실험, 비교집단의 정규성 검정 결과를 각각 <표 IV-61>, <표 IV-62>에 제시하였다.

<표 IV-61> 실험집단 창의력 검사 정규성 검정

하위요소	기술통계(N=13)				Sig.	p
	M	SD	Max	Min		
유창성	99.846	25.357	149	60	.980	.977
독창성	90.154	21.260	130	54	.924	.287
제목의 추상성	97.538	20.691	118	40	.817	.011*
정교성	110.000	21.668	145	72	.943	.497
성급한 종결에 대한 저항	99.154	10.534	121	84	.955	.678
평균	99.308	15.526	124	73	.973	.925
창의성 지수	110.000	17.268	138	82	.963	.804

*p<.05

실험집단의 정규성 검정 결과 제목의 추상성을 제외한 하위요소에서 정규성이 만족된 것으로 나타났다.

<표 IV-62> 비교집단 창의력 검사 정규성 검정

하위요소	기술통계(N=15)				Sig.	p
	M	SD	Max	Min		
유창성	101.533	17.687	135	74	.960	.693
독창성	92.800	14.800	134	75	.860	.025*
제목의 추상성	110.600	38.424	160	40	.917	.175
정교성	108.267	9.691	129	91	.969	.838
성급한 종결에 대한 저항	97.133	10.162	117	75	.979	.959
평균	101.667	12.075	120	78	.941	.401
창의성 지수	115.000	15.464	141	86	.958	.660

*p<.05

비교집단의 정규성 검정 결과 독창성에서 정규성이 만족되지 않았으며 나머지 영역에 대해서는 정규성을 만족하였다.

실험, 비교집단의 정규성 검정 결과에 따라 집단 간, 집단 별 비교 시에 정규성이 확보된 하위요소에 대하여 모수검정을 사용하여 연구결과를 분석하였다.

- 집단 간 비교

<표 IV-63>에서 제시한 것처럼 두 집단 모두 정규성을 확보한 창의력의 하위요소인 유창성, 정교성, 성급한 종결에 대한 저항, 창의력 평균, 창의력 지수에 대한 사전검사를 독립표본 t검정을 통해 비교한 결과 검사를 실시한 모든 영역에서 실험집단, 비교집단이 동질집단인 것으로 나타났다.

<표 IV-63> 창의력 사전검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
유창성	실험	13	99.846	25.357	-.206	.838
	비교	15	101.533	17.687		
정교성	실험	13	110.000	21.668	.266	.793
	비교	15	108.267	9.691		
성급한 종결에 대한 저항	실험	13	99.154	10.534	.516	.610
	비교	15	97.133	10.162		
평균	실험	13	99.308	15.526	-.452	.655
	비교	15	101.667	12.075		
창의성 지수	실험	13	110.000	17.268	-.808	.426
	비교	15	115.000	15.464		

정규성 검정에서 정규분포가 아닌 것으로 나타난 독창성과 제목의 추상성에 대해서는 비모수 통계 방법인 Mann-Whitney U 검증방법을 이용해 두 집단의 창의력 사전검사 결과를 비교하였고 그 결과를 <표 IV-64>에 제시하였다.

<표 IV-64> 창의력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
독창성	실험	13	90.154	21.260	13.81	88.500	.678
	비교	15	92.800	14.800	15.10		
제목의 추상성	실험	13	97.538	20.691	11.81	62.500	.106
	비교	15	110.600	38.424	16.83		

Mann-Whitney U 검증을 실시한 결과 유의확률이 유의수준보다 높게 나타났으며 이는 두 집단 간의 독창성, 제목의 추상성 부분에 대해서도 동질집단임을 알려주고 있다.

실험 처치 후 사후검사 비교에서는 <표 IV-65>에서 제시한 것과 같다. 성급한 종결에 대한 저항에서 실험집단이 비교집단에 비해 통계적으로 유의미한 상승이 있었다.

<표 IV-65> 창의력 사후검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
유창성	실험	13	101.846	16.113	-1.340	.192
	비교	15	111.133	19.971		
정교성	실험	13	116.769	19.833	1.063	.303
	비교	15	110.333	9.788		
성급한 종결에 대한 저항	실험	13	107.923	7.274	2.738	.011*
	비교	15	99.000	9.592		
평균	실험	13	107.846	8.552	.428	.672
	비교	15	106.000	13.342		
창의성 지수	실험	13	121.385	10.500	.354	.726
	비교	15	119.600	15.324		

*p<.05

정규성을 확보하지 못한 독창성과 제목의 추상성에 대해서도 사후검사 결과를 비교하여 <표 IV-66>에 제시하였다. 두 요소 모두 유의확률이 유의수준보다 낮아 두 집단 간 유의미한 차이가 나타나지 않았다.

<표 IV-66> 창의력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
독창성	실험	13	104.077	18.852	14.42	96.500	.963
	비교	15	102.933	18.801	14.57		
제목의 추상성	실험	13	108.538	19.312	14.23	94.000	.872
	비교	15	108.667	37.523	14.73		

창의력 검사에 대한 두 집단 간 사후검사를 비교한 결과 성급한 종결에 대한 저항의 영역에서 실험집단이 비교집단에 비해 유의미한 상승이 나타났다

• 집단 내 비교

이번에는 각 집단별로 사전·사후 결과를 비교해 보았다. 실험집단의 경우 정규성을 확보하지 못한 제목의 추상성을 제외한 영역에 대해 대응표본 t검정을 실시하였으며 그 결과는 <표 IV-67>과 같다.

<표 IV-67> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	99.846	25.357	-.345	.736
	사후	101.846	16.113		
독창성	사전	90.154	21.260	-2.597	.023*
	사후	104.077	18.852		
정교성	사전	110.000	21.668	-.981	.346
	사후	116.769	19.833		
성급한 종결에 대한 저항	사전	99.154	10.534	-2.822	.015*
	사후	107.923	7.274		
평균	사전	99.308	15.526	-2.747	.018*
	사후	107.846	8.552		
창의성 지수	사전	110.000	17.268	-3.188	.008**
	사후	121.385	10.500		

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

비교 결과 제목의 추상성, 성급한 종결에 대한 저항, 평균, 창의성 지수 영역에서 통계적으로 매우 유의미한 상승이 나타났다.

정규분포를 따르지 않는 제목의 추상성에 대해서는 Wilcoxon 부호순위 검정

을 이용하여 사전·사후검사 결과를 비교하여 <표 IV-68>에 제시하였다.

<표 IV-68> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
제목의 추상성	실험	13	사전	97.538	20.691	-.400	.689
			사후	108.538	19.312		

검정 결과 제목의 추상성에 대한 Z 통계값이 -.400, 유의확률이 .689로 유의한 차이가 나타나지 않았다.

비교집단은 정규성을 확보하지 못한 독창성을 제외한 하위요소에 대해서 사전, 사후검사 결과를 비교하였으며 그 결과를 <표 IV-69>에 제시하였다.

<표 IV-69> 비교집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	101.533	17.687	-2.015	.064
	사후	111.133	19.971		
제목의 추상성	사전	110.600	38.424	.235	.817
	사후	108.667	37.523		
정교성	사전	108.267	9.691	-1.269	.225
	사후	110.333	9.788		
성급한 종결에 대한 저항	사전	97.133	10.162	-.784	.446
	사후	99.000	9.592		
평균	사전	101.667	12.075	-1.514	.152
	사후	106.000	13.342		
창의성 지수	사전	115.000	15.464	-1.353	.198
	사후	119.600	15.324		

독창성을 제외한 비교집단의 사전·사후검사 결과 모든 하위요소의 평균 비교에서 유의미한 변화가 나타나지 않아 창의력이 신장되지 않았음을 알 수 있었다.

정규분포를 따르지 않는 비교집단의 독창성 결과에 대해서 Wilcoxon 부호순

위 검정을 실시하여 <표 IV-70>에 제시하였다.

<표 IV-70> 비교집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
독창성	실험	15	사전	97.538	20.691	-1.958	.050*
			사후	108.538	19.312		

*p<.05

비교집단의 독창성에 대한 사전·사후검사 비교 결과 Z값이 -1.958, 유의확률이 .050로서 유의수준 .05에서 유의미한 차이를 보였다.

창의력 검사에 대한 집단 내 사전·사후검사 비교 결과 실험집단에는 독창성, 성급한 종결에 대한 저항, 창의력 평균, 창의성 지수에서 유의미한 점수 상승이 있었고 비교집단에는 독창성에서 유의미한 상승이 있는 것으로 나타났다.

② 논리적 사고력 변화

• 정규성 검정

먼저 프로그램 투입 전 연구대상의 논리적 사고가 정규분포를 갖는지 확인하기 위해 논리적 사고력 사전검사 결과에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-71>와 <표 IV-72>에 제시하였다.

<표 IV-71> 실험집단 논리적 사고력 검사 정규성 검정

하위요소	기술통계(N=14)				Sig.	p
	M	SD	Max	Min		
보존	1.071	0.616	2	0	.779	.003*
비례	0.357	0.497	1	0	.616	.000*
변인통제	0.643	0.633	2	0	.771	.002*
확률	0.143	0.363	1	0	.428	.000*
상관	0.143	0.363	1	0	.428	.000*
조합	0.786	0.699	2	0	.806	.006*
합계	3.143	1.512	6	1	.839	.016**

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

<표 IV-72> 비교집단 논리적 사고력 검사 정규성 검정

하위요소	기술통계(N=16)				Sig.	p
	M	SD	Max	Min		
보존	1.000	0.632	2	0	.787	.002*
비례	0.188	0.403	1	0	.484	.000*
변인통제	0.563	0.814	2	0	.687	.000*
확률	0.250	0.577	2	0	.507	.000*
상관	0.188	0.403	1	0	.484	.000*
조합	1.375	0.719	2	0	.768	.001*
합계	3.563	2.279	9	0	.940	.349

**p<.01

실험집단의 경우 <표 IV-71>에서처럼 모든 하위요소와 논리적 사고력의 합계 결과가 정규분포가 아닌 것으로 나타났다. 비교집단의 경우에도 논리적 사고력 합계를 제외한 모든 영역에서 정규성을 확보하지 못하였으며 이에 두 집단의 논리적 사고력의 변화를 살펴보기 위해 비모수적 통계 방법을 이용한 검정을 실시하였다.

- 집단 간 비교

실험 처치 후 실험집단과 비교집단의 논리적 사고력에 미친 영향을 알아보기 위해 두 집단의 논리적 사고력 사전, 사후결과에 대해 각각 Mann-Whitney U검정을 실시하여 비교하였다. 먼저, 프로그램 투입 전 두 집단이 동질집단임을 확인하기 위해 논리적 사고력 사전검사 결과를 비교하여 <표 IV-73>에서 제시하였다.

사전검사 비교 결과 조합논리에서 평균이 .786인 실험집단에 비해 비교집단의 평균이 1.375로 나타났으며 유의확률이 .032로서 유의수준 .05에서 유의한 차이가 있는 것으로 나타났다. 따라서 사후검사에 대한 집단 간 비교에 대해서는 조합논리를 제외하여 두 집단을 비교하였다.

<표 IV-73> 논리적 사고력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	14	1.071	0.616	15.96	105.500	.752
	비교	16	1.000	0.632	15.09		
비례	실험	14	0.357	0.497	16.86	93.000	.303
	비교	16	0.188	0.403	14.31		
변인통제	실험	14	0.643	0.633	16.43	99.000	.548
	비교	16	0.563	0.814	14.69		
확률	실험	14	0.143	0.363	15.07	106.000	.700
	비교	16	0.250	0.577	15.88		
상관	실험	14	0.143	0.363	15.14	107.000	.748
	비교	16	0.188	0.403	15.81		
조합	실험	14	0.786	0.699	12.07	64.000	.032*
	비교	16	1.375	0.719	18.50		
합계	실험	14	3.143	1.512	14.96	104.500	.750
	비교	16	3.563	2.279	15.97		

*p<.05

두 집단의 사후검사 결과 비교를 <표 IV-74>에 제시하였다. 모든 영역에서 두 집단 간 유의한 차이가 나타나지 않았다.

<표 IV-74> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	14	1.286	0.726	17.36	86.000	.232
	비교	16	1.000	0.632	13.88		
비례	실험	14	0.143	0.363	14.57	99.000	.437
	비교	16	0.313	0.602	16.31		
변인통제	실험	14	0.714	0.726	16.64	96.000	.462
	비교	16	0.563	0.814	14.50		
확률	실험	14	0.429	0.646	16.43	99.000	.486
	비교	16	0.375	0.806	14.69		
상관	실험	14	0.286	0.469	17.29	87.000	.108
	비교	16	0.063	0.250	13.94		
합계	실험	14	4.429	2.243	16.89	92.500	.412
	비교	16	3.875	2.446	14.28		

• 집단 내 비교

이번에는 각 집단 별 사전, 사후검사 결과에 따른 집단 내에서의 변화를 살펴보기 위해 각 집단에 대해 Wilcoxon 부호순위 검정을 실시하였고 그 결과를 <표 IV-75>에 제시하였다.

<표 IV-75> 집단별 논리적 사고력 결과 비교(Wilcoxon 부호순위 검정)

하위요소	집단	N	시기	M	SD	Z	p
보존	실험	14	사전	1.071	0.616	-1.732	.083
			사후	1.000	0.632		
	비교	16	사전	1.286	0.726	.000	1.000
			사후	1.000	0.632		
비례	실험	14	사전	0.357	0.497	-1.732	.083
			사후	0.188	0.403		
	비교	16	사전	0.143	0.363	-1.414	.157
			사후	0.313	0.602		
변인통제	실험	14	사전	0.643	0.633	-.378	.705
			사후	0.563	0.814		
	비교	16	사전	0.714	0.726	.000	1.000
			사후	0.563	0.814		
확률	실험	14	사전	0.143	0.363	-2.000	.046*
			사후	0.250	0.577		
	비교	16	사전	0.429	0.646	-.816	.414
			사후	0.375	0.806		
상관	실험	14	사전	0.143	0.363	-1.000	.317
			사후	0.188	0.403		
	비교	16	사전	0.286	0.469	-1.000	.317
			사후	0.063	0.250		
조합	실험	14	사전	0.786	0.699	-2.598	.009**
			사후	1.375	0.719		
	비교	16	사전	1.571	0.646	-1.732	.083
			사후	1.563	0.629		
합계	실험	14	사전	3.143	1.512	-2.313	.021*
			사후	3.563	2.279		
	비교	16	사전	4.429	2.243	-1.098	.272
			사후	3.875	2.446		

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

분석 결과 실험집단의 경우 전 영역에서 순위점수의 상승이 있었고 그 중에서도 확률, 조합논리와 논리적 사고력 합계 점수가 유의미하게 상승한 것으로 나타났다. 반면 비교집단의 경우 사전·사후검사에 유의미한 변화가 없는 것으로 나타났다.

③ 연구 결과 분석

본 연구에서 개발한 방과후학교에서의 스크래치 프로그래밍 중심의 STEAM 교육 프로그램의 교육적 효과를 살펴보기 위해 창의력, 논리적 사고력에 대한 검사 결과 종합하여 <표 IV-76>에 제시하였다.

<표 IV-76> 실험의 결과 분석

	집단	상승 하위요소	
집단 간 비교	실험집단 우위	창의력	성급한 종결에 대한 저항
		논리적 사고력	-
	비교집단 우위	창의력	-
		논리적 사고력	-
집단 내 비교	실험집단	창의력	독창성, 성급한 종결에 대한 저항 창의력 평균, 창의력 지수
		논리적 사고력	확률논리, 조합논리, 논리적 사고력 합계
	비교집단	창의력	독창성
		논리적 사고력	-

4. 정규 교육과정에서의 스크래치 프로그래밍 중심의 STEAM 교육 프로그램

1) 요구분석

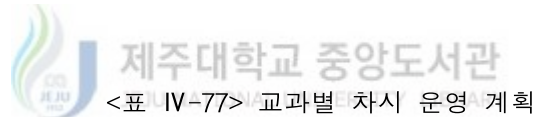
프로그래밍 교육을 활성화하고 학생들 전체가 프로그래밍을 학습하기 위해서는 방과후학교나 교육기부 프로그램이 아닌 정규 교육과정에서 프로그래밍 학습이 이루어져야 할 것이다. 최근 초등학교에는 컴퓨터과학 교과가 없고 창의적 체험활동 등을 통한 컴퓨터과학 교육의 시수 역시 점점 줄어들고 있는 추세이기 때문에 프로그래밍 교육을 실시하기 위한 시수가 절대 부족하다. 본

연구에서는 스크래치 프로그래밍을 통한 STEAM 교육 프로그램을 위하여 과학 글쓰기와 STEAM 프로그래밍을 통합하여 정규 교육과정에서 프로그래밍을 도입하고자 하였다.

2) 설계 및 개발

(1) 교육과정 분석

스크래치 프로그래밍 중심의 STEAM 교육 프로그램을 통해 과학 학습에 대한 흥미를 유발하고 프로그래밍을 학습하기 위하여 디지털 스토리텔링 기반으로 수업 모델을 작성하였다. 2009 개정 과학과 교육과정 중 6학년 2학기 내용을 분석하여 스토리텔링의 학습 주제를 선정하였다 선정된 주제는 6학년 2학기 2단원 여러 가지 기체에서 ‘여러 가지 기체의 쓰임’ 과 3단원 에너지와 도구에서 ‘신재생 에너지’, ‘지구 온난화’이다. 과학 교과를 중심교과로 실과, 미술, 창의적 체험활동을 융합하였다. 교육과정 분석에 따른 교과별 차시 운영 계획은 <표 IV-77>과 같다.



차시	구분	관련 교과(확보 차시)
1-6	스크래치 소양 익히기	실과(4), 미술(1), 창체(1)
7-12	과학주제 디지털 스토리텔링	과학(4), 미술(2)

(2) 주제에 따른 운영 계획

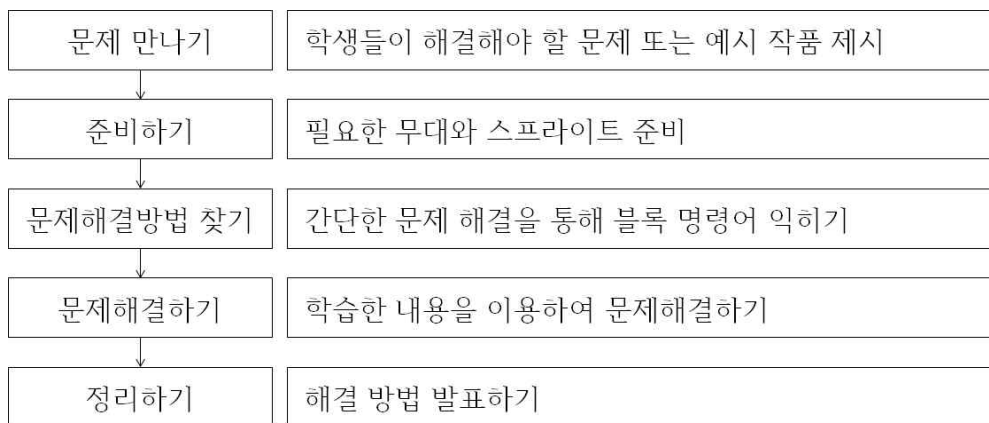
선정된 과학 주제별로 각 2차시 분량의 수업을 계획하여 진행하였다. <표 IV-76>은 그 중 ‘여러 가지 기체의 쓰임’ 주제에 대한 운영 계획이다.

<표 IV-78> 여러 가지 기체의 쓰임 주제 개요

구분	내용
개요	1. 여러 가지 기체의 쓰임 소개 2. 스토리보드를 활용하여 이야기 구성 3. 무대 및 스프라이트 디자인 4. 스크래치 프로그래밍 5. 디지털 스토리텔링(짹, 모뎀, 전체 발표)
목표	1. 여러 가지 기체의 쓰임을 알 수 있다. 2. 경험과 아이디어를 토대로 여러 가지 기체의 쓰임을 주제로 이야기를 구성할 수 있다. 3. 여러 가지 기체의 쓰임을 주제로 자기만의 스크래치 프로그래밍을 하고 디지털 스토리텔링 할 수 있다.
STEAM 교과 요소	S : 여러 가지 기체의 쓰임 알기 T : 스크래치 프로그래밍 하기 E : 스크래치 프로그램 설계하기 A : 무대와 스프라이트 디자인, 디지털 스토리텔링 M : 프로그래밍을 위한 논리, 좌표, 계산식

(3) 교육 프로그램 진행

교육 프로그램은 스크래치 소양 익히기 단계와 과학 디지털 스토리텔링 단계로 구분하였다.

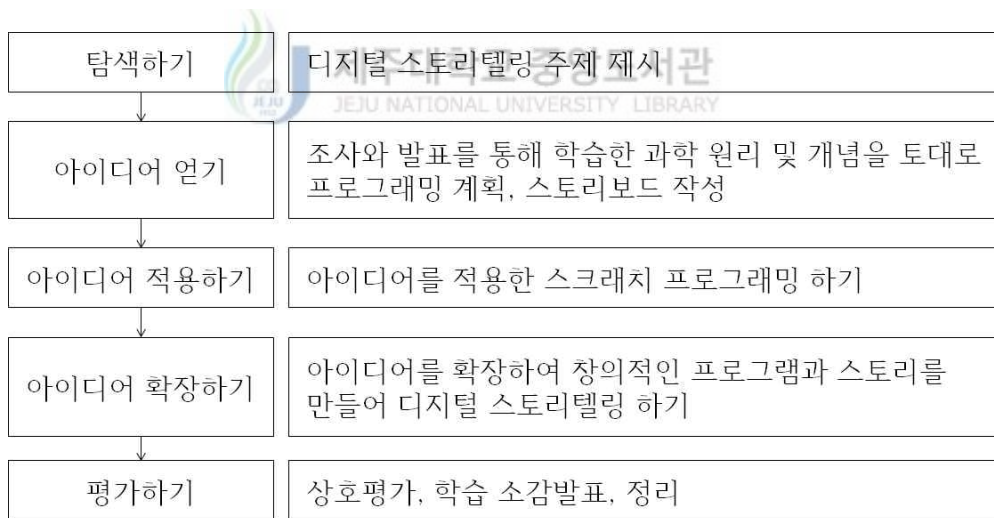


[그림 IV-2] 스크래치 소양 익히기 단계 학습 순서

스크래치 소양 익히기 단계에서는 단순히 교사의 코딩을 따라 하는 활동을

지양하고 학생 스스로 해결해야 할 문제를 제시하고 교사와 함께 학습한 예제 프로그램을 확장시켜 문제를 해결하는 방식으로 진행하였다. 디지털 스토리텔링 단계에서 자신의 이야기를 직접 구상, 설계하여 프로그래밍하기 위해서는 단순한 스크래치 기능의 학습보다는 문제를 해결하기 위한 스크래치 활용이 중요하기 때문이다.

디지털 스토리텔링 단계에서는 과학 수업시간에 학습한 내용을 개인적인 경험과 과학적 아이디어를 첨가한 이야기로 만들고 이를 스크래치로 구현하기 위해 발산적 발문을 통해 다양한 아이디어를 수용하고 이를 스토리보드를 통해 구상하는 시간을 가졌다. 학생들은 각자의 스토리텔링을 위해 직접 스프라이트와 무대를 디자인하여 생성하고 블록을 이용하여 프로그래밍 하였다. 기존의 프로그래밍 교육과는 달리 블록을 활용한 코딩만큼이나 스프라이트나 무대 디자인 역시 강조하였다. 각각의 스프라이트와 무대가 스크래치 프로그래밍에서 뿐만 아니라 과학에 대한 이야기를 구성하는데 핵심이 되기 때문이다.



[그림 IV-3] 디지털 스토리텔링 단계 학습 모형

3) 적용 및 효과 분석

(1) 연구 가설

① 가설 1

- 연구가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을

통한 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 없다.

- 대립가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램에 의한 학습자의 창의력에는 차이가 있다.

② 가설 2

- 연구가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 없다.
- 대립가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램에 의한 학습자의 논리적 사고력에는 차이가 있다.

③ 가설 3

- 연구가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 없다.
- 대립가설: 초등학교 6학년 정규 교육과정에서의 프로그래밍 활동을 통한 STEAM 교육 프로그램에 의한 학습자의 과학과 관련된 정의적 영역에는 차이가 있다.

(2) 연구 대상

제주특별자치도의 S초등학교의 6학년 중 2학급을 각각 실험집단과 비교집단으로 선정하여 본 연구에서 개발한 교육 프로그램의 효과를 살펴보았다. 두 집단 모두 프로그래밍 경험이 있는 학생은 없었으며 <표 IV-79>에 자세한 연구 대상의 대상을 제시하였다.

<표 IV-79> 연구대상

	남	여	합계
실험집단	13	10	24
비교집단	14	11	25

(3) 연구 절차

교육 프로그램 투입 전 실험, 비교집단에게 논리적 사고력, 창의력 사전검사를 실시하여 두 집단이 동질집단임을 확인하였다.

교육 프로그램은 약 9주 간 총 12차시의 내용으로 구성하였고 투입 일정은 <표 IV-80>에 제시하였다.

<표 IV-80> 프로그램 투입 일정

차시	구분	학습 주제	학습 내용
1-2	스크래치 소양 익히기	스크래치 만나기	메뉴 익히기, 동작블록
3-4		간단한 게임 만들기	제어·관찰블록
5-6		미디어 아트, 점수를 사용한 게임	난수, 변수 펜·소리·연산·형태블록
7-8	과학주제 디지털 스토리텔링	여러 가지 기체의 쓰임	예제학습, 개인별 프로젝트
9-10		신재생 에너지	
11-12		지구온난화	

실험 처치 후 실험집단과 비교집단에 교육효과를 살펴보기 위하여 논리적 사고력, 창의력 사후검사를 실시하였으며 본 연구의 설계를 도식화 하여 <표 IV-81>에 제시하였다.

<표 IV-81> 실험 설계

	사전검사	처치	사후검사
실험집단	O ₁ , O ₃	X ₁	O ₂ , O ₄
비교집단	O ₁	X ₂	O ₂

- O₁ : 논리적 사고력, 창의력 사전검사
- O₂ : 논리적 사고력, 창의력 사후검사
- O₃ : 과학과 관련된 정의적 영역 사전검사
- O₄ : 과학과 관련된 정의적 영역 사후검사
- X₁ : 프로그래밍 중심 STEAM 교육 프로그램
- X₂ : 일반적인 교과 수업

(4) 연구 결과

① 창의력 변화

- 정규성 검정

창의력 검사 결과의 집단 간, 각 집단 별 비교 전에 두 집단 모두 30명 이하의 집단이기 때문에 정규분포임을 확인하기 위하여 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였다. 실험, 비교집단의 정규성 검정 결과를 각각 <표 IV-82>, <표 IV-83>에 제시하였다.

<표 IV-82> 실험집단 창의력 검사 정규성 검정

하위요소	기술통계(N=24)				Sig.	p
	M	SD	Max	Min		
유창성	99.8	16.8	128	61	.974	.762
독창성	97.9	14.5	125	73	.957	.383
제목의 추상성	108.8	19.5	145	63	.973	.732
정교성	99.1	16.7	134	79	.894	.016*
성급한 종결에 대한 저항	101.7	12.3	118	70	.926	.079
평균	101.3	11.6	127	70	.969	.631
창의성 지수	112.0	13.9	138	77	.953	.310

*p<.05

실험집단의 정규성 검정 결과 정교성을 제외한 하위요소에서 정규성이 만족된 것으로 나타났다.

<표 IV-83> 비교집단 창의력 검사 정규성 검정

하위요소	기술통계(N=25)				Sig.	p
	M	SD	Max	Min		
유창성	102.8	20.7	149	77	.836	.001**
독창성	96.8	15.0	138	77	.891	.012*
제목의 추상성	111.6	23.3	152	63	.973	.714
정교성	119.1	21.1	150	74	.927	.073
성급한 종결에 대한 저항	102.8	18.1	129	59	.931	.093
평균	106.6	13.7	130	73	.963	.469
창의성 지수	119.0	17.7	151	76	.974	.753

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

비교집단의 정규성 검정 결과 유창성과 독창성에서 정규성이 만족되지 않았으며 나머지 영역에 대해서는 정규성을 만족하였다.

실험, 비교집단의 정규성 검정 결과에 따라 집단 간, 집단 별 비교 시에 정규성이 확보된 하위요소에 대하여 모수검정을 사용하여 연구결과를 분석하였다.

• 집단 간 비교

<표 IV-84>에서 제시한 것처럼 두 집단 모두 정규성을 확보한 창의력의 하위요소인 제목의 추상성, 성급한 종결에 대한 저항, 창의력 평균, 창의력 지수에 대한 사전검사를 독립표본 t검정을 통해 비교하였다.

두 집단 중 어느 하나라도 정규분포를 따르지 않는 유창성, 독창성, 정교성에 대해서는 Mann-Whitney U검정을 이용하여 분석하였고 그 결과를 <표 IV-85>에 제시하였다

<표 IV-84> 창의력 사전검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
제목의 추상성	실험	24	108.833	19.450	-.451	.654
	비교	25	111.600	23.261		
성급한 종결에 대한 저항	실험	24	101.667	12.324	-.264	.793
	비교	25	102.840	18.140		
창의력 평균	실험	24	101.333	11.601	-1.448	.154
	비교	25	106.600	13.720		
창의성 지수	실험	24	112.042	13.877	-1.533	.132
	비교	25	119.040	17.747		

<표 IV-85> 창의력 사전검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
유창성	실험	24	99.792	16.772	25.38	291.000	.857
	비교	25	102.760	20.727	24.64		
독창성	실험	24	97.875	14.540	26.35	267.500	.515
	비교	25	96.800	15.017	23.70		
정교성	실험	24	99.083	16.710	18.08	134.000	.001**
	비교	25	119.120	21.100	31.64		

**p<.01

두 집단의 창의력 사전검사를 비교한 결과 정교성의 영역에 대해서는 유의수준 .05에서 통계적으로 유의한 것으로 나타나 두 집단이 이질 집단인 것으로 나타났다.

실험 처치 후 사후검사 비교에서는 <표 IV-86>, <표 IV-87>와에서 제시한 것과 같다. 제목의 추상성, 성급한 종결에 대한 저항, 평균, 창의성 지수 영역에서 비교집단에 비해 실험집단의 평균상승이 두드러지게 나타났으나 두 집단은 통계적으로 동질집단인 것으로 나타났다.

<표 IV-86> 창의력 사후검사 결과 비교(독립표본 t검정)

하위요소	집단	N	M	SD	t	p
제목의 추상성	실험	24	124.625	17.305	2.004	.052
	비교	25	111.000	29.040		
성급한 종결에 대한 저항	실험	24	108.833	13.736	.733	.467
	비교	25	105.880	14.423		
창의력 평균	실험	24	110.208	10.525	.490	.627
	비교	25	108.160	17.948		
창의성 지수	실험	24	124.750	11.471	.886	.381
	비교	25	120.400	21.583		

<표 IV-87> 창의력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
유창성	실험	24	100.583	17.683	22.06	229.500	.158
	비교	25	110.200	24.813	27.82		
독창성	실험	24	103.292	18.579	23.19	256.500	.384
	비교	25	108.920	24.520	26.74		
정교성	실험	24	113.792	19.706	27.88	231.000	.165
	비교	25	105.240	23.022	22.24		

창의력 사후검사 결과를 비교한 결과 창의력 모든 영역에서 두 집단간 유의미한 차이가 나타나지 않았다. 다만 사전검사에서 비교집단이 통계적으로 유의미하게 높은 점수를 받았던 정교성 부분에서 두 집단이 동질집단이 되었다는 것은 실험집단의 정교성 점수 상승 때문에 생긴 것이 아닌가에 대한 의문을

확인할 필요가 있었다.

- 집단 내 비교

교육 프로그램이 학습자의 창의력에 미치는 영향을 심도있게 살펴보고자 각 집단 별로 사전·사후결과를 분석·비교하였다. 실험집단의 경우 정규분포를 따르지 않는 정교성을 제외한 영역에 대해 대응표본 t검정을 실시하였으며 그 결과는 <표 IV-88>과 같다. 정교성에 대해서는 Wilcoxon 부호순위 검정을 실시하여 그 결과를 <표 IV-89>에 제시하였다.

<표 IV-88> 실험집단 창의력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
유창성	사전	99.792	16.772	-.204	.840
	사후	100.583	17.683		
독창성	사전	97.875	14.540	-1.569	.130
	사후	103.292	18.579		
제목의 추상성	사전	108.833	19.450	-4.470	.000**
	사후	124.625	17.305		
성급한 종결에 대한 저항	사전	101.667	12.324	-3.299	.003**
	사후	108.833	13.736		
창의력 평균	사전	101.333	11.601	-4.808	.000**
	사후	110.208	10.525		
창의성 지수	사전	112.042	13.877	-6.150	.000**
	사후	124.750	11.471		

**p<.01

<표 IV-89> 실험집단 창의력 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
정교성	24	사전	99.083	16.710	-3.654	.000**
		사후	113.792	19.706		

**p<.01

실험집단의 창의력 사전·사후검사를 비교한 결과 제목의 추상성, 성급한 종결에 대한 저항, 창의력 평균, 창의성 지수 영역에서 통계적으로 매우 유의미한 상승이 나타났다. 그리고 정교성에서 유의미한 상승이 있었음을 알 수 있었다.

비교집단은 정규성을 확보하지 못한 유창성과 독창성을 제외한 하위요소에 대해서 대응표본 t검정을 실시하여 사전, 사후검사 결과를 비교하였으며 유창성과 독창성은 Wilcoxon 부호순위 검정을 실시하여 분석하였고 그 결과를 <표 IV-90>와 <표 IV-91>에 제시하였다.

<표 IV-90> 비교집단 창의적 사고력 사전·사후 결과 비교(대응표본 t검정)

하위요소	시기	M	SD	t	p
제목의 추상성	사전	111.600	23.261	.130	.898
	사후	111.000	29.040		
정교성	사전	119.120	21.100	4.137	.000**
	사후	105.240	23.022		
성급한 종결에 대한 저항	사전	102.840	18.140	-1.114	.276
	사후	105.880	14.423		
평균	사전	106.600	13.720	-.555	.584
	사후	108.160	17.948		
창의성 지수	사전	119.040	17.747	-.427	.673
	사후	120.400	21.583		

**p<.01

<표 IV-91> 비교집단 창의적 사전·사후검사 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
유창성	25	사전	102.760	20.727	-1.413	.158
		사후	110.200	24.813		
독창성	25	사전	96.800	15.017	-2.665	.008**
		사후	108.920	24.520		

**p<.01

비교집단의 경우 독창성과 정교성 영역 점수에서 유의미한 차이가 나타났다. 독창성의 경우 점수가 상승한 반면 정교성의 경우 오히려 유의미한 하락을 한 것으로 나타났다. 그 원인으로는 유창성과 독창성 부분의 점수가 많이 향상된 것으로 볼 때 비교집단 학생들은 창의력의 하위 요소 중 유창성과 독창성 부분에 치중한 문제해결이 이루어졌기 때문에 상대적으로 각 도형의 정교성 부분에서는 시간이나 노력에 대한 집중이 부족했던 것으로 판단된다.

② 논리적 사고력 변화

- 정규성 검정

논리적 사고력 검사 역시 먼저 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-92>, <표 IV-93>에 제시하였다.

<표 IV-92> 실험집단 논리적 사고력 검사 정규성 검정

하위 요소	기술통계(N=24)				통계량	유의도
	M	SD	Max	Min		
보존	1.125	0.680	2	0	.800	.000**
비례	0.417	0.584	2	0	.681	.000**
변인통제	0.542	0.658	2	0	.737	.000**
확률	0.750	0.847	2	0	.751	.000**
상관	0.208	0.415	1	0	.503	.000**
조합	1.167	0.702	2	0	.802	.000**
합계	4.208	2.043	8	1	.933	.116

**p<.01

<표 IV-93> 비교집단 논리적 사고력 검사 정규성 검정

하위 요소	기술통계(N=25)				통계량	유의도
	M	SD	Max	Min		
보존	1.280	0.678	2	0	.785	.000**
비례	0.720	0.737	2	0	.785	.000**
변인통제	0.720	0.737	2	0	.785	.000**
확률	0.840	0.850	2	0	.773	.000**
상관	0.120	0.332	1	0	.384	.000**
조합	1.320	0.690	2	0	.778	.000**
합계	5.000	2.398	9	1	.938	.136

**p<.01

두 집단 모두 논리적 사고력 합계는 정규분포를 따르지만 논리적 사고력 각 하위요소에 대해서는 정규성을 만족하지 않아 논리적 사고력 합계에는 모수통계 나머지 영역에 대해서는 비모수 통계방법을 사용하여 집단 간·집단 내 비교를 실시하였다.

• 집단 간 비교

본 연구에서 개발한 교육 프로그램이 실험집단의 논리적 사고력에 미친 영향을 알아보기 위해 프로그램 투입 전 실험집단과 비교집단의 논리적 사고력에 대한 동질집단임을 확인하였다. 논리적 사고력 합계에 대해서는 두 집단의 점수를 독립표본 t검정으로 분석하였고 나머지 하위요소에 대해서는 Mann-Whitney U검정을 실시하였다. 두 집단의 사전검사 결과를 비교하여 <표 IV-94>과 <표 IV-95>에 제시하였다.

<표 IV-94> 논리적 사고력 사전검사 결과 비교(독립표본 t검정)

하위요소	시기	집단	N	M	SD	t	p
논리적 사고력 합계	사전	실험	24	4.208	2.043	-1.246	.221
		비교	25	5.000	2.398		

<표 IV-95> 논리적 사고력 사전 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	24	1.125	0.680	23.46	263.000	.415
	비교	25	1.280	0.678	26.48		
비례	실험	24	0.417	0.584	22.23	233.500	.137
	비교	25	0.720	0.737	27.66		
변인통제	실험	24	0.542	0.658	23.40	261.500	.396
	비교	25	0.720	0.737	26.54		
확률	실험	24	0.750	0.847	24.25	282.000	.698
	비교	25	0.840	0.850	25.72		
상관	실험	24	0.208	0.415	26.10	273.500	.408
	비교	25	0.120	0.332	23.94		
조합	실험	24	1.167	0.702	23.50	264.000	.431
	비교	25	1.320	0.690	26.44		

두 집단의 논리적 사고력 사전검사를 비교한 결과 통계적으로 유의한 차이가 나타나지 않았으며 따라서 두 집단이 동질집단임을 알 수 있었다.

실험 처치 후 두 집단의 변화를 알아보기 위해 사전검사와 마찬가지로 사후검사에 대한 두 집단의 결과를 비교하여 표에 <표 IV-96>과 <표 IV-97>에 제시하였다.

<표 IV-96> 논리적 사고력 사후검사 결과 비교(독립표본 t검정)

하위요소	시기	집단	N	M	SD	t	p
논리적 사고력 합계	사전	실험	24	4.208	2.043	-.355	.724
		비교	25	5.000	2.398		

<표 IV-97> 논리적 사고력 사후검사 결과 비교(Mann-Whitney U검정)

하위요소	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	실험	24	1.375	0.647	23.73	269.500	.490
	비교	25	1.520	0.510	26.22		
비례	실험	24	0.708	0.751	24.58	290.000	.829
	비교	25	0.760	0.779	25.40		
변인통제	실험	24	0.625	0.711	23.88	273.000	.556
	비교	25	0.760	0.779	26.08		
확률	실험	24	1.000	0.885	25.63	285.000	.748
	비교	25	0.920	0.909	24.40		
상관	실험	24	0.042	0.204	23.48	263.500	.164
	비교	25	0.240	0.597	26.46		
조합	실험	24	1.583	0.584	27.00	252.000	.278
	비교	25	1.320	0.802	23.08		

두 집단의 사후검사를 비교한 결과 두 집단에서 어떠한 유의한 차이가 나타나지 않았다. 좀 더 세밀한 분석을 위해 집단 내 비교를 통해 논리적 사고력에 미친 영향을 살펴보고자 하였다.

- 집단 내 비교

각 집단에 실시한 교육 처치가 학습자의 논리적 사고력에 끼친 영향을 살펴

보기 위해 각 집단 별로 사전·사후검사 결과를 비교하였다. 두 집단 모두 정규분포를 따르는 논리적 사고력 합계에 대해서는 대응표본 t검정을 실시하였고 나머지 논리적 사고력 하위요소에 대해서는 Wilcoxon 부호순위 검정을 실시하여 분석하였다.

먼저 실험집단의 분석 결과를 <표 IV-98>, <표 IV-99>에 제시하였다.

<표 IV-98> 실험집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	N	시기	M	SD	t	p
논리적 사고력 합계	24	사전	4.208	2.043	-2.991	.007**
		사후	5.250	2.674		

**p<.01

<표 IV-99> 실험집단 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
보존	24	사전	1.125	0.680	-1.255	.210
		사후	1.375	0.647		
비례	24	사전	0.417	0.584	-2.111	.035*
		사후	0.708	0.751		
변인통제	24	사전	0.542	0.658	-.707	.480
		사후	0.625	0.711		
확률	24	사전	0.750	0.847	-1.732	.083
		사후	1.000	0.885		
상관	24	사전	0.208	0.415	-2.000	.046*
		사후	0.042	0.204		
조합	24	사전	1.167	0.702	-2.140	.032*
		사후	1.583	0.584		

*p<.05

실험집단의 논리적 사고력 사전·사후검사를 비교한 결과 논리적 사고력 합계, 비례, 조합논리에 대해서는 사전에 비해 사후에 통계적으로 유의미한 상승이 있는 것으로 나타난 반면 상관논리에 대해서는 오히려 통계적으로 유의미

한 하락이 있는 것으로 나타났다.

비교집단에도 같은 방법으로 사전·사후 논리적 사고력 검사 결과를 비교하여 <표 IV-100>, <표 IV-101>에 제시하였다.

<표 IV-100> 비교집단 논리적 사고력 사전·사후검사 결과 비교(대응표본 t검정)

하위요소	N	시기	M	SD	t	p
논리적 사고력 합계	25	사전	5.000	2.398	-1.797	.085
		사후	5.520	2.648		

<표 IV-101> 비교집단 논리적 사고력 사전·사후 결과 비교(Wilcoxon 부호순위 검정)

하위요소	N	시기	M	SD	Z	p
보존	25	사전	1.280	0.678	-1.897	.058
		사후	1.520	0.510		
비례	25	사전	0.720	0.737	-.187	.851
		사후	0.760	0.779		
변인통제	25	사전	0.720	0.737	-.302	.763
		사후	0.760	0.779		
확률	25	사전	0.840	0.850	-.577	.564
		사후	0.920	0.909		
상관	25	사전	0.120	0.332	-1.732	.083
		사후	0.240	0.597		
조합	25	사전	1.320	0.690	.000	1.000
		사후	1.320	0.802		

비교집단의 논리적 사고력 사전·사후검사를 비교한 결과 모든 영역에서 통계적으로 유의미한 차이가 없는 것으로 나타났다.

③ 과학과 관련된 정의적 영역

6학년 정규 교육과정에서 스크래치 프로그래밍 중심의 STEAM 교육 프로그램이 실험집단의 과학과 관련된 정의적 영역에 어떤 영역을 미쳤는지 알아보기 위하여 먼저 사전검사에 대하여 Shapiro-Wilks 정규성 검정을 실시하였고 그 결과를 <표 IV-102>에 제시하였다. 검사지 중 일부 질문에 응답을 하지 않

은 학습지에 대해서는 결측값으로 처리하였고 총 22명의 학습자를 대상으로 연구 결과를 분석하였다.

<표 IV-102> 과학과 관련된 정의적 영역 검사 정규성 검정

하위 요소	기술통계(N=22)				통계량	유의도
	M	SD	Max	Min		
인식	3.681	0.492	4.580	2.830	.951	.328
흥미	3.391	0.771	4.400	1.870	.933	.142
태도	3.556	0.633	4.900	2.290	.958	.455
평균	3.536	0.566	4.580	2.350	.987	.989

정규성 검정 결과 과학과 관련된 인식, 흥미, 태도와 평균 점수에 대한 유의확률이 모두 유의수준인 .05보다 크기 때문에 귀무가설이 채택되어 정규분포임을 확인하였다.

본 연구에서 개발한 프로그램이 과학과 관련된 정의적 특성에 미친 영향을 알아보기 위해 사전·사후검사 결과를 분석하여 <표 IV-103>에 제시하였다.

서 제시하였듯이 과학에 대한 인식, 흥미, 태도, 평균의 모든 측면에서 점수가 상승하였지만 모든 영역의 유의확률이 유의수준인 .05보다 크기 때문에 통계적으로는 유의미 하지 않은 것으로 나타났다.

<표 IV-103> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교

하위 요소	시기	N	M	SD	t	유의도
인식	사전	22	3.681	0.492	-1.905	.071
	사후	22	3.895	0.768		
흥미	사전	22	3.391	0.771	-1.588	.127
	사후	22	3.603	0.876		
태도	사전	22	3.556	0.633	-.555	.585
	사후	22	3.635	0.829		
평균	사전	22	3.536	0.566	-1.302	.207
	사후	22	3.690	0.768		

과학과 관련된 정의적 영역 사전·사후검사를 비교한 결과 과학에 대한 인식, 흥미, 태도, 평균이 모두 상승하였지만 통계적으로 유의하지 않았다.

좀더 세밀한 분석을 위해 과학에 대한 정의적 검사지의 세부 요소에 대한 사전·사후검사를 비교하였으며 그 결과는 <표 IV-104>에 제시하였다.

<표 IV-104> 과학과 관련된 정의적 영역 사전·사후검사 결과 비교(세부요소)

하위 요소	시기	N	M	SD	t	유의도
cs	사전	22	3.424	0.592	-2.058	.052
	사후	22	3.788	0.827		
cl	사전	22	3.835	0.775	-.813	.426
	사후	22	3.971	0.890		
cc	사전	22	3.620	0.670	-3.384	.003**
	사후	22	4.076	0.823		
ct	사전	22	3.788	0.787	.211	.835
	사후	22	3.743	0.919		
is	사전	22	3.484	0.935	-.330	.745
	사후	22	3.545	0.935		
il	사전	22	3.606	0.740	-.454	.654
	사후	22	3.682	1.042		
ia	사전	22	3.015	0.893	-1.874	.075
	사후	22	3.395	1.036		
ic	사전	22	2.985	0.979	-2.403	.026*
	사후	22	3.439	0.988		
ix	사전	22	3.545	0.877	-2.334	.030*
	사후	22	3.955	0.956		
au	사전	22	3.728	0.740	.892	.382
	사후	22	3.591	0.958		
ap	사전	22	3.455	0.629	-2.239	.036*
	사후	22	3.757	0.750		
ar	사전	22	3.378	0.758	-1.291	.211
	사후	22	3.620	0.928		
ao	사전	22	3.727	0.739	-.067	.947
	사후	22	3.743	0.965		
av	사전	22	3.531	0.801	-.539	.595
	사후	22	3.621	0.898		
ae	사전	22	3.485	0.833	-.817	.423
	사후	22	3.620	0.905		
ac	사전	22	3.454	0.781	-.290	.775
	사후	22	3.500	0.941		

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

과학의 정의적 영역의 세부 요소에 대하여 실험집단의 사전·사후검사를 비

교한 결과 CC(과학과 관련된 직업에 대한 인식), IC(과학과 관련된 직업에 대한 흥미), IX(과학 불안), AP(개방성)의 세부 요소에서 통계적으로 유의미한 변화가 나타났다. 특히 과학과 관련된 직업에 대한 인식이나 흥미에 대한 점수가 높아진 점은 STEAM 교과에서 추구하는 STEAM 영역에 대한 경쟁력이나 우수 인재 양성이라는 목적에도 부합한다고 할 수 있다.

④ 연구 결과 분석

본 연구에서 개발한 정규교육과정에서의 스크래치 프로그래밍 중심의 STEAM 교육 프로그램의 교육적 효과를 살펴보기 위해 창의력, 논리적 사고력, 과학의 정의적 영역에 대한 검사 결과 종합하여 <표 IV-105>에 제시하였다.

<표 IV-105> 실험의 결과 분석

	집단	상승 하위요소	
집단 간 비교	실험집단 우위	창의력	-
		논리적 사고력	-
	비교집단 우위	창의력	-
		논리적 사고력	-
집단 내 비교	실험집단	창의력	제목의 추상성, 성급한 종결에 대한 저항, 창의력 평균, 창의력 지수, 정교성
		논리적 사고력	비례논리, 상관논리, 조합논리, 논리적 사고력 합계
		과학의 정의적 영역	과학과 관련된 직업에 대한 인식, 과학과 관련된 직업에 대한 흥미, 과학 불안, 개방성
	비교집단	창의력	-
		논리적 사고력	-

V. 결론 및 제언

본 연구에서는 초등학생을 대상으로 컴퓨팅 사고력을 신장시키기 위하여 프로그래밍 중심의 STEAM 교육 프로그램을 제안하고 ADDIE 모형의 개발 단계에 따라 교육 프로그램을 개발 및 적용하였다.

요구 분석은 문헌 연구, 선행연구 분석, 학습자 요구 분석을 수행하였다. 먼저 컴퓨팅 사고력 및 STEAM 교육, 프로그래밍 교육의 이론적 고찰을 통해 문헌 연구를 실시하였다. 다음으로 기 개발된 STEAM 교육 프로그램, 프로그래밍 기반 STEAM 교육 프로그램의 사례를 분석하였다. 마지막으로 학습자 요구 분석은 Rossett의 모형 절차를 따라 초등학생들을 대상으로 설문을 통한 조사연구를 중심으로 수행하였다. 요구 분석 결과 다음과 같은 내용을 추출하였다.

첫째, 일반적인 초등학생 학습자의 경우 프로그래밍 경험이 없는 경우가 대부분이므로 STEAM 교육 프로그램에 활용할 프로그래밍 도구로 학습자의 수준에 적합한 교육용 프로그래밍 언어의 선정이 필요하다. 또한, 학습자의 프로그래밍 학습을 위한 별도의 학습 단계나 시스템을 지원할 필요가 있다.

둘째, STEAM 교육 프로그램을 구성함에 있어 학습에 대한 흥미와 자신감을 유지할 수 있도록 지식이나 개념 중심보다는 학생들의 직접적인 조작하고 체험할 수 있도록 구성할 필요가 있다.

셋째, 학습자가 STEAM 교육에 대한 경험이 적고 초등학생임을 감안하여 STEAM 교육 프로그램에서 다루는 융합의 수준을 적절하게 조절할 필요가 있다. 처음부터 어렵고 복잡한 문제를 해결하기 보다는 간단하고 쉬운 문제를 제시하여 학습 흥미와 자신감을 신장시킬 수 있도록 융합 수준을 고려해야 한다.

요구 분석 내용을 바탕으로 프로그래밍 중심 STEAM 교육 프로그램을 설계 및 개발하였다.

설계 단계에서는 STEAM 교육 프로그램의 주제 선정을 위해 교육과정을 분석하였고 STEAM 교과 요소의 융합을 위해 학습준거와 통합유형을 선정하였

다. 또한 STEAM 교육 프로그램을 수업에 투입하기 위한 수업 모형으로 순환 학습 모형을 선정하였다.

개발 단계에서 STEAM 교육 프로그램 개발을 위한 학습 주제 및 내용을 선정하였다. 프로그래밍 중심 STEAM 교육 프로그램의 교육적 효과를 높이기 위해 프로그래밍 학습 단계, STEAM 주제 학습 단계, 프로젝트 단계를 선정하여 개발하였다.

개발한 STEAM 교육 프로그램은 학습에 사용된 교육용 프로그래밍 언어와 투입방법 등에 따라 4가지로 구분하여 초등학생을 대상으로 적용하였다. 4가지 교육 프로그램은 코딩 프로그래밍 중심 STEAM 교육 프로그램, 교육기부를 위한 센서 기반 스크래치 프로그래밍 중심 STEAM 교육 프로그램, 방과후학교를 위한 스크래치 프로그래밍 중심 STEAM 교육 프로그램, 정규 교육과정에서의 스크래치 프로그래밍 중심 STEAM 교육 프로그램이다. 각 프로그램을 적용한 후에 학습 효과를 분석하였다. 컴퓨팅 사고력을 측정하기 위한 도구로 창의력 검사 도구인 TTCT 도형 A형 검사지와 논리적 사고력 검사 도구로 GALT 검사지 축소본(12문항)을 사용하였고, 융합적 소양을 측정하기 위한 검사 도구로 과학과 관련된 정의적 영역 검사지를 사용하였다.

평가 단계에서 분석한 교육 프로그램의 학습 효과는 다음과 같다.

첫째, 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램이 초등학생의 컴퓨팅 사고력 신장에 효과적인 것으로 나타났다. 4가지 교육 프로그램을 학습한 각각의 실험집단 모두에서 창의력과 논리적 사고력에서 유의미한 점수 상승이 있었다. 특히 비교집단에 투입한 프로그래밍 언어를 활용한 게임 개발 프로그램보다 창의력과 논리적 사고력이 향상되었다는 점에서 본 연구에서 개발한 교육 프로그램이 일반적인 프로그래밍 교육에 비해 컴퓨팅 사고력 신장에 효과적이라고 할 수 있다.

둘째, 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램이 학습자의 융합적 소양 신장에 효과적이다. 코딩 프로그래밍 중심 STEAM 교육 프로그램을 제외하고 스크래치 프로그래밍 중심 STEAM 교육 프로그램에서는 과학과 관련된 정의적 영역에서 유의미한 상승이 있는 것으로 나타났다. 이는 본 연구에서 개발한 교육 프로그램이 과학에 대한 흥미와 자신감을 향상시키

려는 STEAM 교육의 목표를 달성하였음을 보여준다.

본 연구의 시사점은 다음과 같다.

첫째, 프로그래밍 중심 STEAM 교육 프로그램이 컴퓨팅 사고력 향상에 효과적이다. 과학 주제를 학습하고 이를 추상화하여 프로그램으로 자동화하는 과정에서 학습자는 컴퓨팅 사고력을 경험하게 된다. 기존의 프로그래밍 교육이 프로그래밍 스킬이나 그를 통한 컴퓨팅 사고 증진에 초점을 두었다면 본 연구에서는 STEAM 교육을 통해 과학 주제를 학습하고 그 학습내용을 바탕으로 프로그램을 설계하며 이를 프로그래밍을 통해 구현하는 과정에서 융합적인 소양을 쌓는 과정을 추가하였다. 과학 주제를 학습하는 과정에서 프로그래밍을 활용하여 문제를 해결하는 경험을 통해 컴퓨팅 사고력과 융합적 소양을 기를 수 있다. 또한 이는 교육현장에서 프로그래밍과 컴퓨터과학 교육을 자연스럽게 도입할 수 있는 방법적인 측면에서도 기여할 수 있을 것이다.

둘째, 스크래치나 코듀 등 교육용 프로그래밍 언어를 활용한 프로그래밍 활동이 STEAM 교육 프로그램의 주요 내용 및 도구로 사용할 수 있음을 제시하였다. 프로그래밍은 컴퓨팅 사고력을 바탕으로 이루어지며 이런 컴퓨팅 사고력은 컴퓨터과학 만이 아니라 과학, 수학 등의 다른 교과에도 효과적으로 사용될 수 있다. 또한 프로그래밍 활동이 갖는 창의적이고 교과 융합적인 특징이 예술 (Arts) 분야와 자연스럽게 연계될 수 있는 사례를 보여줌으로써 프로그래밍의 교육적 활용 범위를 확장시키는데 기여할 수 있다.

셋째, 프로그래밍 중심 STEAM 교육을 위한 교수 전략으로 교육용 프로그래밍 언어를 통한 디지털 스토리텔링, 피지컬 컴퓨팅, 로보틱스 등을 제시하였다. 각 전략을 활용한 STEAM 교육 프로그램이 충분한 교육적 효과를 가졌음이 검증되었으므로 앞으로 관련 분야의 후속 연구에도 참고가 될 수 있을 것이다.

본 연구를 수행하면서 얻어진 결과를 토대로 다음과 같은 제언을 하고자 한다.

첫째, 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램은 4가지로 각각의 교육적 효과를 검증하기 위해 총 4회에 걸쳐 실험집단에 투입하였다. 하지만 현실적인 문제 등으로 각 교육 프로그램의 실험집단이 30명을 넘지

않는 소집단으로 이루어졌기 때문에 실험결과로 얻은 데이터들의 상관관계를 확인하기 어렵고 일반화 하는 데에도 한계가 있다. 추후에도 다수의 학습자들을 대상으로 교육 프로그램을 투입하여 프로그래밍 중심 STEAM 교육 프로그램의 효과와 컴퓨팅 사고력, 융합적 소양에 영향을 주는 여러 변수들에 대한 분석이 필요할 것이다.

둘째, 컴퓨팅 사고력에 대한 연구가 많이 진행되고 있지만 학습자들의 컴퓨팅 사고력을 검사하기 위한 타당도와 신뢰도를 갖춘 검사도구가 없는 실정이다. 앞으로 컴퓨터과학 교육의 핵심인 컴퓨팅 사고력 관련 연구의 효과를 검증할 수 있도록 컴퓨팅 사고력을 측정할 수 있는 검사지 개발에 대한 연구가 진행되어야 할 것이다.

셋째, 창의력, 논리적 사고력, 과학과 관련된 정의적 영역 검사지를 통해 본 연구에서 개발한 프로그래밍 중심 STEAM 교육 프로그램이 컴퓨팅 사고력과 융합적 소양 신장에 효과가 있음을 증명하였다. 하지만 교육 프로그램의 어떤 요소가 사고력의 어떤 요소의 신장에 관여하였는지에 대한 상관관계를 제시하지는 못하였다. 추후에는 프로그래밍 중심의 STEAM 교육 프로그램과 학습자의 사고력 간의 상관관계에 대한 연구가 진행될 필요가 있다.

마지막으로, 교사와 학생을 위한 다양한 프로그래밍 교육, STEAM 교육 프로그램이 개발되어야 한다. 아직도 대부분의 교사들에게 STEAM 교육과 프로그래밍 교육은 어렵게 느껴지고 있다. 교육부의 발표에 따르면 2015년까지 초·중등 SW교육이 확대되어 모든 학생들에게 SW 개발 교육이 이루어진다. 내실 있는 교육 프로그램을 통해 모든 학생이 컴퓨팅 사고력을 활용한 문제해결의 경험을 할 수 있는 기반이 마련되어야 할 것이다. 이를 위해서는 교사가 쉽게 지도할 수 있고 학생이 재미있게 학습할 수 있는 교육 프로그램이 만들어지고 교사들을 위한 연수 프로그램도 필요할 것이다.

과학의 발전은 현대 사회에 엄청난 영향을 주고 있고 교육 분야에서도 과학 학습에 대한 연구가 활발하게 이루어지고 있다. 반면 컴퓨터과학이 우리 사회에 가져온 다양한 방면의 엄청난 변화에도 불구하고 아직도 교육 현장에서는 컴퓨터과학 교육에 대한 기반이 자리 잡지 못한 실정이다. 본 논문에서 제안한 프로그래밍 중심의 STEAM 교육 프로그램을 통하여 학습자나 교수자들이 프

로그래밍을 학습하고 가르치는데 조금이라도 도움이 됐으면 하는 바람이다. 또한 프로그래밍과 컴퓨터과학 교육에 대한 관심이 많아져 좀 더 많은 교육 프로그램이 개발되어 21세기를 이끌어갈 미래 인재들에게 의미있는 교육이 이루어져야 할 것이다.



참 고 문 헌

- 교육과학기술부 (2011a), *과학 6-2 초등학교 교사용 지도서*. 서울: 금성출판사.
- 교육과학기술부 (2011b). *중학교 선택 교과 교육과정*, 서울: 교육과학기술부.
- 교육과학기술부 (2011c). *창의적 과학기술 인재대국 실현을 위한 과학기술인재 육성·지원 기본 계획('11-'15)*. Retrieved from <http://www.mest.go.kr>.
- 교육과학기술부 (2010). *2011년 업무보고 창의인재와 선진과학기술로 여는 미래 대한민국*. Retrieved from <http://www.mest.go.kr>.
- 교육인적자원부 (2005). *초중등학교 정보통신기술 교육 운영지침 개정안*, 교육인적자원부.
- 권대용 (2013). 텐지블 프로그래밍 도구를 활용한 논리적 사고력 기반의 초등 로봇 과제 개발 및 적용. *컴퓨터교육학회논문지*, 16(4), 13-21.
- 권대용 (2011). *Algorithmic brick based tangible robot and hybrid programming environments for enhancing computational thinking*. 박사학위 논문, 고려대학교 대학원.
- 권혁일. (2008). 디지털 스토리텔링이 초등학생의 수학 학업성취도 및 태도에 미치는 효과. *교육과학연구*, 39(3), 139-170.
- 김병수 (2013). *생산적 사고력 신장을 위한 PPS 기반 프로그래밍 교육 프로그램*. 박사학위 논문, 제주대학교 대학원 .
- 김병수, 김종훈 (2012). 초등 예비교사의 컴퓨터과학에 대한 인식 변화를 위한 생산적 사고 기반 알고리즘 학습의 설계 및 적용. *수산해양교육연구*, 24(4), 528-542.
- 김병수, 김종훈 (2013). CPS 모형 기반 스크래치 프로그래밍 학습이 언어 창의성에 미치는 영향. *컴퓨터교육학회논문지*, 16(6), 11-19.
- 김석희, 유현창 (2013). Hands on 센서 기반 고도화된 STEAM 교육 프로그램의 효과, *컴퓨터교육학회논문지*, 16(3), 79-89.
- 김정아, 김병수, 이지훤, 김종훈 (2011). 융합형 인재 양성을 위한 IT 기반 STEAM 교수학습 방안 연구. *수산해양교육연구*, 23(3), 445-460.

- 김종진, 현동립, 김승완, 김종훈, 원유현 (2010). 교육용 프로그래밍 언어인 로고와 스크래치 교재 개발 및 비교 실험. *한국콘텐츠학회, 10*(7), 459-469
- 김종혜 (2009). *정보과학적 사고 기반의 문제 해결 능력 향상을 위한 중등 교육 프로그램*. 박사 학위논문, 고려대학교 대학원.
- 김종훈, 양영훈, 김병수, 김승완, 김은길, 김태훈, 현동립 (2014). *스크래치로 배우는 STEAM 교육*. 경기: 다올미디어.
- 김진수 (2012). *STEAM 교육론*. 경기: 양서원.
- 김진수 (2011). STEAM 통합 교육의 수업 자료 제작을 위한 PDIE 모형 개발. *대한공업교육학회 학술대회 발표논문*, 386-392.
- 김태훈, 김병수, 김종훈 (2014). 센서 기반 스크래치 프로그래밍 중심의 STEAM 교육 프로그램 개발 및 적용. *정보교육학회논문지, 18*(1), 65-74.
- 김태훈, 김종훈(2014). 스크래치 프로그래밍 중심의 STEAM 교육 프로그램 개발 및 적용. *컴퓨터교육학회논문지, 17*(6), 49-57.
- 김태훈, 김종훈 (2013). Kodu를 이용한 프로그래밍 학습이 초등학생의 논리적 사고력과 학습 흥미에 미치는 영향. *컴퓨터교육학회논문지, 16*(3), 13-22.
- 김태훈, 김종훈 (2012). 초등학생의 프로그래밍 교육을 위한 STEAM 기반의 교과 융합 프로그램 개발. *한국컴퓨터교육학회 학술발표대회논문집, 16*(2), 73-68.
- 김태훈, 양영훈, 김종훈 (2013). Kodu를 이용한 프로그래밍 중심 STEAM 교육 프로그램 개발 및 적용. *수산해양교육연구, 25*(5), 1020-1030.
- 김형철 (2011). *컴퓨터과학 교육용 계산 원리 학습도구의 기능요소 고찰*. 석사학위 논문, 제주대학교 대학원.
- 김효남, 정완호, 정진우 (1998). 국가수준의 과학에 관련된 정의적 특성의 평가체제 개발. *한국과학교육학회지, 18*(3), 357-369.
- 나보라 (2011). *디지털 스토리텔링 수업 모형 개발 및 적용*. 석사학위 논문, 서울교육대학교 대학원.
- 노정원 (1997). *과학 교육 연구에 사용된 GALT 원본과 축소본에 대한 조사 연구*. 석사학위 논문, 이화여자대학교.
- 류충구 (2012). *Scratch가 초등 영재들의 창의적 문제해결력에 미치는 효과*. 석사학위논문, 경인교육대학교 교육대학원.

- 미래창조과학부 (2013). *SW 혁신전략*. Retrieved from http://www.msip.go.kr/www/brd/m_159/list.do
- 박용철, 이수정 (201). 스크래치 프로그래밍 교육이 초등학생의 자기 주도적 학습 능력에 미치는 효과. *정보교육학회논문지*, 15(1), 93-10.
- 박성익, 임철일, 이재경, 최정임 (2011). *교육방법의 교육공학적 이해*. 경기: 교육과학사.
- 박정신, 조석봉 (2012). 프로그래밍입문 수업에서 스크래치 활용 효과분석. *디지털정책연구*, 10(9), 49-456
- 박정호, 김철 (2012). 스토리텔링을 활용한 로봇 프로그래밍 수업의 효과. *정보교육학회논문지*, 16(2), 211-222.
- 배건 (1997). *초등학교 통합교육과정의 통합 유형 분석*. 석사학위 논문, 한국교원대학교 대학원.
- 배선아 (2009). *공업계열 전문계 고등학교 전기·전자·통신 분야의 활동 중심 STEM 교육프로그램 개발*. 박사학위 논문, 한국교원대학교 대학원.
- 백윤수, 박현주, 김영민, 노석구, 박종윤, 이주연, 정진수, 최유현, 한혜숙 (2011). 우리나라 STEAM 교육의 방향. *학습자중심교과교육연구*, 11(4), 149-171.
- 백윤수, 박현주, 김영민, 노석구, 이주연, 정진수, 최유현, 한혜숙, 최종현 (2012). *융합인재교육(STEAM) 실행방향 정립을 위한 기초연구*, 한국과학창의재단.
- 송정미 (2011). *스크래치를 활용한 정보과학적 사고 기반 퍼즐 교육*. 석사학위 논문, 고려대학교 교육대학원.
- 송정범 (2010). *STEM 통합교육을 위한 교실친화적 로봇교육 모형 및 프로그램 개발에 관한 연구*. 박사학위 논문, 한국교원대학교 대학원.
- 송정범, 조성환, 이태욱, (208). 스크래치 프로그래밍 학습이 학습자의 동기와 문제해결력에 미치는 영향. *정보교육학회논문지*, 12(3), 323-332
- 신승용 (2011). *로봇 프로그래밍 학습 참여자의 학습의도 영향요인 분석*. 박사학위 논문, 성균관대학교 대학원.
- 안경미, 손원성, 최윤철 (201). 스크래치 프로그래밍 교육이 초등학생의 학습 몰입과 프로그래밍 능력에 미치는 효과. *정보교육학회논문지*, 15(1), 1-10
- 염혜진 (2006). *스토리텔링을 활용한 말하기 능력 신장연구*. 석사학위 논문, 한국

- 외국어대학교 대학원.
- 오정철, 이지훤, 김정아, 김종훈 (2012). 스크래치를 활용한 STEAM 기반 교육 프로그램 개발 및 적용. *컴퓨터교육학회논문지*, 15(3), 11-23.
- 윤일규, 김종혜, 이원규 (2010). 정보 교과와 문제해결과정에서 논리적 사고력 구성요소에 대한 조작적 정의. *컴퓨터교육학회논문지*, 13(2), 1-14.
- 이상갑 (2001). *주제 중심 통합적 접근에 의한 기술교과 교육프로그램 개발*. 박사학위 논문, 한국교원대학교 대학원.
- 이소이 (2011). *STEM 통합 접근의 기술 수업 설계 모형 개발*. 박사학위 논문, 충남대학교 대학원.
- 이영준, 백성혜, 신재홍, 유현창, 정인기, 안상진, 최정원, 정성균 (2014). 초중등 단계 Computational Thinking 도입을 위한 기초 연구. 한국과학창의재단.
- 이옥화, 허휘옥, 조미현, 김민경, 김미량, 강신천, 강오한, 이준희, 최재혁, 추병완 (2008). *정보교육의 ABC*. 경기: 교육과학사.
- 이은경 (2013). 계산적 사고 향상을 위한 창의적 스크래치 프로그래밍 학습. *컴퓨터교육학회논문지*, 16(1), 1-9.
- 이은경 (2009). *Computational Thinking 능력 향상을 위한 로봇 프로그래밍 교수 학습 모형*. 박사학위 논문, 한국교원대학교 대학원.
- 이재무 (2014). Rossett 모형을 적용한 적응형 이러닝 시스템을 위한 요구 분석. *한국콘텐츠학회논문지*, 14(6), 529-538.
- 이진희 (2009). 스토리텔링을 활용한 통합수업이 아동의 창의적 표현력에 미치는 영향: 초등학교 2학년 전래동화 그리기 활동을 중심으로. 석사학위 논문, 이화여자대학교 대학원.
- 이찬도 (2010). 처음 배우는 프로그래밍 언어로 무엇을 어떻게 가르칠까?. *정보과학회지*, 28(4), 24-27.
- 이창훈 (2007). *창의 공학 설계 교육 프로그램이 공학 입문자의 창의력과 공학 설계 능력에 미치는 효과*. 박사학위 논문, 충남대학교 대학원.
- 이하룡 (2013). *디지털 스토리텔링 기반의 STEAM 수업 프로그램 개발 및 적용 효과*. 박사학위 논문, 부산대학교 대학원.
- 임화경, 조용남 (2012). Kodu 비주얼 프로그래밍 언어를 사용한 초등학생의 창의

- 적 3D 게임프로그래밍 학습. *컴퓨터정보학회논문지*, 17(11), 53-61.
- 정재삼 (2004). *교육프로그램평가*. 서울: 교육과학사.
- 조성환, 송정범, 김성식, 이경화 (2008). CPS에 기반한 스크래치 EPL이 문제해결력과 프로그래밍 태도에 미치는 효과. *정보교육학회논문지*, 12(1), 7-8.
- 조준필 (2012). *기술교육에서 중학생의 논리적 사고력 함양을 위한 스크래치 학습 프로그램 개발*. 석사 학위논문, 한국교원대학교 대학원.
- 조향숙, 김훈, 허준영 (2012). *현장 적용 사례를 통한 융합인재교육(STEAM)의 이해*. 한국교육개발원.
- 한선관, 김수환, 서정보 (2010). 스크래치 프로그래밍을 활용한 게임중독 치료 프로그램의 개발. *정보교육학회논문지*, 14(1). 61-68.
- 한아름 (2009). *스토리텔링을 이용한 유아영어교육 방법론*. 석사학위 논문, 이화여자대학교 대학원.
- 한정혜, 박주현, 조미현, 박일우, 김진오 (2011). 초등정규교육과정에서 STEAM을 위한 로봇활용교육. *정보교육학회논문지*, 15(3), 483-492.
- 함성진, 양창모 (201). 스크래치를 이용한 초등학교 컴퓨터 교육과정 설계. *정보교육학회논문지*, 15(3), 413-423.
- 허경 (2011). PBL 기반 초등 로봇 프로그래밍 교육과정 개발. *한국정보교육학회 논문지*, 15(4). 543-550.
- EBS 다큐프라임 ‘이야기의 힘’ 제작팀 (2011). *이야기의 힘*. 서울: 황금물고기.
- Torrance, E. P. 김영채 역 (2010). *Torrance 창의력 검사 TTCT (도형) A형 검사요강*, 대구: 창의력 한국 FPSP.
- Rossett, A. (1987). *Training needs assessment*, Educational Technology Publications.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?. *Acm Inroads*, 2(1), 48-54.
- Benedsen, J., & Caspersen, M. E. (2008, September). Abstraction ability as an indicator of success for learning computing science?. *Proceedings of the Fourth international Workshop on Computing Education Research* (p.

- 15-26). ACM. doi: 10.145/1404520.1404523
- Bybee, R. W. (2010). Advancing STEM Education: A 2020 Vision. *Technology and Engineer Teacher*, 70(1), 30-35.
- Cresswell, J., & Vayssettes, S. (2006). *Assessing scientific, reading and mathematical literacy: a framework for PISA 2006*. Organisation for Economic Cooperation and Development (OECD).
- Denning, P. J. (2010). *Ubiquity symposium 'What is computation?': Opening statement*. Ubiquity, 2010(November), 1.
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13-18.
- Devlin, K. (2003, September). Why universities require computer science students to take math. *Communications of the ACM*, 46(9), 37-39.
- Egan, K. (1989). *Teaching as story telling: An alternative approach to teaching and curriculum in the elementary school*. University of Chicago Press.
- Fowler, A. and Cusack, B. (2011). Enhancing introductory programming with Kodu Game Lab: an exploratory study. *Proceedings of the 2nd Annual Conference of Computing and Information Technology Education and Research in New Zealand*, (pp. 69-79).
- Fowler, A., Fristoe, T., & MacLauren, M. (2012). Kodu Game Lab: a programming environment. *The Computer Games Journal: Whitsun 2012*, 17-22.
- Gallets, M. P. (2005). *Story telling and story comprehension*. A Thesis presented at the department of curriculum and instruction, East Tennessee State University.
- Hazan, O. (1999). Reducing abstraction level when learning abstract algebra concepts. *Educational Studies in Mathematics*, 40(1), 71-90. doi:10.1023/A:103780613628
- Kane, L., Berger, W., Anton, G., Shapiro, B., & Squire, K. (2011). Studio K:

- a game design curriculum for computational thinking. *GLS 8.0 Conference Proceedings*, 469-472.
- Koppelman, H., & van Dijk, B. (2010, June). Teaching abstraction in introductory courses. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (p. 174-178). ACM. doi: 10.145/182090.182140
- Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, 50(4), 36-42. doi: 10.145/1232743.1232745
- Logo, C. S. T. A. (2011). *CSTA K-12 Computer Science Standards* (Doctoral dissertation, Bowdoin College).
- MacLaurin, M. (2011). The design of kodu : a tiny visual programming language for children on the xbox 360. *POPL '11 Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 241-246.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 16:1-16:15.
- McWilliams, T. P., & McWilliams, V. B. (2011). Another Look At Theoretical And Em-pirical Issues In Event Study Methodology. *Journal of Applied Business Research (JABR)*, 16(3).
- National Research Council. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. National Academies Press.
- National Research Council. (2011). *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. National Academies Press.
- Cresswell, J., & Vayssettes, S. (2006). *Assessing scientific, reading and mathematical literacy: a framework for PISA 2006*. Organisation for Economic Cooperation and Development (OECD).
- Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai,

- Y. (2009). *Scratch: Programming for All*, *Communications of the ACM*, 52(11), 60-67.
- Roadrangka, V., Yeany, R. H., & Padilla, M. J. (1983, April). The construction of a group assessment of logical thinking (GALT). *In the annual meeting of the National Association for Research in Science Teaching*. Dallas, Texas, April (pp. 5-8).
- Romeike, R. (2017b, November). Applying creativity in CS high school education: criteria, teaching example and evaluation. *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research—Volume 8* (p. 87-96). Australian Computer Society, Inc.
- Rossett, A. (1987). *Training needs assessment*, Educational Technology Publications.
- Satchwell, R. E., & Loepp, F. L. (2002). Designing and implementing an integrated mathematics, science, and technology curriculum for the middle school. *Journal of Industrial Teacher Education*, 39(3).
- Sanders, M. (2009). Stem, stem education, stemmania. *The Technology Teacher*, 68(4), 20-26.
- Schank, R. C. (1990). *Tell me a story: A new look at real and artificial memory*. Charles Scribner's Sons.
- Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. *In Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 99-104). ACM.
- Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. *In Proceeding of the 44th ACM technical symposium on Computer science education*, 609-614.
- Wells, G. (1986). *The meaning makers: Children learning language and using language to learn*. Heinemann Educational Books Inc., 70 Court St., Portsmouth, NH 03801.

- Wing, J. M. (2006). Computational thinking, *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725.
- Yakman, G. (2008). STEAM education: an overview of creating a model of integrative education. *Pupil's Attitudes Toward Technology*, 19, 335-358.
- Yakman, G. (2010). *What is the point of STEAM? - A Brief Overview*. Retrieved from http://www.steamedu.com/2006-2010_Short_WHAT_IS_STEAM.pdf.



<ABSTRACT>

STEAM Education Program based on Programming to Improve Computational Thinking Ability

TaeHun Kim

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

Supervised by professor JongHoon Kim

The purpose of this research is to develop STEAM education program based on programming on to improve computational thinking ability, and to apply this to the field of school and to verify the educational effectiveness.

The development of advanced science and technology changes modern society rapidly. In the 21st century, the issues, which should be based on integrated knowledge and thinking such as environmental pollution, food shortage, energy depletion, are increasing, and there are limitations to resolving these issues with the existing knowledge-based approach, and in order to resolve these, integrated and creative thinking skills and power of execution which transcend knowledge are needed. In order for the learners who will live in the integration era to grow into major talents to lead the integrated society, they should learn computing principles and technology that penetrate the integrated knowledge and the core of integration and cultivate the ability that is necessary to solve the problems creatively and efficiently. Under these circumstances, this research developed the programing-oriented STEAM educational program to enhance students' computational thinking and integrated thinking.

In this research, in order to develop the STEAM education program based on programming, research was conducted according to the stages of ADDIE model.

In the analysis stage, literature research analysis on computational thinking skills and STEAM education and programming education and the analysis of preceding researches on STEAM education utilizing programming, and the learners' demand analysis were conducted.

In the designing stage, in order to develop the educational program that reflects the demand analysis of the analysis stage, the elementary school 2009 revised curriculum was analyzed and based on this, the learning standards were selected and the role of each subject element of STEAM was determined. In order to determine the form of integration, integration types were selected, and for the STEAM education program, the loop learning model was selected as the learning model.

Based on the design prepared in the design stage, 4 STEAM education programs based on programming were developed. The developed educational programs were divided, according to teaching strategies and input method, into the 코뮤티 프로그래밍-orientated STEAM education program, the sensor-based scratch 프로그래밍-orientated STEAM education program for donation for education, the scratch 프로그래밍-orientated STEAM education program for afterschool class, and the scratch 프로그래밍-orientated STEAM education program for the regular school curriculum.

The STEAM education programs based on programming were applied to the experimental group through education donation, after-school class, regular class for a total of 4 times, and the control group, which was treated with general programming learning or traditional separate subject class, was selected and the program was applied.

In order to examine if the STEAM education program based on programming developed in this research is effective in enhancing the affective domain of the learners' creativity, logical thinking skills, and science, the null hypothesis was

established and tested.

As a result of evaluation, it was confirmed that the experimental group which learned the STEAM education program based on programming developed in this research was significantly enhanced in creativity, logical thinking skills, and science, compared with the control group which had general programming class or traditional separate subject class. Such results of this research can be interpreted that the STEAM education program based on programming had positive effects on the learners' computational thinking skills and integrated knowledge. If, through follow-up researches, the STEAM education program based on programming developed in this research is continuously improved and its effects on diverse learners are analyzed more systematically, the programming education to enhance computational thinking skills and the learning effectiveness of STEAM education for integrated knowledge will be maximized.



부 록

<부록 1> 사전·사후 집단 간 Mann-Whitney 검정	156
<부록 2> 사전·사후 집단 내 Wilcoxon 부호순위 검정	157
<부록 3> 코딩 프로그래밍 교재	158
<부록 4> 스크래치 프로그래밍 교재	175
<부록 5> 코딩 프로그래밍 중심 STEAM 교육 프로그램 교재	191
<부록 6> 스크래치 프로그래밍 중심 STEAM 교육 프로그램 교재	211
<부록 7> 정규교육과정에서의 STEAM 교육 프로그램 교재	225



<부록 1> 사전·사후 집단 간 Mann-Whitney 검정

사전·사후 집단 간 Mann-Whitney 검정

하위요소	기간	집단	N	M	SD	평균순위	Mann-Whitney의 U	p
보존	사전	실험	23	2.609	1.118	26.15	203.500	.162
		비교	23	2.261	.864	20.85		
	사후	실험	23	2.913	.733	28.41	151.500	.007**
		비교	23	2.304	.765	18.59		
비례	사전	실험	23	1.478	1.675	23.20	257.500	.873
		비교	23	1.348	1.335	23.80		
	사후	실험	23	1.435	1.727	23.54	263.500	.982
		비교	23	1.348	1.555	23.46		
변인통제	사전	실험	23	1.261	1.251	23.93	254.500	.819
		비교	23	1.217	1.313	23.07		
	사후	실험	23	1.261	1.322	23.39	262.000	.954
		비교	23	1.217	1.166	23.61		
확률	사전	실험	23	0.522	.790	23.91	255.000	.801
		비교	23	0.478	.790	23.09		
	사후	실험	23	0.478	.790	22.78	248.000	.661
		비교	23	0.609	.891	24.22		
상관	사전	실험	23	0.130	.344	23.00	253.000	.685
		비교	23	0.174	.388	24.00		
	사후	실험	23	0.348	.573	25.09	228.000	.279
		비교	23	0.174	.388	21.91		
조합	사전	실험	23	1.000	.905	26.54	194.500	.097
		비교	23	0.652	.982	20.46		
	사후	실험	23	1.783	.850	28.07	159.500	.015*
		비교	23	1.087	.949	18.93		
합계	사전	실험	23	7.000	4.068	24.91	232.000	.473
		비교	23	6.130	3.559	22.09		
	사후	실험	23	8.217	4.210	25.72	213.500	.260
		비교	23	6.739	3.828	21.28		

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

<부록 2> 집단 내 사전·사후 Wilcoxon 부호순위 검정

사전·사후 집단 내 Wilcoxon 부호순위 검정

집단	하위요소	시기	N	M	SD	Z	p
실험	보존	사전	23	2.609	1.118	-1.615	.106
		사후	23	2.913	.733		
	비례	사전	23	1.478	1.675	-.284	.776
		사후	23	1.435	1.727		
	변인통제	사전	23	1.174	1.267	-.615	.539
		사후	23	1.261	1.322		
	확률	사전	23	0.609	.839	-.722	.470
		사후	23	0.478	.790		
	상관	사전	23	0.174	.388	-1.265	.206
		사후	23	0.348	.573		
	조합	사전	23	1.000	.905	-3.626	.000**
		사후	23	1.783	.850		
	합계	사전	23	7.043	4.095	-2.185	.029*
		사후	23	8.217	4.210		
비교	보존	사전	23	2.261	.864	-.500	.617
		사후	23	2.304	.765		
	비례	사전	23	1.348	1.335	-.328	.743
		사후	23	1.348	1.555		
	변인통제	사전	23	1.217	1.313	-.037	.971
		사후	23	1.217	1.166		
	확률	사전	23	0.478	.790	-1.000	.317
		사후	23	0.609	.891		
	상관	사전	23	.17	.388	.000	1.000
		사후	23	.17	.388		
	조합	사전	23	0.652	.982	-2.500	.012*
		사후	23	1.087	.949		
	합계	사전	23	6.130	3.559	-1.422	.155
		사후	23	6.739	3.828		

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

<부록 3> 코두 프로그래밍 교재



KODU반 ()학년 이름 ()

Kodu 프로그래밍 수업 계획	
	프로그래밍이란?
1교시	코두 설치하기 객체 추가하기 명령 내려보기 센서와 동작기 알아보기
2교시	메뉴 살펴보기 지형 편집하기 경로 만들기
3교시	타이머 스코어
4교시	creatable 카메라
5교시	on water on land 페이지
6교시	동작기와 센서를 이용한 이야기 만들기

1교시	KODU와 만나기
------------	------------------

1. 프로그래밍이란?

로봇(컴퓨터)에게 여러분이 원하는 일을 시키기 위해서 로봇이 이해할 수 있는 언어로 그 일을 설명(표현)하는 것을 **프로그래밍**이라고 하고 이렇게 프로그래밍을 통해 만들어진 명령의 모음을 **프로그램**이라고 합니다.

아래와 같은 상황에서 로봇에게 사용 가능한 명령을 이용하여 청소라는 프로그램을 프로그래밍하려면 어떻게 표현하면 될까요?



로봇에게 청소 설명하기

①		④	
②		⑤	
③		⑥	

① ~ ⑥ 번의 과정을 반복하면 로봇은 청소라는 프로그램을 수행하게 됩니다. 이러한 표현한 청소라는 프로그램은 로봇에게 사용 가능한 명령 5번으로 추가시킬 수도 있겠네요. 이렇게 로봇(컴퓨터)은 자기만의 언어와 명령을 갖고 있고 이것을 이용하여 새로운 프로그램을 만들고 로봇의 명령으로 추가할 수 있습니다.

2. 코두 설치하기

코두 프로그래밍을 하기 위해 코두를 설치하여 봅시다. 코두를 설치하기 위해선 Kodu 프로그램을 다운 받아야 합니다. 코두를 다운 받는 사이트는 (<http://fuse.microsoft.com/page/kodu>)입니다. 교육기부 홈페이지 코두반 게시판(<http://cerg.ijeju.ac.kr>) 1번 게시글에 코두 설치 방법과 함께 설치파일을 올려놓았습니다.

- 간단하게 설치 방법을 설명하면 다음과 같습니다.
- ① 설치할 파일을 다운 받습니다.
 - ② 순서대로 파일을 설치합니다.

③ 자기 컴퓨터에 맞게 환경설정을 합니다.

컴퓨터의 운영체제나 사양에 따라 설치가 되지 않는 경우가 있습니다. 이럴 경우 주위 분들의 도움을 요청하세요.

3. 객체 추가하기

코듀에는 다양한 로봇과 건물, 물건들이 있습니다. 이렇게 코듀에 집어넣어 프로그래밍 할 수 있는 다양한 것들을 객체라고 합니다. 코듀를 추가해 봅시다.



화면 아래에 있는 메뉴에서 코듀 모양의 객체 도구(Object Tool)를 선택하고 코듀를 추가하고 싶은 곳에 클릭합니다. 추가하고 싶은 객체를 선택하면 그 위치에 객체가 추가됩니다.

4. 간단한 명령 내리기



코듀에게 사과를 먹으라는 명령을 내리기 위해 사과를 몇 개 추가하여 봅시다. 코듀에게 명령을 내리기 위해서 프로그래밍 모드(코듀 위에서 마우스 오른쪽 버튼)로 들어갑니다.



프로그래밍 모드로 들어가면 + 버튼을 눌러 아이콘을 선택하면 됩니다.



① 사과를 보면 그 방향으로 이동한다.
② 사과와 부딪히면 먹는다.

위에서처럼 코드의 프로그래밍은 두 부분으로 나뉩니다.



WHEN see apple, DO move toward
=> 사과를 보면, 그 방향으로 이동한다.

(1) When

When 부분에는 주변의 상황의 변화를 감지하는 센서가 옵니다. 센서에는 see(보기), bump(부딪히기), hear(듣기) 등이 있습니다. When 부분에는 Do 부분에는 설명되는 동작기를 작동시키기 위한 조건을 설명합니다. 예를 들어 어떤 일이 발생했을 때 사과를 먹는지, 공을 차는지, 미사일을 발사하는지를 설명해주는 것입니다.

센서	아이콘	의미	센서	아이콘	의미
see		~가 보이면	gamepad		게임 패드 ~를 누르면
hear		~가 들리면	health		체력이 ~이면
bump		~와 부딪히면	on water		~번 물 위에 있으면
timer		시간이 ~초 지나면	on land		~번 땅 위에 있으면
got		~을 갖으면	always		언제나
scored		점수가 ~점이면	held by		~에게 잡혔으면
keyboard		키보드 ~을 누르면	shot hit		~을 맞추면
mouse		마우스 ~을 누르면			

(2) Do

Do 부분에는 로봇이 실행할 동작기를 설명합니다. 사과를 보면 어떤 동작을 할지, 공과 부딪히면 어떤 동작을 할지를 설명하는 부분입니다. 동작기에는 move(움직이기), shoot(발사하기), jump(점프하기), eat(먹기) 등이 있습니다.

동작기		아이콘	의미	동작기	아이콘	의미
	move		움직인다	view	1st person	카메라가 주인공이 보는 화면을 비춘다
	shoot		~을 쏜다		follow	카메라가 주인공을 따라 다닌다
	turn		회전한다		ignore	카메라가 주인공을 무시한 화면을 비춘다
	eat		먹는다	holding	drop	(잡고 있던) 물건을 떨어뜨린다
	switch		페이지를 바꾼다		give	(가지고 있는) 물건을 준다
	inline		다른 페이지를 갖고 온다		grab	물건을 잡는다
actions	open		입구를 연다	game	end	게임이 끝난다 (game over)
	close		입구를 닫는다		win	게임에서 승리한다 (winner)
	color		색깔을 바꾼다		score	점수를 얻는다
	create		물건을 만든다		subtract	점수를 잃는다
	express		감정을 표현한다		reset	점수, 세계를 다시 설정한다
	glow		빛을 낸다		boom	폭발한다
	launch		물건을 내보낸다 (뿜겨낸다)		damage	손상을 입는다
	play		음악을 연주한다		heal	치료한다
	quiet		음악을 정지한다		knockout	의식을 잃는다
	say		말한다		stun	기절한다
				vanish	사라진다	

코딩에서 아래의 프로그램을 직접 작성해 봅시다.

명령	When	Do
키보드의 화살표를 누르면 움직인다.	When	Do
공과 부딪히면 그것을 찬다(튕겨낸다).	When	Do
사과를 먹으면 빨간 빛을 낸다.	When	Do
마우스 왼쪽 클릭을 하면 미사일을 쏜다	When	Do



2교시	WORLD(세계) 만들기
------------	----------------------

1. 메뉴 살펴보기

코듀에서 객체와 세계를 추가/편집/삭제할 수 있는 화면은 편집모드라고 합니다. 아래의 두 그림은 코듀 편집 모드와 편집 메뉴를 보여줍니다.



번호	메뉴 이름	하는 일
①	Home Menu(홈 메뉴)	세계의 저장 / 불러오기, 새로운 세계 불러오기, 메인메뉴로 이동
②	Play Game(게임 실행)	게임 시작하기 (편집모드는 esc키)
③	Move Camera(화면 위치 조정)	화면을 확대 / 축소 / 이동 / 회전
④	Object Tool(객체 도구)	객체 추가 / 삭제 / 프로그래밍 / 상태변경
⑤	Path Tool(경로 도구)	객체를 위한 경로(길)만들기
⑥	Ground Brush(지형 붓)	지형을 추가 / 삭제 / 색깔변경
⑦	Up/Down(언덕 도구)	지형에 언덕을 추가하거나 삭제하기
⑧	Flatten(평평하게)	고르지 못한 땅을 평평하게 고르는 기능
⑨	Roughen(뽀족하게)	뽀족한 지형을 만드는 기능
⑩	Water Tool(물 도구)	물을 추가 / 삭제 / 변경
⑪	Delete Tool(삭제 도구)	객체를 지울 수 있는 지우개 기능
⑫	Change World Setting	WORLD(세계)의 설정을 변경하는 부분

2. 나만의 세계 만들기

앞서 살펴 본 메뉴를 이용하여 자신만의 세계를 만들어 봅시다. 메뉴를 이용하여 다양한 지형을 만들 수 있습니다. 메뉴 중에서 지형을 만드는 5개가 있습니다.

메뉴 이름	아이콘	하는 일

Ground Brush 땅 그리기		땅을 색칠하거나 추가하거나 지울 수 있습니다.
Up/Down 언덕 만들기		언덕이나 계곡을 만들 수 있습니다.
Flatten 평평하게 하기		땅을 평평하게 만들거나 층을 만들 수 있습니다.
Roughen 뾰족하게 하기		땅을 뾰족하거나 언덕이 많은 형태로 만들 수 있습니다.
Water Tool 물 도구		물을 추가하거나 없애고 물의 색깔을 바꿀 수 있습니다.

3. 홈 메뉴 사용하기



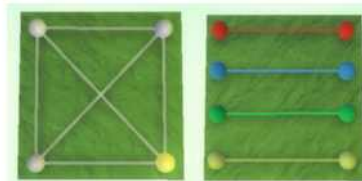
홈 메뉴에서는 세계를 불러오고 저장할 수 있는 메뉴가 있습니다. 새로운 세계를 만들어 저장하고 싶을 때는 홈 메뉴의 save my world를 선택하면 됩니다.

Play world(플레이(게임)하기), Edit world(세계 편집하기), Save my world(내가 만든 세계 저장하기), Load world(세계 불러오기), New world(기본 세계 불러오기) 등의 메뉴가 있습니다.

4. 경로 만들기



경로는 각 객체들의 이동을 조절하는 용도로 사용됩니다. 원하는 방향으로 이동시키기 위해서 경로를 이용하면 됩니다. 객체도구를 이용하여 아래 모양의 경로를 만들어 봅시다.



※ 문제해결하기

1. 지형 만들기 도구를 이용하여 나만의 세계 만들기

- 제주도 모양의 세계 만들기
- 미래의 로봇 도시 만들기

2. 경로를 이용한 레이싱 게임 만들기

- 1번에서 만든 세계를 사용한다
- 두 개 이상의 경로(색깔을 다르게)를 추가한다.
- 서로 다른 경로를 움직이는 두 개의 로봇 프로그래밍 하기



3교시	Score & Timer 이용하기
------------	-------------------------------

1. 슈팅 게임 만들기

이번 사용할 세계는 Shooting Fish입니다.

LOAD WORLD - Shooting Fish



아래 주어진 조건에 맞게 움직이고 미사일을 발사하는 코두를 프로그래밍 해 보세요.

<프로그래밍 하기 1> : 코두(KODU)

- ① 스페이스 바를 누르면 미사일을 발사한다.
- ② 화살표를 누르면 (그 방향으로) 빠르게 움직인다.

2. score(점수) 사용하기

score를 사용하면 각 객체의 점수를 기록할 수 있습니다. 그 외에도 score는 점수 뿐만 아니라 물체의 개수를 저장할 때도 사용됩니다. score 기능을 사용하여 점수를 기록하는 게임을 프로그래밍 하세요.

<프로그래밍 하기 2> : 코두(KODU)

- ① 스페이스 바를 누르면 미사일을 발사한다.
- ② 화살표를 누르면 (그 방향으로) 빠르게 움직인다.
- ③ 물고기를 쏘면 빨간색 점수를 1점 더한다.

점수는 색깔, 알파벳으로 구분할 수 있습니다. 점수를 구분한 이유는 무엇일까요? 점수를 구분해야 다양한 객체의 점수나 개수를 저장할 수 있기 때문입니다.

3. Timer(시간 센서) 이용하기

코두에는 시간을 쫓 수 있는 센서가 있어서 원하는 만큼 시간을 재서 동작기를 움직이도록 할 수 있습니

다. Timer를 이용하여 적군인 물고기가 코듀를 공격하도록 프로그래밍 해 봅시다.

(1) 적군이 공격하게 프로그래밍 하기

<프로그래밍 하기 3>처럼 프로그래밍을 했을 때의 문제점은 무엇일까요?

<프로그래밍 하기 3> : Fly Fish(나는 물고기)

- ① (언제나) 경로 위를 움직인다.
- ② Fly Fish를 보면 피해서 움직인다.
- ③ 코듀를 보면 총알(blip)을 쏜다

아래의 두 명령을 비교해 봅시다. 어떤 차이가 있나요?

③ 코듀를 보면 총알을 쏩니다.	③ <u>하얀색</u> 코듀를 보면 총알을 <u>한 번</u> 쏩니다.

(2) Timer 센서 사용하기

(1)번에서 코듀를 공격하는 물고기는 총알을 몇 발이나 쏘나요? 왜 그럴까요?

이번에는 물고기가 3초가 지날 때 마다 코듀를 공격하게 프로그래밍을 해 봅시다.

<프로그래밍 하기 4> : Fly Fish(나는 물고기)

- ① (언제나) 경로 위를 움직인다.
- ② Fly Fish를 보면 피해서 움직인다.
- ③ 타이머가 3초가 지나고, 코듀를 보면 총알(blip)을 한 번 쏜다

(3) 여러 개의 센서 + 동작기, 하나의 센서 + 여러개의 동작기

④번 문장이 ③번 문장에 비해 뒤로 움직인 것을 볼 수 있습니다. 숫자 ④위에 커서를 옮겨놓고 뒤로 드래그하면 문장이 뒤로 옮겨집니다. 동시에 두 가지 센서가 반응해야 동작기를 실행하라는 의미입니다. 즉, 시간이 3초가 지나고(센서1 - 시간 센서), 하얀색 코듀가 보일 때(센서 2 - 보기 센서) 한 방의 총알을 발사합니다. 그보다 많은 조건을 줄 수도 있습니다.

반대로 하나의 센서가 반응할 때 2개 이상의 동작기를 실행하도록 만들 수도 있습니다. 아래 명령은 3초가 지나면 총알을 쏘고 빨간색 점수를 1점 올리라는 명령입니다.



4교시	Creatable & 카메라
------------	----------------------------

1. create 동작기

create 동작기는 로봇이 객체(object)를 만들 때 사용됩니다. 예를 들어 아래의 그림은 코듀에게 3초에 한 번씩 사과를 만들어 내라는 명령입니다.



로봇을 일반적으로 사과, 공, 동전 같은 객체만 만들어 낼 수 있습니다. 하지만 로봇에 creatable 속성을 이용하면 로봇들도 만들어 낼 수 있습니다.

2. Creatable 속성 - 프로그래머가 만들 수(create) 있도록 해주는 속성

Creatable은 만들어 낼 수 있다는 의미입니다. 로봇에게 Creatable 속성을 주면 다른 로봇들이 명령을 통해 그 로봇을 만들어 낼 수 있습니다.

Creatable 속성을 사용하면 몇 가지 장점이 있습니다.

- ① 프로그램에서 create 동작기를 이용해 객체 뿐 만 아니라 로봇도 만들어 낼 수 있습니다.
- ② 프로그래밍 된 객체와 로봇을 만들어 낼 수 있습니다. 일반적인 방법으로 만들어 낼 사과는 만들어 낼 이후에 아무런 변화가 없지만 Creatable로 만든 사과는 색깔이 변하고 먹을 때마다 점수도 달라지게 할 수 있습니다.
- ③ Creatable 로봇을 복사할 경우 하나의 프로그램을 수정하면 나머지 복사한 로봇들의 프로그램도 모두 수정이 됩니다.

Creatable 속성을 주기 위해서 객체 위에서 마우스 오른쪽 버튼을 클릭합니다. 속성 변경하기(Change Setting)로 들어가면 로봇의 상태(속도, 체력 등)를 변경할 수 있습니다. Creatable라는 속성에 초록색 불이 들어오게 체크를 하면 로봇이 Creatable 속성을 가지게 됩니다.








세계에 creatable의 속성을 가진 로봇을 만들면 create 동작기에서 creatables라는 선택사항이 생기게 되고 이것을 선택하면 그 로봇을 만들 수 있습니다.

※ 문제해결하기

1. 자유롭게 움직이는 구름(crowd)을 만들어 내는 공장(factory)을 프로그래밍 해 봅시다.
2. 3초 뒤에 사라지는 빛(light)을 프로그래밍 해 봅시다.

3. 카메라 동작기 사용하기

카메라 동작기를 잘 사용하면 좀 더 멋진 효과를 낼 수 있습니다. 카메라를 사용하는 방법은 크게 세가지가 있습니다. view(보기) 동작기에는 다음과 같이 세가지 카메라 동작기가 있습니다.

- ① follow(따라가기)  : 카메라가 주인공(카메라 동작기를 사용한 로봇)을 따라 가며 보여줍니다.
- ② 1st person(주인공)  : 카메라가 주인공이 보는 세계를 보여줍니다.
- ③ ignore(무시하기)  : 주인공과는 상관없이 카메라를 보여줍니다. 하지만, 프로그램 실행 중에도 마우스를 이용해서 각도나 위치를 변경할 수 있다는 장점이 있습니다.

※ 문제해결하기

1. 3가지 카메라 동작기를 모두 사용하는 프로그램을 만들어 봅시다.
 - 프로그램이 시작되면 카메라는 ignore(무시하기) 화면입니다.
 - 직접 조종하여 사과를 먹습니다.
 - 사과를 먹으면 카메라는 1st person(주인공) 화면으로 변합니다.
 - 동전을 먹으면 카메라는 follow(따라가기) 화면으로 변합니다.
2. 각각의 카메라 동작기가 어떠한 상황에 어울릴지 생각해 봅시다.

5교시	on land, on land & page
------------	------------------------------------

1. on land(땅 감지) 센서

코뮤티에는 다양한 색깔의 땅이 있고 각 땅에는 번호(1~121번)가 있습니다. 코뮤티의 각 로봇에는 어느 땅에 위치했는지 알아낼 수 있는 센서가 있습니다. 예를 들어 10번 땅에 있을 때는 느리게 움직이고, 27번 땅에 있을 때는 점수를 1점 얻도록 만들 수도 있습니다.



- (1) 축구공에 on land 센서를 이용하여 골을 감지하는 프로그램을 만들어 봅시다.
- (2) 순간이동을 하는 프로그램을 만들어 봅시다.

2. on water(물 감지) 센서

물을 이용하여 세계를 만들어 낼 수 있습니다. 물에는 1~10번 까지 10개의 종류가 있습니다. 코뮤티에는 어떤 수면에 위치하였는지 알아낼 수 있는 센서가 있습니다. 예를 들어 1번 물 위에 있을 때 물을 발견했다는 말을 하도록(say) 할 수 있습니다.



※ 문제해결하기

- 1. 축구공에 on land 센서를 이용하여 골을 감지하는 프로그램을 만들어 봅시다.
- 2. 순간이동을 하는 로봇을 만들어 봅시다.
- 3. 마음대로 움직이다가 물 위로 이동하면 기쁨을 표현하는 로봇을 프로그래밍 해 봅시다.

3. page(페이지) 이용하기

모든 책에 페이지가 있듯이 코뮤티에서도 페이지로 프로그래밍을 구분할 수 있습니다. switch와 inline를 사용하면 다른 페이지로 이동하거나 다른 페이지의 내용을 갖고 와서 실행할 수 있습니다. 페이지를 이용하면 매우 편리하게 로봇의 동작을 바꿀 수 있습니다.



위의 그림에서 페이지 1에 있을 때는 스페이스 바를 누를 때 점프를 하지만 사과를 먹고 페이지 2로 이동했을 때는 미사일을 발사합니다. 이렇게 페이지는 기존의 명령 중에 바꾸어야 할 내용이 있을 때 이용하면 편리합니다.

(1) switch(페이지 바꾸기)

switch는 페이지를 다른 페이지로 이동시키겠다는 동작기이다. 로봇은 새로 불러온 페이지의 명령들만 실행합니다.

(2) inline(페이지 가져오기)

inline는 지금 페이지에서 다른 페이지의 내용을 불러와서 실행시킵니다. 예를 들어 페이지 1에서 inline page3을 하면 페이지 1 내용을 실행하다 특정 센서가 반응할 때 페이지 3내용을 가져와서 실행합니다.

※ 문제해결하기

1. shooting fish 맵을 이용하여 프로그래밍을 만들어 봅시다.
 - 프로그램이 시작되면 카메라 모드는 코두의 주인공 (1st person) 화면이 됩니다.
 - 돌맹이에서 fly fish를 만들어 냅니다.(Creatable 이용)
 - fly fish는 만들어 진지10초가 지나면 코두를 공격합니다.(Page, switch 이용)
 - 또 10초가 지나면 나무로 이동하여 나무가 fly fish를 먹습니다.
 - Kodu가 5점을 얻으면 승리, 나무가 5점을 얻으면 게임에서 집니다.

코뮤티에는 말을 할 수 있는 동작기(say)와 들을 수 있는 센서(hear)가 있어서 다양한 상황을 만들어 낼 수 있습니다.

1. say(말하기) 동작기

Actions 동작기 모음에 Say를 선택하면 로봇에게 말을 하도록 프로그래밍 할 수 있습니다. 단 영어만 사용할 수 있고 는 말을 할 수 있는 동작기가 있습니다. Say 동작기를 이용하여 다른 로봇에게 명령을 내릴 수 있고 대화를 하는 프로그래밍을 할 수 있습니다.

2. hear(듣기) 센서

Hear 센서를 이용하면 다른 로봇이 한 말을 감지할 수 있습니다. 또한 다른 로봇의 소리, 음악 소리 등을 감지할 수 도 있습니다. 이를 이용하여 다양한 프로그래밍을 할 수 있습니다.

3. play(연주하기) 동작기

음악을 이용하면 다양한 분위기를 낼 수 있습니다. play 동작기를 이용하면 코뮤티에서 지원하는 다양한 효과음을 이용하여 더욱 재미있는 프로그램을 만들 수 있습니다.

4. express(표현하기) 동작기

로봇이 감정을 갖고 있다면 어떨까요? 동작기, 센서로는 표현하기 힘든 감정을 재미있는 표정과 아이콘을 이용해서 표현할 수도 있습니다.

※ 문제해결하기

1. 위의 4가지 동작기, 센서를 이용하여 로봇끼리 간단한 대화를 하는 상황을 재미있게 만들어 봅시다.

<부록 4> 스크래치 프로그래밍 교재



상상·프로그래밍·공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아갑니다!

SCRATCH반 ()학년 이름 ()

First-Day

1교시: 09:00~09:50 (50분)

✓ 메뉴 살펴보기 (5분)

- ① 8가지 블록의 종류 알아보기
- ② 파일메뉴: 저장하기, 다른 이름으로 저장하기
- ③ 편집메뉴: 삭제취소

✓ 스프라이트 이용하기 (5분)

- ① 복사, 삭제, 확대, 축소 버튼 이용하기
- ② 무대화면 작게 보기, 무대화면 크게 보기, 프리젠테이션

✓ 블록 복사하고 옮기는 방법 익히기 (5분)

- ① 스프라이트 선택(Friction Marble)하여 이용해보기
- ② 조작 연습하기(블록을 옮기거나 복사, 삭제하는 방법 익히기)



✓ 프로젝트1 (5분) : 계속 걷는 고양이


단 계 1	단 계 2
<pre> ▶ 클릭되었을 때 무한반복 { 다음 모양 } </pre>	<pre> ▶ 클릭되었을 때 무한반복 { 다음 모양 (10)만큼 움직이기 } </pre>
단 계 3	
<pre> ▶ 클릭되었을 때 무한반복 { 벽에 닿으면 튕기기 다음 모양 (10)만큼 움직이기 } </pre>	

✓ 프로젝트2 (5분) : 계속 걷는 고양이2 (병렬 프로그래밍의 이해)

단 계 1		
<pre> ▶ 클릭되었을 때 무한반복 { 벽에 닿으면 튕기기 } </pre>	<pre> ▶ 클릭되었을 때 무한반복 { 다음 모양 } </pre>	<pre> ▶ 클릭되었을 때 무한반복 { (10)만큼 움직이기 } </pre>

✓ 프로젝트3 (5분) : 움직이는 사람

- ① 스프라이트 직접 그리기 (얼굴)
- ② 모양을 이용하여 애니메이션 효과 주기
- ③ 움직이는 효과를 주기 위해서는 이미 배웠던 무한반복을 사용해본다.
- ④ 너무 빨리 움직일 때를 위해서 (1)초 기다리기를 사용한다.

단 계 1	모양1	모양2
<pre> ▶ 클릭되었을 때 무한반복 { 벽에 닿으면 튕기기 다음 모양 (10)만큼 움직이기 (0.1)초 기다리기 } </pre>		

※ 기존 스프라이트 이용하기: People > pacer

✓ 프로젝트4 (5분) : 큰 사각형을 그리며 움직이기

- ① 각도 이해하기, 90도 이해하기, 오른쪽으로 돌기와 왼쪽으로 돌기 이해하기
- ② 스프라이트 선택 (cat2)

단 계 1	단 계 2
<u>▶ 클릭되었을 때</u> (100)만큼 움직이기 (90)왼쪽 돌기	<u>▶ 클릭되었을 때</u> 무한반복 { (100)만큼 움직이기 (90)왼쪽 돌기 }

✓ 프로젝트5 (5분) : 방향보기

90: 오른쪽, -90: 왼쪽, 0: 위, 180: 아래쪽 보기 익히기

단 계 1
<u>▶ 클릭되었을 때</u> (90)도 방향 보기 (1)초 기다리기 (-90)도 방향 보기 (1)초 기다리기 (0)도 방향 보기 (1)초 기다리기 (180)도 방향 보기



✓ 프로젝트6 (5분) : 스프라이트 이름 만들기

- ① ()쪽 보기 이해하기
- ② 해당 스프라이트쪽으로 이동하기 위한 방법 찾기
- ③ ()위치로 가기

✓ 프로젝트7 (5분) : 이벤트에 따라 움직이기

- ① ▶ 클릭되었을 때
- ② 키 눌렀을 때
- ③ 클릭되었을 때

▶ 클릭되었을 때	[스페이스]키 눌렀을 때	스프라이트] 클릭되었을 때
<u>▶ 클릭되었을 때</u> (10)만큼 움직이기	<u>[스페이스]키 눌렀을 때</u> (10)만큼 움직이기	<u>스프라이트] 클릭되었을 때</u> (10)만큼 움직이기

First-Day

2교시: 10:00~10:50 (50분)

✓ 프로젝트1 (10분) : 무한반복, 반복 (10)회 차이점 알기

① 100만큼 이동하기 위한 방법 찾기

방법 1	방법 2	방법 3
▶ 클릭되었을 때 (1)만큼 움직이기 × 100번	▶ 클릭되었을 때 (5)만큼 움직이기 × 20번	▶ 클릭되었을 때 (100)만큼 움직이기

② 반복문과의 차이점 알기

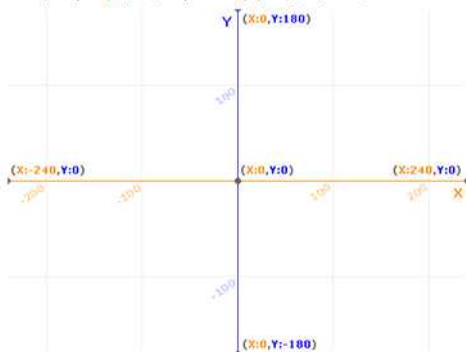
방법 1	방법 2	방법 3
▶ 클릭되었을 때 반복 (100회) { (1)만큼 움직이기 }	▶ 클릭되었을 때 반복 (20회) { (5)만큼 움직이기 }	▶ 클릭되었을 때 (100)만큼 움직이기

③ 이런 것도 가능한가요? 큰 사각형을 그리는 움직임

큰 사각형을 그리는 움직임	더 나은 방법	더 멋지게?
▶ 클릭되었을 때 반복 (4회) { (100)만큼 움직이기 (90)도 왼쪽 돌기 }	▶ 클릭되었을 때 반복 (4회) { 반복 (10회) { (1)만큼 움직이기 } (90)도 왼쪽 돌기 }	▶ 클릭되었을 때 반복 (4회) { 반복 (10회) { (1)만큼 움직이기 } 반복 (9회) { (10)도 왼쪽 돌기 } }

✓ 프로젝트2 (10분) : X좌표, Y좌표 이해하기

① 무대에서 좌표 배경 불러오기



- ② 좌표를 이용하여 좌표쪽으로 이동하기
X: (), Y: () 쪽으로 가기
- ③ 1초 동안 X: (0), Y: (0)쪽으로 움직이기
- ④ X좌표 (10)만큼 바꾸기 / Y좌표 (10)만큼 바꾸기
- ⑤ X좌표 (0)로 정하기 / Y좌표 (0)로 정하기

단 계 1	단 계 2	단 계 3	단 계 4
<pre> ▶ 클릭되었을 때 반복 (10)회 { X좌표 (10)만큼 바꾸기 } </pre>	<pre> ▶ 클릭되었을 때 반복 (10)회 { X좌표 (-10)만큼 바꾸기 } </pre>	<pre> ▶ 클릭되었을 때 무한반복 { 반복 (10)회 { X좌표 (10)만큼 바꾸기 } 반복 (10)회 { X좌표 (-10)만큼 바꾸기 } } </pre>	<pre> ▶ 클릭되었을 때 무한반복 { 반복 (10)회 { X좌표 (10)만큼 바꾸기 } 반복 (10)회 { Y좌표 (-10)만큼 바꾸기 } 반복 (10)회 { X좌표 (-10)만큼 바꾸기 } 반복 (10)회 { Y좌표 (10)만큼 바꾸기 } } </pre>

✓ 프로젝트3 (10분) : < >에 닿기? 이용하기

'만약 < > 라면 + []에 닿기?'를 이용하여 제어하기

해석: 만약 []에 닿기라면~으로 해석

해석: 만약 []에 닿기라면~ 아니면~

해석: 반복 < > 까지

방 법 1	방 법 2	방 법 3
<pre> ▶ 클릭되었을 때 무한 반복 { 만약 <스프라이트에 닿기?> 라면 { [안녕] (1)초 동안 말하기 } (10)만큼 움직이기 } </pre>	<pre> ▶ 클릭되었을 때 무한 반복 { 만약 <스프라이트에 닿기?> 라면 { [안녕] (1)초 동안 말하기 } 아니면 { (10)만큼 움직이기 } } </pre>	<pre> ▶ 클릭되었을 때 반복 <스프라이트에 닿기?> 라면 { (10)만큼 움직이기 } [안녕] (1)초 동안 말하기 </pre>

① 스크립트 멈추기

② 모두 멈추기 이용하기 (스프라이트 여러개로 테스트하면 됨)

✓ 프로젝트4 (10분) : 화살표 키 눌렀을 때
 화살표 키 누르면 움직이는 방법

방법 1	방법 2	방법 3	방법 4
<u>(위쪽화살표)키 눌렀을 때</u> y좌표 (10)만큼 바꾸기	<u>(위쪽화살표)키 눌렀을 때</u> (0)도 방향 보기 (10)만큼 움직이기	※만약 < >라면 이용하기	
<u>(아래쪽화살표)키 눌렀을 때</u> y좌표 (-10)만큼 바꾸기	<u>(아래쪽화살표)키 눌렀을 때</u> (180)도 방향 보기 (10)만큼 움직이기	방법 1과 방법 2와 같은 방법에서 이벤트를 '▶클릭되었을 때' 로 바꾸고, 만약 < >라면을 사용한다.	
<u>(오른쪽화살표)키 눌렀을 때</u> x좌표 (10)만큼 바꾸기	<u>(오른쪽화살표)키 눌렀을 때</u> (90)도 방향 보기 (10)만큼 움직이기		
<u>(왼쪽화살표)키 눌렀을 때</u> x좌표 (-10)만큼 바꾸기	<u>(왼쪽화살표)키 눌렀을 때</u> (-90)도 방향 보기 (10)만큼 움직이기		



First-Day

3교시: 11:00~11:50 (40~50분)

형태와 펜 / 소리

- ✓ 프로젝트1 (5분) : 색깔효과 ()만큼 바꾸기 / 색깔효과 ()만큼 주기
 - 색깔, 어안렌즈, 소용돌이, 픽셀화, 모자이크, 밝기, 반투명 효과 만들기
 - 그래픽 효과 지우기

- ✓ 프로젝트2 (5분) : 크기 ()만큼 바꾸기 / 크기 ()%로 정하기
 - 크기 바꾸기 / 정하기
 - 맨 앞으로 나오기 / (1)번째로 물러나기

※ 기존 스프라이트 사용하기: people > Gymnast , whirling_girl

- ✓ 프로젝트3 (5분) : 펜 블록 기능 알아보기
 - 펜 내리기 / 펜 올리기
 - 조종하여 그리기

방 법 1	방 법 2
▶클릭되었을 때 펜 내리기 무한반복 { 만약 <위쪽화살표>키 클릭하기? 라면 { y좌표 (10)만큼 바꾸기 } 만약 <아래방향 화살표>키 클릭하기? 라면 { y좌표 (-10)만큼 바꾸기 } 만약 <왼쪽화살표>키 클릭하기? 라면 { x좌표 (-10)만큼 바꾸기 } 만약 <오른쪽화살표>키 클릭하기? 라면 { x좌표 (10)만큼 바꾸기 } } }	[스페이스]키 눌렀을 때 펜의 색 (10)만큼 바꾸기 or [스페이스]키 눌렀을 때 펜의 색 (10)만큼 바꾸기 펜의 크기 (1)만큼 바꾸기

- 스탬프 이용하기

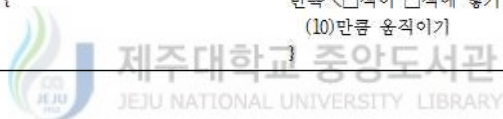
- ✓ 프로젝트4 (5분) : 소리 내기
 - [] 소리내기
 - 끝날때까지 [] 소리내기
 - 모든 소리 끄기
- ※ 녹음한 음악 사용하기

방 법 1	Clickable Drum 스프라이트
위의 코드를 그대로 이용하여 움직일 때마다 소리를 내는 방법	<u>드럼 클릭되었을 때</u> Drum 소리내기 or <u>드럼 클릭되었을 때</u> 끝날때까지 Drum 소리내기

- ✓ 프로젝트4 (5분) : 소리 내기
 - [] 소리내기
 - 끝날때까지 [] 소리내기
 - 모든 소리 끄기
- ※ 기존 스프라이트 사용하기: people > dancer

- ✓ 프로젝트5 (5분) : 관찰 블록 이용하기
 - 반복 <□색에 닿기?> 까지

방 법 1	방 법 2
<u>▶클릭되었을 때</u> 반복 <□색에 닿기?> 까지 { (10)만큼 움직이기 }	<u>▶클릭되었을 때</u> 반복 <□색이 □색에 닿기?> 까지 { (10)만큼 움직이기 }



- ✓ 미디어 아트 (10분)
 - 난수 이용하기
 - 튕기기 효과를 이용해서 움직이면서 그래픽 효과 내기
(Basketball 스프라이트로 움직이며 펜으로 그리기)
 - 소리내기

예	
<u>▶클릭되었을 때</u> 펜 내리기 펜의 크기 (1)로 정하기 무한반복{ 펜의 크기 ((-10)부터 (10)사이의 난수)만큼 바꾸기 펜의 색 (10)만큼 바꾸기 벽에 닿으면 튕기기 (10)만큼 움직이기 }	<u>▶클릭되었을 때</u> 무한반복 { 끝날때까지 [HumanBeatbox1] 소리내기 }

First-Day 4교시: 13:00~13:50 (50분)

✓ 방송하기 익히기 (20분)

① 버튼을 누르면 말하기

버튼 스프라이트	사람 스프라이트
버튼 클릭되었을 때 [하이] 방송하기	[하이] 받을 때 [안녕] (1)초동안 말하기

② 달리기 시합

- 버튼을 눌렀을 때 방송하고 두 사람이 달리기 시작하자.

버튼 스프라이트	사람 스프라이트
버튼 클릭되었을 때 [달리기시작] 방송하기	[달리기시작] 받을 때 반복 <벽에 닿기?> 까지 { (10)만큼 움직이기 }

③ 릴레이 달리기 시합

- 위 달리기 시합에서 두 개의 '사람' 스프라이트 추가

사람 스프라이트	사람2 스프라이트	사람3 스프라이트
[달리기시작] 받을 때 반복 <사람2에 닿기?> 까지 { (10)만큼 움직이기 }	[2번뛰어] 받을 때 반복 <사람3에 닿기?> 까지 { (10)만큼 움직이기 }	[3번뛰어] 받을 때 반복 <벽에 닿기?> 까지 { (10)만큼 움직이기 }
[2번뛰어] 방송하기	[3번 뛰어] 방송하기	

④ 헬리콥터 조종하기

- 버튼 4개, 헬리콥터 스프라이트 만들기

헬리콥터	버튼 좌	버튼 우	버튼 위	버튼 아래
[위] 받을 때 ... [아래] 받을 때 ...	[버튼 좌] 눌렀을 때 [좌] 방송하기	[버튼 우] 눌렀을 때 [우] 방송하기	[버튼 위] 눌렀을 때 [위] 방송하기	[버튼 아래] 눌렀을 때 [아래] 방송하기

✓ 변수의 개념 이해하기 (10분)

① 변수의 뜻 알기

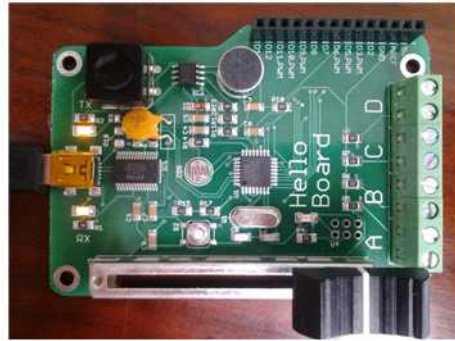
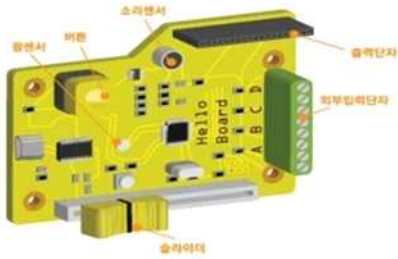
② 변수 만들기

- 타이머 만들기 / 초기화
- 변수에 타이머 넣기
- 변수 보기

③ []을 묻고 기다리기 + 연산

예 1	예 2
▶클릭되었을 때 [10+20 ?]을 묻고 기다리기 만약 < 대답 = 30 > 라면{ "정답입니다." (1)초동안 말하기 }	▶클릭되었을 때 [가]에 (1)저장 [나]에 (2)저장 [다]에 ((가)+(나)) 저장 (가)+(나) (1)초동안 말하기

✓ 센서 알기



◆HelloBoard

✓ 슬라이더 센서와 소리센서 이용하기 (20분)

- ① 슬라이더 센서, 소리센서의 범위: 0~100
- ② 변수 만들기

예 1	예 2
<pre> ▶클릭되었을 때 무한 반복{ 벽에 닿으면 핑기기 [슬라이더 센서값]만큼 움직이기 } </pre>	<pre> ▶클릭되었을 때 지우기 펜 내리기 펜의 색 □로 정하기 펜의 크기 (10)로 정하기 무한 반복 { 15도 왼쪽 돌기 (슬라이더 센서값)만큼 움직이기 펜의 그림자 (10)만큼 바꾸기 펜의 크기 (소리 센서값)로 정하기 } </pre>

③ X좌표, Y좌표로 센서값 이용하기

예 1	예 2
<pre> ▶클릭되었을 때 지우기 무한 반복{ X좌표 (슬라이더 센서값)로 정하기 } </pre>	<pre> ▶클릭되었을 때 지우기 무한 반복{ X좌표 (슬라이더 센서값)로 정하기 Y좌표 (소리 센서값)로 정하기 } </pre>

✓ 광 센서와 버튼 이용하기 (10분)

- ① 무대 Nature 그림 넣기
- ② 광센서 이용하기: 범위 0~100
- ③ 버튼 이용하기: True 또는 False

예 1 (배경에 스크립트 넣기)	예 2
<pre> ▶클릭되었을 때 무한 반복{ 밝기 효과 (빛 센서값 - (50))만큼 주기 } </pre>	<pre> ▶클릭되었을 때 무한 반복{ 만약 <센서의 버튼 누르기?> 라면 { 벽에 닿으면 핑기기 (10)만큼 움직이기 } } </pre>




제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

First-Day 5교시: 14:00~14:50 (50분)

✓ 프로젝트1: 기존 스프라이트 이용하기 (30분)

- ① Animal > hungry_fish
 - 마우스 포인터 사용하기
- ② Animal > running_ant
 - 화살표 키보드 사용하기
- ③ Animal > crawling_ant
 - 쪽 움직이는 개미
- ④ Transportation > rotating_car
 - 각도로 움직이는 자동차

자동차 (기본 rotating_car)	예시 그림
<pre>▶클릭되었을 때 [출발] 위치로 가기 무한반복 { 만약 <□색에 닿기?> 라면 { [출발] 위치로 가기 } }</pre>	

※ 무대 바꾸기로 다음 단계를 만들어보자.

- ④ JetPack_Girl
 - 로켓 가방 메고 다니는 우주인

JetPack_Girl	예시 그림
<ul style="list-style-type: none">- 원래 JetPack_Girl 은 검은색에 닿으면 멈춤- 본 게임은 출발지점에서 도착지점까지 가는 것이 목표	

※ 무대 바꾸기로 다음 단계를 만들어보자.

- ✓ 프로젝트1: 눈 내리는 날 (15분)
- 스프라이트: 눈사람, 땅, 하늘, 눈1...

눈1	예시 그림
<pre> ▶클릭되었을 때 무한 반복{ 보이기 X: ((-240)부터 (240)사이의 난수), Y: ((하늘)의 Y좌표) 쪽으로 가기 반복 <땅에 닿기?> 까지 { (180)도 방향 보기 (10)만큼 움직이기 } 숨기기 ((1)부터 (3)사이의 난수)초 기다리기 } </pre>	

- ✓ 프로젝트2: 내리는 눈을 먹는 강아지 (15분)
- 스프라이트: 강아지

강아지	예시 그림
<pre> ▶클릭되었을 때 무한 반복{ X좌표 ((슬라이더 센서값 * 4.8) - 240)로 정하기 } </pre>	
<p>눈1에 추가</p> <pre> ▶클릭되었을 때 [점수]에 [0]저장 무한 반복{ 만약 <강아지에 닿기?> 라면{ [GOOD] (0.5) 초동안 말하기 [점수]에 (1)씩 누적하기 숨기기 } } </pre>	

폭 탄	사라지기 버튼 센서
<pre> ▶클릭되었을 때 무한 반복{ 만약 <강아지에 닿기?> 라면{ 모두 멈추기 ● } } </pre>	<pre> ▶클릭되었을 때 무한 반복{ 만약 <센서의 버튼 누르기?> 라면 숨기기 (2)초 기다리기 보이기 } </pre>

- 위 게임에서 슬라이더 센서를 빛 센서, 소리 센서로 바꾸어 게임 해보기

✓ 프로젝트3: 어두워지면 움직이는 동물들 (5분)

무 대	야행성 동물
<pre> ▶클릭되었을 때 무한 반복{ [밝기] 효과 ((빛 센서값)-70)만큼 주기 } </pre>	<pre> ▶클릭되었을 때 무한 반복{ 만약 < (빛 센서값 - 30) < 0 > 라면 { 벽에 닿으면 튕기기 (10)만큼 움직이기 }아니면{ (1)초 동안 X: (집의 X좌표), Y: (집의 Y좌표)쪽으로 움직이기 } } </pre>



First-Day 6교시: 15:00~15:50 (50분)

✓ 프로젝트1: 슛팅 비행기 (25분)

- 설계 및 프로그래밍 하기



스프라이트 #1 : 비행기	#2 : 총알	#3 : UFO	#4 : 하늘
 <p>슬라이더 센서값으로 조종한다.</p>	 <p>버튼 누르기가 되면 총알이 나간다.</p>	 <p>하늘에서 내려오는 UFO, 내 비행기가 부딪히면 게임종료.</p>	 <p>우주그림과 비슷한 하늘, 여기에서 UFO들이 나온다.</p>

비행기 스프라이트	UFO 스프라이트
<pre> ▶클릭 되었을 때 무한반복 { X좌표 (((슬라이더 센서값) * 4.8) - 240)로 정하기 } </pre>	<pre> ▶클릭 되었을 때 숨기기 무한반복 { X: -(-240)부터 (240)사이의 난수, Y: (하늘의 y좌표)쪽으로 가기 보이기 반복 <벽에 닿기?> 까지 { Y좌표 (-10)만큼 바꾸기 } 숨기기 } </pre>
총알 스프라이트	
<pre> ▶클릭 되었을 때 숨기기 무한반복 { 만약 <센서의 [버튼누르기]?> 라면 { X좌표 (비행기의 X좌표)로 정하기 Y좌표 (비행기의 Y좌표)로 정하기 보이기 반복 <벽에 닿기?> 까지 { Y좌표 (10)만큼 바꾸기 } } } 숨기기 } </pre>	<pre> ▶클릭 되었을 때 무한반복 { 만약 <비행기에 닿기?> 라면 { 모두 멈추기 ● } } ▶클릭 되었을 때 무한반복 { 만약 <총알에 닿기?> 라면 { 숨기기 } } } </pre>

✓ 프로젝트2: 점프하기 이용하여 웹빙 건강관리 게임 만들기 (25분)

고양이 스프라이트
<pre> ▶클릭 되었을 때 무한반복 { 만약 <센서의 [버튼누르기]? > 라면 { [중력]에 (0) 저장 반복 (31) 회 { Y좌표 ((15)-중력)만큼 바꾸기 [중력]에 (1)씩 누적하기 } } } } </pre>

점프하기를 이용하여 웹빙식사로 건강 챙기기 게임 만들기

<부록 5> 코듀 프로그래밍 중심 STEAM 교육 프로그램 교재

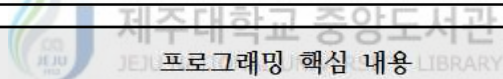


KODU반 ()학년 이름 ()

[오늘의 주제] 지구가 뜨거워져요

공부할 문제

1. 지구 온난화가 발생하는 원인을 알고 우리 주변에서 온실가스가 발생하는 장소를 찾아봅시다.
2. 온실가스의 발생을 줄이기 위해서 우리가 할 수 있는 일을 말해봅시다.
3. 자기만의 창의적인 아이디어로 지형과 객체를 디자인하고 코듀를 이용하여 온실가스를 줄이는 기능성 게임을 프로그래밍 해 봅시다.
4. 직접 구현한 기능성 게임을 이용하여 환경 보호를 주제로 하는 디지털 스토리텔링을 해 봅시다.




프로그래밍 핵심 내용

핵심 컴퓨터 원리 (KODU 핵심 기능)	creatable
사용할 센서	see, bump, keyboard, timer
사용할 동작기	eat, create, move, score

[메모]

1. 이야기 들려주기



[생각해 봅시다]

1. 북극곰에게 얼음덩어리 또는 빙하가 필요한 이유는 무엇인가요?
2. 북극곰은 어떤 어려움을 겪고 있나요?
3. 제목을 '얼음 위를 걷고 싶어요' 라고 지은 이유는 무엇일까요?

2. 이야기 속 문제 찾기

- (1) 빙하가 녹고 있는 이유는 무엇일까요?
- (2) 지구가 점점 뜨거워지는 이유는 무엇인가요?
- (3) 지구 온난화와 온실 가스



지구 온난화

지구의 대기(공기)와 바다의 평균 온도가 꾸준히 상승하는 현상을 **지구 온난화**라고 합니다. 지구 온난화로 가뭄, 홍수 등 기상이변이 발생합니다.

온실 가스

공기 중에는 사람이 숨을 쉬는데 사용되는 산소 이외에도 다양한 기체가 있습니다. 그 중에서도 이산화탄소, 메탄 등이 주변의 열을 흡수하여 지구 온난화의 원인이 됩니다. 이렇게 지구 온난화의 원인이 되는 이 기체들을 **온실 가스**라고 합니다.

- (4) 우리 주변에서 찾을 수 있는 온실가스의 발생

(5) 온실가스를 줄이기 위해서 우리가 할 수 있는 일은 무엇이 있을까요?

3. 프로그래밍 문제

오늘 만들 프로그램 : 온실가스를 제거하라.

[게임 설명]

나무는 온실가스인 이산화탄소를 흡수하여 지구 온난화를 예방하는 역할을 합니다. 나무를 심어 공장, 자동차에서 생겨나는 이산화탄소를 제거하는 게임을 만들어 봅시다.

[필수 조건]

1. 주인공 로봇은 키보드의 방향키를 이용하여 이동하고, 나무를 심는다.
2. 이산화탄소는 구름을 이용한다. 구름은 creatable로 만들고 자유롭게 움직인다.
3. 공장, 자동차는 이산화탄소를 만들어 낸다.
4. 이산화탄소를 흡수하는 나무는 5개 이상 만들 수 없다.
5. 나무는 가까이에 있는 이산화탄소만 흡수할 수 있다.
6. 이산화탄소가 10개 이상이면 지구온난화를 막지 못하였으므로 게임 실패, 흡수한 이산화탄소가 10개 이상이면 게임에서 승리한다.(적절한 종료조건은 직접 생각해낼 것)

4. 설계하기

전체 프로그래밍 디자인하기	객체 1 :	객체 2 :
	객체 3 :	객체 4 :

5. 프로그래밍 하기

여러분이 설계한 프로그램을 직접 프로그래밍 해 봅시다.

“온실가스를 제거하라” 프로그래밍 하기 - 여러분이 작성한 프로그래밍의 과정을 적어봅시다.(센서, 동작기)

6. 창의적인 요소를 넣어 이야기 만들기

나만의 이야기 만들기



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

7. 디지털 스토리텔링 하기

여러분이 만든 이야기를 친구들 앞에서 발표해 봅시다.

8. 정리하기

(1) 오늘 지구 온난화에 대해서 공부하고 프로그래밍을 하면서 느낌 소감을 발표해 봅시다.

(2) 지구 온난화를 막기 위해 여러분이 실천할 다짐을 적어봅시다.



KODU반 ()학년 이름 ()

[오늘의 주제] 생태계를 지켜라

공부할 문제

1. 생태계가 위협받는 원인을 알고 생물과 생태계를 보호하기 위해 우리가 할 수 있는 노력을 찾아 실천해 봅시다.
2. 생태계를 보호하겠다는 의지를 담아 해양 오염과 천적으로부터 해양 생태계를 지키는 프로그램을 디자인하고 코드를 이용하여 프로그래밍 해 봅시다.
3. 직접 구현한 프로그램을 이용하여 생태계 보호를 주제로 하는 디지털 스토리텔링을 해 봅시다.

프로그래밍 핵심 내용

핵심 컴퓨터 원리 (KODU 핵심 기능)	creatable, page, path
사용할 센서	see, bump, keyboard, timer, play, glow, say
사용할 동작기	eat, create, move, score, shoot, win, end

[메모]

1. 이야기 들려주기

[생각해 봅시다]



1. 하프물범의 처음 보았을 때 어떠한 생각이 들었나요?
2. 왜 하프물범을 몰래 사냥하는 사람들이 있을까요?
3. 하프물범을 보호하기 위해서 우리가 할 수 있는 일은 어떤 것이 있을까요?

2. 이야기 속 문제 찾기

(1) 다음 용어의 뜻을 찾아 적어봅시다.

① 생물 :

② 자연환경 :

③ 생태계 :

④ 생태계 평형 :

⑤ 천적 :



(2) 생태계에 속하는 생물이나 환경에는 어떠한 것이 있을까요?

[참고]

어떤 장소에서 살아가는 모든 생물과 환경이 상호작용하는 것을 _____라고 합니다.

(3) 생물과 생태계가 위협받는 원인에는 어떠한 것이 있을까요?

(4) 생물과 생태계가 파괴되는 것은 인류에게 어떤 영향을 미칠까요?

(5) 생물과 생태계를 보호하기 위해서 우리가 할 수 있는 일은 무엇이 있을까요?

3. 프로그래밍 문제

오늘 만들 프로그램 : 천연기념물 물고기를 지켜라!

[게임 설명]

천연기념물 물고기가 멸종 위기에 빠졌습니다. 물고기에게는 신선한 먹이를 주어야 합니다. 오래되어 썩은 먹이와 물고기를 잡아먹는 괴물 물고기로부터 물고기를 보호해야 합니다.

[필수 조건]

1. 물고기는 평상시에는 특정한 경로만 이동한다.
2. 주인공 로봇배는 배가 고픈 물고기들에게 먹이(사과)를 주어야 한다.
3. 먹이는 10초 동안은 신선하지만 10초가 지나면 썩어서 검게 변하고 이를 먹으면 물고기가 죽는다.
4. 물고기는 20초가 지나면 배가 고프고 배가 고프 상태에서 15초가 지나면 죽는다.
5. 돌멩이로부터 나오는 괴물 물고기는 천연기념물의 천적으로 물고기를 잡아먹는다.
6. 로봇배는 괴물 물고기가 천연기념물 물고기를 잡아먹기 전에 격추해야 한다.

4. 설계하기

전체 프로그래밍 디자인하기  제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY	객체 1 :	객체 2 :
	객체 3 :	객체 4 :

5. 프로그래밍 하기

여러분이 설계한 프로그램을 직접 프로그래밍 해 봅시다.

“천연기념물 물고기를 지켜라!” 프로그래밍 하기 - 여러분이 작성한 프로그래밍의 과정을 적어봅시다.(센서, 동작기)

6. 창의적인 요소를 넣어 이야기 만들기

나만의 이야기 만들기



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

7. 디지털 스토리텔링 하기

여러분이 만든 이야기를 친구들 앞에서 발표해 봅시다.

8. 정리하기

(1) 오늘 생태계 보호에 대해서 공부하고 프로그래밍을 하면서 느낌 소감을 발표해 봅시다.

(2) 생태계를 보호하기 위해 여러분이 실천할 다짐을 적어봅시다.



KODU반 ()학년 이름 ()

[오늘의 주제] 에너지를 절약해요

공부할 문제

1. 에너지의 중요성을 알고 에너지를 절약해야하는 이유를 알아봅시다.
2. 에너지를 절약하기 위해서 우리가 할 수 있는 일을 말해봅시다.
3. 자기만의 창의적인 아이디어로 지형과 객체를 디자인하고 코드를 이용하여 에너지를 관리하는 기능성 게임을 프로그래밍 해 봅시다.
4. 직접 구현한 기능성 게임을 이용하여 에너지 절약을 주제로 하는 디지털 스토리텔링을 해 봅시다.

프로그래밍 핵심 내용

핵심 컴퓨터 원리 (KODU 핵심 기능)	creatable, page, hit points
사용할 센서	see, bump, keyboard, timer, health, scored, held by
사용할 동작기	eat, create, move, set score, score, grab, drop, damage, reset, color

[메모]

1. 이야기 들려주기

[생각해 봅시다]



1. 에너지란 무엇일까요?
2. 우리가 쓰고 있는 에너지에는 어떤 것들이 있나요?
3. 화석에너지를 대신할 수 있는 신·재생 에너지에는 어떤 것들이 있나요?

2. 이야기 속 문제 찾기

(1) 에너지는 왜 중요한가요?

(2) 왜 에너지를 절약해야 하나요?

(3) '블랙아웃'의 뜻을 인터넷에서 찾아봅시다.

(4) 에너지 절약 홈페이지(<http://www.powersave.or.kr/>)에 들어가서 현재 예비전력이 얼마인지 알아봅시다.

(5) 에너지가 없어졌을 때 일어날 수 있는 일들을 실생활과 관련해서 상상하여 써 봅시다.

(6) 에너지 절약을 위해 우리가 할 수 있는 일은 어떤 것이 있을까요?



3. 프로그래밍 문제

오늘 만들 프로그램 : 에너지를 관리하자! (블랙아웃을 막자!)

[게임 설명]

우리 생활 거의 모든 부분에 에너지가 사용됩니다. 특히 전기에너지는 이제는 없어서는 안 될 존재가 되었습니다. 하지만 사람들이 무분별하게 전기를 사용하게 되면 발전소에서 만들어 내는 전기가 부족해져서 블랙아웃(대규모 정전상태)이 발생합니다. 그런 일이 안 일어나도록 에너지를 관리하는 게임을 만들어 봅시다.

[필수 조건]

1. 게임을 시작하면 에너지 점수 50점을 주며, 100점 이상 되면 승리, 0점 이하가 되면 패배합니다.
2. 주인공 로봇은 키보드의 방향키를 이용하여 이동하고, 사과 운반 로봇을 치료해 줍니다.
3. 나무에서는 10초마다 에너지원인 사과가 생산됩니다.
4. 사과 운반 로봇은 나무에서 나온 사과를 주워 발전소에 가져다주고 발전소는 사과를 먹고 에너지 점수 5점을 얻습니다.
5. 사과 운반 로봇은 5초마다 에너지 점수 1점을 쓰고, 1초마다 체력이 1점씩 줄어듭니다.
6. 사과 운반 로봇의 체력이 20점 이하가 되면 움직임을 멈추고, 0점 이하가 되면 폭발합니다.
7. 발전소에서 사과 운반 로봇을 만들 수 있고, 이것을 만드는데 에너지 점수 10점이 쓰입니다.
8. 발전소는 3초마다 에너지 점수 1점을 사용합니다.

4. 설계하기

전체 프로그래밍 디자인하기  제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY	객체 1 :	객체 2 :
	객체 3 :	객체 4 :



KODU반 ()학년 이름 ()

[오늘의 주제] 우주 생명의 흔적을 찾아라

공부할 문제

1. 여러 나라들이 우주 과학을 연구하는 이유를 알고 우주 과학의 중요성을 알 수 있다.
2. 행성 탐사 로봇의 하는 일을 알고 행성탐사 계획을 세울 수 있다.
3. 행성 탐사 로봇을 설계하고 프로그래밍해서 자신의 프로그램을 발표해 봅시다.

프로그래밍 핵심 내용

핵심 컴퓨터 원리
(KODU 핵심 기능)

사용할 센서

사용할 동작기

제주대학교 중앙도서관
EJU NATIONAL UNIVERSITY LIBRARY

1. 이야기 듣기



[생각해 봅시다]

1. 무엇에 대한 동영상인가요?
2. 왜 이름을 큐리오시티라고 지었을까요?
3. 동영상을 본 느낌이 어떤가요?

[좀 더 자세한 활동 모습 보기]

<http://www.youtube.com/watch?v=P4boyXQuUtw&feature=related>

2. 이야기 속 문제 찾기

- (1) 왜 전 세계의 여러 나라들이 우주 개발에 관심을 갖고 있을까요?
- (2) 행성을 탐사 계획을 세울 때 미리 알고 있어야 하는 것에는 무엇이 있을까요?
- (3) 화성에 대해 조사해 봅시다.
 - ① 위치 :
 - ② 모양 :
 - ③ 크기 :
 - ④ 색깔 :
 - ⑤ 환경(표면, 생명체, 물/공기)

※ 참고 - 큐리오시티의 모습과 각 부분의 역할



3. 프로그래밍 문제

오늘 만들 프로그램 : 행성탐사로봇 만들기



[게임 설명]



여러분의 행성탐사로봇을 제작하는 프로젝트에서 일하는 컴퓨터 과학자입니다. 여러분은 최근에 발견된 한 행성을 탐사하기 위해 탐사로봇을 프로그래밍 해야 합니다. 여러분의 생명의 흔적과 물의 흔적을 찾고 행성 주변을 탐색하는 로봇을 만들어야 합니다.

[필수 조건]

1. 행성 주변이 매우 어둡기 때문에 탐사를 위해서는 주변을 밝힐 수 있는 방법을 생각한다.
2. 물이나 생명의 흔적을 찾으면 지구로 교신을 보낸다.
3. 높은 언덕 등의 지형을 탐색하기 위한 방법을 생각한다.
4. 탐사로봇이 탐사 물질을 찾으면 우주선으로 갖고 와야 한다.
5. 게임이 끝나는 조건을 생각해 본다 - 예) 250초 이내에 행성에 있는 생명의 흔적을 5개 이상 찾으면 승리한다.

4. 설계하기 - 어떤 로봇(객체)을 사용할지, 각 로봇이 어떠한 일을 수행할지 자세하게 적어봅니다.

전체 프로그래밍 디자인하기	객체 1 :	객체 2 :
	객체 3 :	객체 4 :



5. 프로그래밍 하기

여러분이 설계한 프로그램을 직접 프로그래밍 해 봅시다.

“행성탐사로봇 만들기 ” 프로그래밍 하기 - 가장 중요한 부분의 센서와 동작기를 적어보세요.



KODU반 ()학년 이름 ()

[오늘의 주제] 안전 교통 도시 설계하기


공부할 문제

1. 안전한 교통 도시를 만드는 데 필요한 장치와 규칙에는 어떠한 것들이 있는지 알아봅시다.
2. 적절한 속력이 교통 안전과 밀접하게 관련되어 있음을 알아봅시다.
3. 자기만의 창의적인 아이디어로 지형과 객체를 디자인하고 코드를 이용하여 교통 안전 도시를 설계하고 프로그래밍 해 봅시다.

프로그래밍 핵심 내용

핵심 컴퓨터 원리 (KODU 핵심 기능)	creatable, path, random
사용할 센서	timer, see, hear, on land
사용할 동작기	move, color, say

1. 이야기 들려주기



[생각해 봅시다]

1. 무엇에 대한 동영상인가요?
2. 교통사고는 왜 발생할까요?
3. 속력과 교통사고는 어떤 관계가 있을까요?

2. 이야기 속 문제 찾기

(1) 안전한 교통 도시를 만드는 데 필요한 장치나 규칙에는 어떠한 것들이 있나요?

(2) 스쿨존은 무엇인가요?

(3) 왜 스쿨존을 만들어 운영하면 교통사고가 줄까요?

(4) 대표적인 교통 표지판을 그려보고 이름과 의미를 찾아 적어 봅시다.

그림				
이름				
의미				

(5) 교통 안전을 실천하기 위해 각 대상이 지켜야 할 것에는 어떠한 것이 있나요?

- 운전자 :

- 보행자 :

(6) 교통 안전을 위해 여러분이 실천할 수 있는 것을 찾아보고 실천 다짐을 적어 봅시다.

3. 프로그래밍 문제

- (1) 횡단보도 만들기
- (2) 고가도로 만들기
- (3) 스쿨존 만들기
- (4) 신호등 만들기

오늘 만들 프로그램 : 교통 안전 도시를 설계해 보자

[프로그램 설명]

위에서 만들어본 다양한 교통장치의 요소를 이용하여 교통안전 도시를 만들고 설계하여 보자.

[필수 조건]

1. 자동차가 다니는 길은 경로를 사용한다.
2. 사고가 발생했을 경우에는 score 동작기를 이용하여 -1점 한다.
3. 종료조건은 사건 발생의 횟수 등을 고려하여 작성한다.
3. 위에 제시된 횡단보도, 고가도로, 스쿨존, 신호등 등이 가급적이면 모두 포함되게 프로그래밍 한다.
4. 보행자의 역할, 자동차의 역할을 맡을 캐릭터를 구분한다.

4. 설계하기

전체 프로그래밍 디자인하기	객체 1 :	객체 2 :
	객체 3 :	객체 4 :

5. 프로그래밍 하기

여러분이 설계한 프로그램을 직접 프로그래밍 해 봅시다.

“교통 안전 도시 설계하기” 프로그래밍 하기 - 여러분이 작성한 프로그래밍의 과정을 적어봅시다.

6. 창의적인 요소를 넣어 이야기 만들기

나만의 이야기 만들기



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

7. 디지털 스토리텔링 하기

여러분이 만든 이야기를 친구들 앞에서 발표해 봅시다.

8. 정리하기

(1) 교통 안전 도시를 설계하고 프로그래밍 한 소감을 발표해 봅시다.

(2) 교통안전을 실천하기 위해서 우리가 실천할 수 있는 일에 대하여 적어보자.



상상 · 프로그래밍 · 공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아갑니다!

SCRATCH반 ()학년 이름 ()

[오늘의 주제1] 밤이 되면 활동하는 동물들 (야행성 동물)

공부할 문제

1. 밤이 되면 활동하는 동물들과 그 동물들의 먹이는 어떤 것이 있는지 알아봅시다.
2. 깨끗한 환경에서만 살 수 있는 동물 중에서 야행성 동물에는 어떤 것이 있는지 알아봅시다.
3. 야행성 동물들은 어떠한 감각이 뛰어난지 생각해 봅시다.
4. 반딧불이가 살 수 있는 환경을 디자인하고, 밤이 되면 활동할 수 있도록 프로그래밍 해 봅시다.
5. 자신만의 아이디어가 들어간 이야기로 작품을 만들어 봅시다.

프로그래밍 핵심 내용

핵심 프로그래밍 원리


스크래치에서 사용할 그래픽효과

헬로보드에서 사용할 센서

[메모]

1. 이야기 들려주기

<http://study.jr.naver.com/donghwa/view.nhn?donghwaNo=700&categoryId=6&donghwaSort=regDate&ageType=ALL&viewAgeType=ALL>

	<p>[생각해 봅시다]</p> <ol style="list-style-type: none">1. 반딧불이는 어디에서 태어나고 어떻게 자라나요?2. 반딧불이 애벌레는 무엇을 먹나요?3. 반딧불이의 반짝이는 부분은 어디인가요?
---	--

2. 더 알아보아요.

<http://cafe.naver.com/hych/404>

(1) 환경오염과 반딧불이와는 어떤 관련이 있나요?

(2) 반딧불이가 우리 주변에서 찾아 볼 수 없는 이유는 무엇인가요?



3. 프로그래밍 문제

- 오늘 만들 프로그램: 반딧불이를 날게 하자.

[게임 설명]

환경이 깨끗한 숲속의 강가 주변에 반딧불이가 살고 있다. 이제껏 본 적 없는 반딧불과 만나는 체험을 해보자.

[필수 조건]

1. 자신의 안식처에 있던 반딧불이는 날이 어두워지면 빛을 내며 날아다닌다.
2. 날아다니는 동안 이슬을 먹으러 다닌다.
3. 날이 다시 밝아지면 안식처로 돌아오게 된다.

4. 설계 및 프로그래밍 하기

전체 프로그래밍 디자인하기 (그림으로 그려주세요)



스프라이트 #1 :	#2 :	#3 :	#4 :
#5 :	#6 :	#7 :	#8 :



상상 · 프로그래밍 · 공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아가입니다!

SCRATCH반 ()학년 이름 ()

[오늘의 주제2] 에너지 어떻게 사용하고 있나요?

공부할 문제

1. 우리의 습관 중에서 에너지를 낭비하고 있는 것이 있나요?
2. 에너지를 절약하는 것이 친환경과 관련이 있나요? 있다면 어떻게 관련있는지 말해봅시다.
3. 집에서 사용하는 에너지에는 어떤 것들이 있나요? 에너지를 절약하기 위한 노력은 어떻게 하고 있나요?
4. 집에서 에너지 절약을 위한 실천은 누가 하나요?

프로그래밍 핵심 내용

핵심 프로그래밍 원리

사용할 변수들

사용할 객체들

[메모]

1. 이야기 들려주기

<http://study.jr.naver.com/donghwa/view.nhn?donghwaNo=1477&categoryId=6&donghwaSort=regDate&ageType=ALL&viewAgeType=ALL>

무한도전 나비효과: <http://www.youtube.com/watch?v=i06vQiCijic>



[생각해 봅시다]

1. 에너지를 낭비해버리는 습관에는 어떤 것이 있나요?
2. 초록 지구를 만들기 위해서 내가 할 수 있는 일은 무엇인가요?

<http://terms.naver.com/entry.nhn?docId=958056&mobile&categoryId=3068>

2. 더 알아보아요.

<http://cafe.naver.com/hych/404>

- (1) 가정에서 에너지 소비가 큰 제품에는 어떤 것들이 있나요?



- (2) 에너지를 절약하는 게임을 만들기 위한 요소에는 어떤 것들이 있나요?

3. 프로그래밍 문제

- 오늘 만들 프로그램: 낭비되는 에너지 줄이기 게임 만들기

[게임 설명]

집에는 다양한 가전제품이 있다. 랜덤한 시간마다 가전제품이 켜진다. 아마 가족중에 누군가가 다 사용한 가전제품을 끄지 않고 나두는 나쁜 습관을 가지고 있는 듯 하다. 낭비되는 에너지를 줄이기 위해 가전제품을 잘 관리해야 한다.

[필수 조건]

1. 가전제품이 여러개 있고 주인공은 사용자의 마우스 클릭한 곳으로 이동하게 해라.
2. 가전제품은 꺼져 있다가도 일정한 시간이 지나면 다시 켜져 버린다.
3. 사람이 가전제품에 닿으면 켜져 있던 가전제품은 꺼진다.
4. 가전제품이 켜져 있을 때 1초마다 에너지소비가 1씩 늘어난다.
5. 정해진 시간안에 에너지소비가 200을 넘지 않으면 임무 수행으로 여긴다.

4. 설계 및 프로그래밍 하기

전체 프로그래밍 디자인하기 (그림으로 그려주세요)

스프라이트 #1 :	#2 :	#3 :	#4 :
#5 :	#6 :	#7 :	#8 :



상상 · 프로그래밍 · 공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아가입니다!

SCRATCH반 ()학년 이름 ()

[오늘의 주제3] 분리수거로 환경을 지켜보자.

공부할 문제

1. 대도시의 문제점들을 말해봅시다.
2. 평소 분리수거를 실천하는지를 생각해보자.
3. 분리수거를 하면 좋은 점 / 하지 않았을 때의 나쁜 점을 알아보자.
4. 분리수거의 좋은 점을 알아봅시다.

프로그래밍 핵심 내용

핵심 프로그래밍 원리

사용할 변수들

사용할 객체들

1. 이야기 듣기 <http://www.youtube.com/watch?v=LeYybM4IaeQ>



[생각해 봅시다]

1. 분리수거가 없던 시절의 쓰레기는 어떻게 처리되었을까요?
2. 분리수거한 쓰레기는 어디로 옮겨지고, 어떻게 처리되나요?
3. 분리수거는 꼭 필요한 것인가요? 왜 그렇게 생각하나요?

2. 더 알아보아요.

(1) 분리수거하여 재생된 제품에는 어떤 것들이 있을까요?

(2) 분리수거를 하면 좋은 점이 무엇이라고 생각하나요?

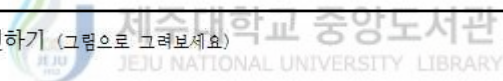
3. 프로그래밍 문제

- 오늘 만들 프로그램 제목:

[게임 설명]

4. 설계 및 프로그래밍 하기

전체 프로그래밍 디자인하기 (그림으로 그려주세요)





상상 · 프로그래밍 · 공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아갑니다!

SCRATCH반 ()학년 이름 ()

[오늘의 주제4] 미래의 로봇

공부할 문제

1. 미래의 로봇들은 어떤 기능을 하게 될까 생각해봅시다.
2. 자신이 상상하는 로봇의 모습과 기능을 말해봅시다.
3. 이러한 로봇을 프로그래밍하기 위해 사용될 센서는 어떤 것이 적당할까요?
4. 자신의 기발한 아이디어를 담은 프로젝트를 만들어봅시다.

프로그래밍 팁(Tip)

프로그램 시작할 때 프로젝트 설명하기를 하려면?	
다른 스프라이트를 동작시키기 위한 스크래치의 기능은?	
모양의 중심 설정은 어떤 기능인가요?	

1. 이야기 듣기 <http://www.youtube.com/watch?v=1f128IETFuw>

[생각해 봅시다]

1. 미래에 개발될 신기술에는 어떤 것들이 있을까요?
2. 신기술이 적용된 미래의 로봇은 어떤 모습일까요?

2. 더 알아보아요.

(1) 인공지능(AI)의 의미는 무엇일까요?

(2) 로봇은 인간보다 지능이 뛰어나다고 할 수 있나요? 당신의 생각은?

3. 프로그래밍 문제

- 오늘 만들 프로그램 제목:

[게임 설명]

4. 설계 및 프로그래밍 하기

전체 프로그래밍 디자인하기 (그림으로 그려주세요)





상상 · 프로그래밍 · 공유

무엇을 상상하든 그것을 만들어 낼 수 있어요!

창의력, 사고력, 논리력, 문제해결력 신장을 위한
스크래치 STEAM 교육이 여러분을 찾아가입니다!

SCRATCH반 ()학년 이름 ()

[오늘의 주제5] 센서 200% 활용하기

공부할 문제

1. 저항센서를 이용해서 만들 수 있는 프로젝트를 생각해봅시다.
2. 주변에 전류가 흐르는 물체와 조금 흐르는 물체를 분류하여 만들 수 있는 프로젝트를 생각해봅시다.
3. 과일이나 채소도 전류가 흐르는지 생각해 보고 이를 프로젝트에도 활용해 봅시다.

프로그래밍 핵심 내용

핵심 프로그래밍 원리

사용할 변수들

사용할 객체들



1. 프로그래밍의 세계 - Scratch의 끝은 과연 어디까지?

<http://www.youtube.com/watch?v=4APzVnQ2070&feature=related>

<http://www.youtube.com/watch?v=hF5DsX1XGBE>

[생각해 봅시다]

1. 프로그래밍을 경험하고 나서 컴퓨터과학에 대한 생각이 어떻게 바뀌었는지 말해봅시다.
2. 프로그래밍을 잘하려면 평소 어떤 공부를 잘 해야 할까요?
3. 어릴적부터 프로그래밍을 하면 좋은 점은 무엇이라고 생각하나요?

2. 프로그래밍

- 오늘 만들 프로그램 제목:

[게임 설명]

3. 설계 및 프로그래밍 하기

전체 프로그래밍 디자인하기 (그림으로 그려주세요)



<부록 7> 정규교육과정에서의 STEAM 교육 프로그램 교재



※ 공부할 문제

1. 스크래치의 화면과 메뉴를 알기
2. 간단한 블록을 연결하여 고양이 프로그래밍 하기
3. 직접 만든 프로그램에 이야기를 만들어 보기

1. 스크래치 화면 익히기



2. 스크래치 갖고 놀기

- ① 고양이 움직이게 하기
- ② 고양이가 뛰게 만들기
- ③ 고양이를 조종하기
- ④ 다른 스프라이트 직접 그려보기
- ⑤ 스크래치 파일 저장하기

3. 이야기 만들어서 발표하기 (디지털 스토리텔링)

4. 내가 만든 프로그램 올리기



스크래치 교실



※ 공부할 문제

1. 스크래치 무대의 x좌표와 y좌표를 이해하고 이를 이용하여 스프라이트 움직이기
2. 제어블록에서 무한반복을 어떠한 경우에 사용하는지 이해하기
3. 간단한 조건문(만약 ~ 이라면)을 사용하여 프로그램 만들기
4. ~에 닿기, ~색에 닿기를 이용하여 프로그램 만들기

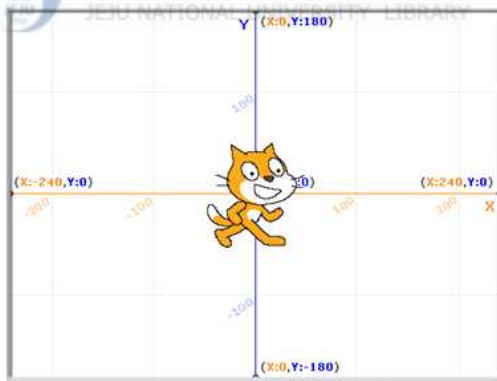
스크래치 무대의 x, y좌표 이해하기

[문제 만나기]

스크래치 무대의 x좌표와 y좌표를 이해하고 이를 이용하여 스프라이트 움직이기

[준비하기]

- ① 무대를 선택하고 배경의 새로운 배경 가져오기를 선택한다.
- ② backgrounds(배경) 폴더에 xy-grid를 선택하기 확인 버튼을 누른다.



[문제 해결방법 찾기]

스크래치의 모든 위치는 x와 y의 좌표로 나타낼 수 있습니다. 예를 들어 지금 고양이 가 서있는 위치는 x좌표 0, 좌표 0인 위치이고 (x:0, y:0)또는 (0, 0)으로 나타낼 수 있습니다. x좌표는 좌우(왼쪽, 오른쪽) y좌표는 상하(위, 아래)를 나타냅니다.

x좌표의 가장 작은 값 : (), 무대의 가장 ()쪽

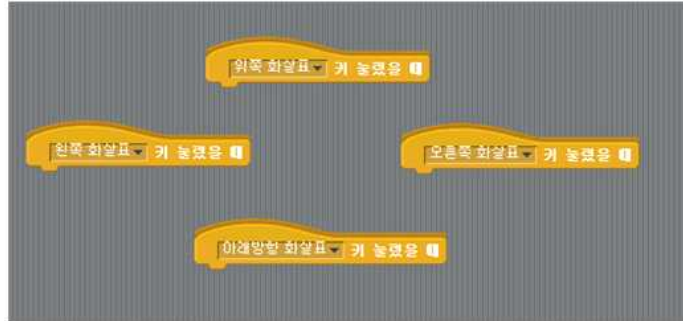
x좌표의 가장 큰 값 : (), 무대의 가장 ()쪽

y좌표의 가장 작은 값 : (), 무대의 가장 ()쪽

y좌표의 가장 큰 값 : (), 무대의 가장 ()쪽

[문제 해결하기]

x좌표와 y좌표를 이용하여 화살표를 누를 때 마다 고양이가 그 방향으로 10씩 움직이는 프로그램을 만들어 봅시다.



고양이를 따라다니는 게 프로그래밍 하기

[문제 만나기]

고양이를 따라다니는 게를 만들어서 게를 피하는 게임을 만들어 봅시다.

[준비하기]

- ① 무대 : nature 폴더의 beach-malibu
- ② 스프라이트 : 고양이, 게
- ③ 고양이와 게의 스프라이트에 이름을 고양이, 게로 짓기



[문제 해결방법 찾기]

고양이는 우리가 직접 조종할 수 있지만 게는 자동으로 고양이를 쫓아오도록 프로그래밍을 해야 합니다. 스프라이트의 각도를 조절해야 하는데 각도와 관련된 명령어 블록은 다음과 같습니다.

15 ↻ 도 돌기	① 오른쪽으로 돌기 ② 왼쪽으로 돌기 ③ 방향 지정하기(왼쪽, 오른쪽, 위, 아래) ④ 특정 스프라이트, 또는 마우스의 포인터 쪽 보기
15 ↺ 도 돌기	
90 ▾ 도 방향 보기	
▾ 쪽 보기	

게에게 고양이를 쫓아가도록 프로그래밍 하려면 어떤 블록을 사용해야 할까요?

[문제 해결하기]

고양이를 쫓아가는 게를 프로그래밍 해 봅시다.





고양이가 계에 물린다면?

[문제 만나기]


고양이가 계와 부딪힌 것을 알 수 있는 방법에는 어떠한 것이 있을까요?

[준비하기]


① 앞선 프로그램에 이어서 계속 진행

[문제 해결방법 찾기]

관찰 블록 모음에는 스프라이트의 닿기, 마우스의 위치, 키보드의 버튼 누름 등을 관찰하는 명령어들이 있습니다. 스프라이트가 어디에 닿았는지를 알아보는 블록은 아래와 같습니다.

	<ul style="list-style-type: none"> ① 특정 스프라이트, 마우스의 포인터, 벽에 닿는 것을 관찰 ② 특정한 색에 닿는 것을 관찰 ③ 스프라이트의 특정 색으로 된 부분이 특정한 색에 닿은 것을 관찰
---	---

제어블록에는 어떤 일이 발생했을 때만 실행하도록 해주는 조건블록이 있습니다.

	<ul style="list-style-type: none"> ① 만약 어떤 일이 발생했을 때 감싸고 있는 명령어 블록 실행 ② 만약 어떤 일이 발생했을 때 감싸고 있는 명령어 블록 실행하고 발생하지 않았을 때는 아니면이 감싸고 있는 명령어 블록 실행
--	--

[문제 해결하기]

고양이에게 계와 부딪히게 된다면 “아야”라고 2초동안 말하는 프로그램을 작성해 봅시다.



※ 더 공부해 봅시다.



멋진 스포츠카 게임을 만들어 봅시다.

자동차는 직접 그려서 만듭니다.

자동차는 늘 1만픽씩 움직입니다.

좌우 화살표를 움직여 방향을 조절할 수 있습니다.

자동차는 파란색으로 된 길만 달릴 수 있고 길이 아닌 곳을 만나면 처음으로 돌아가야 합니다.



※ 공부할 문제

1. 임의(랜덤)의 수를 이용한 미디어아트 프로그램 만들기



미디어아트 프로그램 만들기

주머니 속에 각각 1~10의 숫자가 적힌 똑같은 크기의 공이 10개 있습니다. 몇 번 공을 뽑을지 누구도 알 수 없습니다. 뭐가 뽑힐지 알 수 없는 것, 이것을 컴퓨터 프로그래밍에서 사용할 수는 없을까요? 미디어아트 프로그램을 통해 알아보시다.



[문제 만나기]

임의의 수를 만드는 기능을 활용하여 누구도 알 수 없는 색깔, 크기, 각도, 위치의 변화를 주어 화려한 미디어아트 프로그램을 만들어 봅시다.

[준비하기]

1. 새로운 스프라이트 파일 선택하기 > People > anjuli-1 ~ 5까지 갖고 오기
2. 새로운 스프라이트 그리기 기능을 이용하여 별과 원을 그려봅시다.

[문제 해결방법 찾기]

※ 난수란? 주사위를 던졌을 때 1~6 중에 어떤 수가 나올지 알 수 없습니다. 이렇게 일정한 규칙이 없이 발생할 수 있는 수를 난수라고 합니다.

블록	동작
1 부터 10 사이의 난수	1과 10 사이의 수 중에 임의의 수가 선택됩니다.

[문제 해결하기]

x좌표, y좌표, 방향, 크기, 색깔 등을 난수 블록을 사용하여 바꿔봅시다.

블록	동작
x좌표 1 부터 10 사이의 난수 만큼 바꾸기	스프라이트의 x의 좌표가 1~10 사이의 수 중 하나로 바뀝니다.





※ 공부할 문제

1. 변수의 기능을 이해하여 화살쏘기 게임 만들기

화살쏘기 게임 만들기



[문제 만나기]

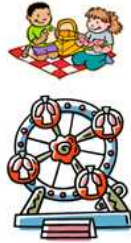
위와 아래로 움직이는 과녁이 있습니다. 파란색을 맞추면 10점, 빨간색을 맞추면 5점, 노란색을 맞추면 1점을 얻습니다. 10번의 기회 동안에 얻게 된 점수를 알기 위해서는 어떻게 프로그래밍을 해야 할지 알아보시다.

[준비하기]

1. 위의 그림에서 보는 대로 화살, 과녁, 고양이를 불러옵니다.
2. 적당한 배경의 무대를 불러옵니다. 예) tennis-backboard

[문제 해결방법 찾기]

※ 변수란? 값을 기억하기 위한 공간을 변수라고 합니다. 예를 들어 전화번호를 잊지 않기 위해서 메모지에 적어두곤 합니다. 컴퓨터 프로그램도 기억해두어야 할 값을 특별한 곳에 저장해두는데 이것을 변수라고 합니다.



놀이공원이라는 변수를 확인해서 열었는지 닫혔는지를 확인할 때

변수이름 : 놀이공원

놀이 공원이 닫혀있어 들어가지 못했습니다.

닫힘

다시 확인해봐도 놀이 공원이 닫혀있습니다.

닫힘

놀이공원이 열려 있어서 들어갈 수 있습니다.

열림

지갑에 들어있는 돈을 계산한 결과를 저장할 때

변수이름 : 지갑



지갑에 돈이 하나도 없습니다.

0

부모님이 1만원의 용돈을 주셨습니다.

10000

친구 생일 선물을 사느라 3천원을 꺼내 썼습니다.

7000

블록

제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

동작



변수를 만듭니다.
변수를 삭제합니다.
기회라는 변수입니다.
점수라는 변수입니다.
화살이라는 변수입니다.
기회 변수에 0을 저장합니다.(완전 새로운 값)
기회 변수에 저장되어 있는 값에 1을 더합니다.
(원래 저장된 값에 값 추가)

< 문제 > 고양이가 파란색 과녁을 쏘서 10점을 얻었습니다. 원래의 점수에 10점을 추가하려면 어떤 명령어 블록을 사용해야 할까요?

[문제 해결하기]

고양이와 화살, 과녁은 각각 아래와 같이 동작합니다. 화살 쏘기 게임을 완성해 봅시다.

고양이	화살	과녁
<ul style="list-style-type: none"> - 위쪽 아래쪽 화살표를 누르면 위 아래로 움직입니다. - 스페이스 키를 누르면 화살이 발사됩니다. 	<ul style="list-style-type: none"> - 발사되었을 때 10만큼 움직입니다. - 과녁에 닿거나 벽에 닿으면 다시 고양이에게 돌아옵니다. - 과녁에 닿으면 각각의 점수를 얻습니다. 	<ul style="list-style-type: none"> - 언제나 3만큼 위아래로 움직입니다. - 벽에 닿으면 튕깁니다. - 10번의 기회가 다되면 점수를 알려줍니다.

※ 힌트 : 변수는 총 3개가 있습니다.

1. 화살 변수 : 화살이 발사 상태 인지 정지 상태인지를 저장하는 변수(발사 또는 정지가 저장)

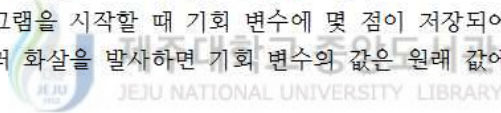
- 1) 깃발을 눌러 프로그램을 시작할 때 화살 변수에 무엇이 저장되어 있어야 할까요?
- 2) 스페이스 키를 누르면 화살 변수는 무엇으로 변해야 할까요?
- 3) 과녁이나 벽에 닿았을 때 화살 변수에는 무엇을 저장해야 할까요?

2. 점수 변수 : 명중한 과녁의 색깔에 따라 얻은 점수를 저장하는 변수

- 1) 깃발을 눌러 프로그램을 시작할 때 점수 변수에 몇 점이 저장되어 있어야 할까요?

3. 기회 점수 : 남아있는 화살 쏘는 기회를 저장하는 변수

- 1) 깃발을 눌러 프로그램을 시작할 때 기회 변수에 몇 점이 저장되어 있어야 할까요?
- 2) 스페이스 키를 눌러 화살을 발사하면 기회 변수의 값은 원래 값에서 어떻게 변할까요?





[주제 1] 우리 주변의 다양한 기체

[문제 탐색하기]

우리 주변에는 다양한 기체가 다양한 용도로 사용되고 있습니다. 다양한 기체의 쓰임을 알아보고 우리 주변에서 사용되고 있는 모습을 프로그램으로 만들어 봅시다.

[과학 원리 탐구하기]

1. 우리 주변에는 어떠한 기체가 있나요?
2. 우리는 산소를 어떠한 용도로 사용하고 있나요?



3. 우리는 질소를 어떠한 용도로 사용하고 있나요?
4. 그 외의 다양한 기체의 쓰임에 대해 조사해 봅시다.


[스크래치 프로그래밍 하기]

1. 등장 스프라이트



아이 점원 과자봉지 열린과장 눈물

2. 프로그램 동작 모습

	
<p>무대이름 : 제목 아이 등장</p>	<p>무대이름 : 상황1 아이 움직임</p>
	
<p>무대이름 : 상황2 점원 등장 과자 등장</p>	<p>무대이름 : 상황 3 점원 사라짐 눈물 등장</p>

3. 프로그래밍 설명

1) 무대 프로그래밍



- ① 깃발이 클릭되었을 때 : 배경을 제목으로 바꿈, 2초 뒤에는 상황 1 방송하기
- ② 상황 1이 방송되었을 때 : 배경을 상황 1로 바꾸기
- ③ 상황 2이 방송되었을 때 : 배경을 상황 2로 바꾸기
- ④ 상황 2이 방송되었을 때 : 배경을 상황 3로 바꾸기
- ⑤ 상황 4이 방송되었을 때 : 배경을 상황 4로 바꾸기

2) 아이 프로그래밍

아이는 첫 장면부터 끝까지 계속 등장하는 스프라이트입니다.



- ① 깃발이 클릭되었을 때 : 오른쪽 보기, 위치 정하기
- ② 상황 1이 방송되었을 때 : 위치 정하기, 왼쪽보기, 움직이기, 말하기, 상황 2 방송하기
- ③ 상황 2이 방송되었을 때 : 위치 정하기, 오른쪽보기, 움직이기, 시간 기다리기, 말하기
- ④ 상황 2이 방송되었을 때 : 위치 정하기, 말하기, 시간 기다리기, 상황 4 방송하기
- ⑤ 상황 4이 방송되었을 때 : 위치 정하기, 말하기

3) 과자붕지 프로그래밍

과자붕지는 깃발을 클릭했을 때는 숨어 있다가 점원이 과자주기를 방송할 때 등장합니다. 그리고 상황 3이 방송될 때 다시 숨겨야 합니다.



[나만의 프로그래밍하기]

1. 나만의 프로그래밍 주제 정하기

우리 주변에서 여러 가지 기체가 사용되는 모습을 프로그래밍으로 만들어 봅시다.

나의 주제 :

2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

--

3. 프로그래밍 설명서 만들기

프로그램 제목	
프로그래머 이름	
만든 이유	
내가 선택한 신재생 에너지	 제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY
등장하는 스프라이트	
프로그램 동작 방법	
여러분이 프로그래밍한 내용을 바탕으로 미래의 모습을 상상해 보고 적어봅시다.	

[정리하기]

1. 오늘 새로 알게 된 점을 적어봅시다.
2. 오늘 공부한 소감을 적어봅시다.





[주제 2] 지구가 뜨거워져요.

- 지구 온난화의 원인과 피해 생각해 보기

[문제 탐색하기]

지구 온난화라는 말을 들어본 적이 있나요? 우리가 이번 단원에서 배운 이산화탄소는 지구 온난화를 야기합니다. 지구 온난화는 왜 발생하고 그로 인해 어떤 문제가 생기는지 알아보시다. 그리고 지구 온난화를 막기 위해 우리가 할 수 있는 일에는 어떤 것이 있는지 알고 실천해 봅시다.

[과학 원리 탐구하기]

1. 지구 온난화는 무엇인가요?
2. 지구 온난화의 발생원인은 무엇인가요?
3. 나와 우리 가족의 행동 중에 지구 온난화를 발생시키는 행동에는 무엇이 있을까요?
4. 지구 온난화 때문에 누구에게 어떤 일이 발생하나요?(지구 온난화 때문에 생기는 피해)
5. 지구 온난화의 원인이 발생하는 장면과 피해 장면을 스크래치로 표현할 만한 이야기를 생각해 봅시다.(어떤 장면을 표현할지 간단하게 적어보세요.)

[스크래치 프로그래밍 하기 - 북극곰이 위험해]



1. 삭제 메시지가 방송되면 사라지는 빙하를 프로그래밍 하기
스프라이트 이름 : 빙하 1, 빙하 2



2. 삭제 메시지에 따라 빙하를 옮기며 피해를 입는 북극곰 프로그래밍 하기
스프라이트 이름 : 북극곰



3. 프로그램 아래에 고양이가 지구 온난화의 원인을 발생시키는 장면을 프로그래밍 해봅시다.
고양이가 그런 행동을 할 때마다 빙하삭제 메시지를 방송하게 하여 프로그래밍을 완성해 봅시다.


[나만의 프로그래밍하기]

1. 나만의 프로그래밍 주제 정하기
지구 온난화를 막아내라
나의 주제 :

2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

지구 온난화 피해 장면
지구 온난화 발생 장면

3. 프로그래밍 설명서 만들기

프로그램 제목	 <p>제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY</p>
프로그래머 이름	
만든 이유	
내가 선택한 지구 온난화의 원인.피해 장면	
등장하는 스프라이트	
프로그램 동작 방법	
여러분이 프로그래밍한 내용을 바탕으로 미래의 모습을 상상해 보고 적어봅시다.	

[정리하기]

1. 오늘 새로 알게 된 점을 적어봅시다.

2. 지구 온난화 예방을 위한 실천 다짐을 적어 봅시다.

3. 오늘 공부한 소감을 적어봅시다.





[주제 3] 이산화탄소량을 줄여볼까요?

- 신·재생에너지를 이용하는 미래의 모습 상상해보기

[문제 탐색하기]

우리가 사용하는 석탄과 석유, 천연가스는 그 양이 정해져 있어 언젠가는 사라진다고 합니다. 또한 석탄과 석유를 사용하면 지구온난화를 유발하는 이산화탄소가 생기면서 지구의 온도가 점점 올라갑니다. 그렇게 되면 극 지방의 빙하가 녹아서 바닷물이 높이가 올라가고, 폭우나 폭설 등의 이상기후가 나타나고 있습니다.

에너지 부족과 환경오염을 막을 수 있는 신·재생 에너지에 대해서 알아보시다.

※ 화석 연료(석유, 석탄, 천연가스)와 달리 환경 친화적이며 제한이 없이 사용이 가능한 자원을 뜻함. 비슷한 용어로는 대체 에너지가 있습니다.

[과학 원리 탐구하기]

1. 북극곰과 투발루 국민들은 어떤 어려움을 겪고 있나요?
2. 신·재생 에너지를 개발하는 이유는 무엇인가요?
3. 신·재생 에너지에는 어떤 것이 있을까요?(인터넷 검색해보기)
5. 태양광 발전기는 어떻게 동작하나요?(인터넷 검색해보기)
6. 풍력 발전기는 어떻게 동작하나요?(인터넷 검색해보기)

7. 미래 사회에는 어떠한 제품들이 신재생에너지로 만들어질지 상상해 봅시다.(인터넷 검색해보기)

[스크래치 프로그래밍 하기]



1. 바람을 만나면 돌아가는 풍력발전기 프로펠러 프로그래밍 하기
스프라이트 이름 : 프로펠러



2. 태양빛을 만나면 에너지를 얻는 태양광발전기 프로그래밍 하기
스프라이트 이름 : 태양광발전기



3. 무대를 이용하여 태양, 바람을 방송하기로 조절하기
무대



4. 태양에서 태양빛을 태양광 발전기로 이동시키기



[나만의 프로그래밍하기]


1. 나만의 프로그래밍 주제 정하기

이산화탄소를 줄이기 위해 신·재생 에너지를 이용하여 에너지를 미래의 자동차, 가전제품을 프로그래밍 해봅시다.

나의 주제 :

2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

3. 프로그래밍 설명서 만들기

프로그램 제목	
프로그래머 이름	
만든 이유	
내가 선택한 신재생 에너지	
등장하는 스프라이트	
프로그램 동작 방법	
<p>여러분이 프로그래밍한 내용을 바탕으로 미래의 모습을 상상해 보고 적어봅시다.</p>  <p>제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY</p>	

[정리하기]

1. 오늘 새로 알게 된 점을 적어봅시다.
2. 오늘 공부한 소감을 적어봅시다.

<부록 7> 방과후학교를 위한 STEAM 교육 프로그램 교재



스크래치 교실



- ※ 공부할 문제
1. 프로그래밍이 무엇인지 생각해 보기
 2. 스크래치 살펴보기

프로그래밍이란?

컴퓨터에게 여러분이 원하는 일을 시키기 위해서 컴퓨터가 이해할 수 있는 언어로 그 일을 설명 (표현)하는 것을 **프로그래밍**이라고 하고 이렇게 프로그래밍을 통해 만들어진 명령의 모음을 **프로그램**이라고 합니다.

아래와 로봇(컴퓨터)은 찾기, 방향으로 이동하기, 집기, 놓기라는 명령(사용이 가능한 명령)을 알고 있습니다. 이런 명령들을 이용해서 청소라는 프로그램을 프로그래밍하려면 어떻게 표현하면 될까요?



로봇에게 청소 설명하기

①		④	
②		⑤	
③		⑥	

① ~ ⑥ 번의 과정을 반복하면 로봇은 청소라는 프로그램을 수행하게 됩니다. 이러한 표현한 청소라는 프로그램은 로봇에게 사용 가능한 명령 5번으로 추가시킬 수도 있겠네요. 이렇게 로봇(컴퓨터)은 자기만의 언어와 명령을 갖고 있고 이것을 이용하여 새로운 프로그램을 만들고 로봇의 명령으로 추가할 수 있습니다.



스크래치 만나기

초등학생도 쉽게 배울 수 있는 프로그래밍 도구입니다.

레고 블록처럼 명령어 블록을 연결하면 나만의 프로그램을 만들 수 있습니다.



스크래치 속에서 여러분은 연극의 작가가 될 수 있습니다. 연극과 스크래치를 비교해 봅시다.

연극	스크래치
무대	
배우	
작가	
대본	
분장	

여러분이 프로그래밍 언어로 스크립트를 작성하면 여러분의 배우인 스프라이트가 여러분의 명령에 따라 움직이게 됩니다. 여러분이 “멋진 연극의 작가다”라고 생각하면 프로그래밍이 더 쉽게 느껴지겠네요.

여러분이 스프라이트에게 내릴 수 있는 명령은 비슷한 것 끼리 모아져 있습니다. 다음 블록 모음들은 어떤 명령어 블록들을 모아놓은 것인지 생각해 봅시다.

동작		제어	
형태		관찰	
소리		연산	
펜		변수	



나의 첫 프로그램 만들기

- 1) 고양이 앞으로 움직이게 하기, 회전하게 하기 (동작 - ○○ 만큼 움직이기, ○○도 돌기)
- 2) 고양이를 자동으로 움직이게 하기 (제어 - 무한반복 블록)
- 3) 고양이를 걷게 만들기(형태 - 다음모양 블록)
- 4) 화살표로 고양이 움직이게 명령하기 (제어 - ○○키 눌렀을 때)
- 5) 고양이에게 했던 명령을 다른 스프라이트 추가해서 명령내리기



새로운 스프라이트 추가하기

- 1) 새로운 스프라이트 버튼에서 스프라이트 불러오기를 클릭한다.



- 2) 원하는 스프라이트를 선택하여 확인버튼을 누르면 새로운 스프라이트를 불러온다.





스크래치 교실

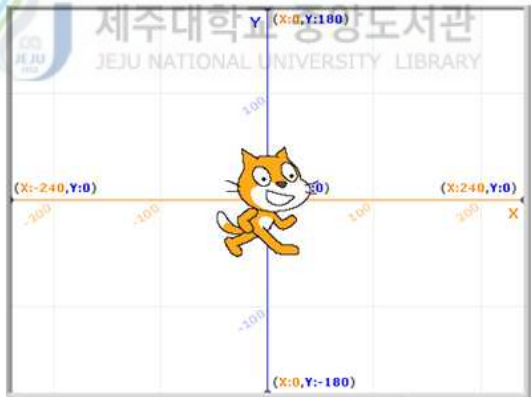
클릭되었을 때
무한 반복
처미있는 프로그래밍 공부 알리기

- ※ 공부할 문제
1. 스크래치 무대의 x좌표와 y좌표를 이해하고 이를 이용하여 스프라이트 움직이기
 2. 제어블록에서 무한반복을 어떠한 경우에 사용하는지 이해하기
 3. 간단한 조건문(만약 ~ 이라면)을 사용하여 프로그램 만들기
 4. ~에 닿기, ~색에 닿기를 이용하여 프로그램 만들기

스크래치 무대의 x, y좌표 이해하기

[문제 만나기]
스크래치 무대의 x좌표와 y좌표를 이해하고 이를 이용하여 스프라이트 움직이기

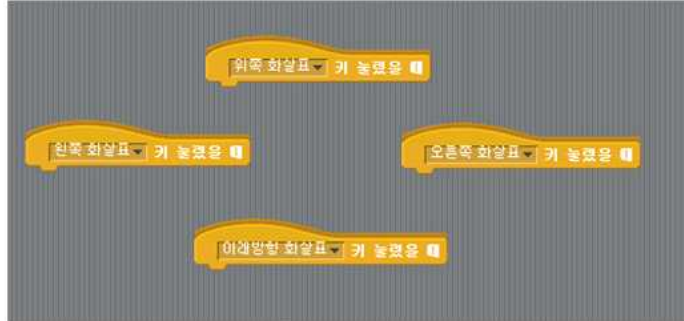
- [준비하기]
- ① 무대를 선택하고 배경의 새로운 배경 가져오기를 선택한다.
 - ② backgrounds(배경) 폴더에 xy-grid를 선택하기 확인 버튼을 누른다.



- [문제 해결방법 찾기]
- 스크래치의 모든 위치는 x와 y의 좌표로 나타낼 수 있습니다. 예를 들어 지금 고양이가 서있는 위치는 x좌표 0, 좌표 0인 위치이고 (x:0, y:0)또는 (0, 0)으로 나타낼 수 있습니다. x좌표는 좌우(왼쪽, 오른쪽) y좌표는 상하(위, 아래)를 나타냅니다.
- x좌표의 가장 작은 값 : (), 무대의 가장 ()쪽
 x좌표의 가장 큰 값 : (), 무대의 가장 ()쪽
 y좌표의 가장 작은 값 : (), 무대의 가장 ()쪽
 y좌표의 가장 큰 값 : (), 무대의 가장 ()쪽

[문제 해결하기]

x좌표와 y좌표를 이용하여 화살표를 누를 때 마다 고양이가 그 방향으로 10씩 움직이는 프로그램을 만들어 봅시다.



고양이를 따라다니는 게 프로그래밍 하기

[문제 만나기]

고양이를 따라다니는 게를 만들어서 게를 피하는 게임을 만들어 봅시다.

[준비하기]

- ① 무대 : nature 폴더의 beach-malibu
- ② 스프라이트 : 고양이, 게
- ③ 고양이와 게의 스프라이트에 이름을 고양이, 게로 짓기

[문제 해결방법 찾기]

고양이는 우리가 직접 조종할 수 있지만 게는 자동으로 고양이를 쫓아오도록 프로그래밍을 해야 합니다. 스프라이트의 각도를 조절해야 하는데 각도와 관련된 명령어 블록은 다음과 같습니다.

<div style="background-color: #4a7ebb; color: white; padding: 2px; margin-bottom: 2px;">15 ↻ 도 돌기</div> <div style="background-color: #4a7ebb; color: white; padding: 2px; margin-bottom: 2px;">15 ↶ 도 돌기</div> <div style="background-color: #4a7ebb; color: white; padding: 2px; margin-bottom: 2px;">90 ▾ 도 방향 보기</div> <div style="background-color: #4a7ebb; color: white; padding: 2px;">▾ 쪽 보기</div>	<ul style="list-style-type: none"> ① 오른쪽으로 돌기 ② 왼쪽으로 돌기 ③ 방향 지정하기(왼쪽, 오른쪽, 위, 아래) ④ 특정 스프라이트, 또는 마우스의 포인터 쪽 보기
---	--

게에게 고양이를 쫓아가도록 프로그래밍 하려면 어떤 블록을 사용해야 할까요?

[문제 해결하기]

고양이를 쫓아가는 게를 프로그래밍 해 봅시다.





고양이가 계에 물린다면?

[문제 만나기]

고양이가 계와 부딪힌 것을 알 수 있는 방법에는 어떠한 것이 있을까요?

[준비하기]

① 앞선 프로그램에 이어서 계속 진행

[문제 해결방법 찾기]

관찰 블록 모음에는 스프라이트의 닿기, 마우스의 위치, 키보드의 버튼 누름 등을 관찰하는 명령어들이 있습니다. 스프라이트가 어디에 닿았는지를 알아보는 블록은 아래와 같습니다.

	<ul style="list-style-type: none"> ① 특정 스프라이트, 마우스의 포인터, 벽에 닿는 것을 관찰 ② 특정한 색에 닿는 것을 관찰 ③ 스프라이트의 특정 색으로 된 부분이 특정한 색에 닿은 것을 관찰
--	---

제어블록에는 어떤 일이 발생했을 때만 실행하도록 해주는 조건블록이 있습니다.

	<ul style="list-style-type: none"> ① 만약 어떤 일이 발생했을 때 감싸고 있는 명령어 블록 실행 ② 만약 어떤 일이 발생했을 때 감싸고 있는 명령어 블록 실행하고 발생하지 않았을 때는 아니면이 감싸고 있는 명령어 블록 실행
--	--

[문제 해결하기]

고양이에게 계와 부딪히게 된다면 “아야”라고 2초동안 말하는 프로그램을 작성해 봅시다.



※ 더 공부해 봅시다.



멋진 스포츠카 게임을 만들어 봅시다.

자동차는 직접 그려서 만듭니다.

자동차는 늘 1만큼씩 움직입니다.

좌우 화살표를 움직여 방향을 조절할 수 있습니다.

자동차는 파란색으로 된 길만 달릴 수 있고 길이 아닌 곳을 만나면 처음으로 돌아가야 합니다.



< 주제: 열대어 색깔의 비밀 >

[문제 탐색하기]

'EBS 생물이 생생!(HD) 열대어 색깔의 비밀'을 보고 왜 열대어는 화려한 색깔을 띠는지 알아봅시다.

[과학 원리 탐구하기]

1. 등장하는 열대어의 특징을 조사해봅시다.
 - 1) 홍해 아네모네 피쉬(니모를 찾아라)
 - 2) 팬더 그루더
 - 3) 피카소 피쉬
 - 4) 배너 피쉬
2. 왜 열대어는 화려한 색깔을 띠나요?



3. 열대어는 왜 뭉쳐다닐까요?
4. 암컷보다 수컷이 더 화려한 이유는 무엇인가요?

[스크래치 프로그래밍 하기]

1. 랜덤하게 움직이는 열대어 만들기
2. 방송하기 또는 변수를 이용하여 상어등장 알람만들기
3. 상어가 등장하면 안전한 곳으로 대피하는 열대어 프로그래밍 만들기
4. 상어가 나가면 춤을 추는 열대어 만들기

[나만의 프로그래밍하기]

나만의 프로그램으로 수정해 봅시다.

[정리하기]

여러분이 만든 이야기를 발표해 봅시다.

< 주제: 사막여우는 어떻게 더운 사막에서 살까? >

[문제 탐색하기]

'EBS 생물이 생생!(HD) 사막여우는 어떻게 더운 사막에서 살까?'를 보고 사막여우가 사막에서 살 수 있는 이유를 알아봅시다.

[과학 원리 탐구하기]

1. 등장하는 열대어의 특징을 조사해봅시다.

- 1) 사막여우
- 2) 북극여우
- 3) 낙타
- 4) 미어캣

2. 사막의 낮과 밤은 어떻게 다른가요?

3. 사막이 아닌 곳의 낮과 밤은 어떤가요?

4. 사막여우가 사막에서 살 수 있는 이유 세가지

- 1) () :
- 2) () :
- 3) () :

5. 사막여우와 북극여우의 다른 점

6. 사막에 사는 다른 동물들이 사막에서 살 수 있는 이유



[스크래치 프로그래밍 하기]

1. 더운 지방에 사는 동물 스프라이트 만들기
2. 동물들의 모양과 색깔 바꾸기
3. 스프라이트끼리 대화하는 장면 만들기

[나만의 프로그래밍하기]

나만의 프로그램으로 수정해 봅시다.

[정리하기]

여러분이 만든 이야기를 발표해 봅시다.



스크래치로 만드는 과학 이야기

클릭되었을 때
무한 반복
재미있는 프로그램 만들기를 알려줘

< 주제: 물의 순환 >

[문제 탐색하기]



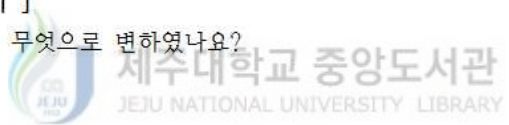
'신기한 스쿨버스 The Magic School Bus-How Water Changes'를 보고 물이 어떻게 변화하는 지 알아봅시다.

- <https://www.youtube.com/watch?v=oaCUyZw4Tjo>
- <https://www.youtube.com/watch?v=TdBG21XbxXk>
- <https://www.youtube.com/watch?v=AQKdkponoZM>

※ <https://www.youtube.com/watch?v=5IMymkFcn70> - 물의 순환송

[과학 원리 탐구하기]

1. 친구들이 가장 먼저 무엇으로 변하였나요?
2. 친구들이 하늘로 날아갈 수 있었던 이유는 무엇인가요?
3. 하늘로 가던 친구들이 구름으로 변한 이유는 무엇일까요?
4. 친구들은 어떻게 다시 땅으로 내려왔나요?
5. 물방울이 된 친구들은 어디로 가게 되나요?
6. 친구들이 탐험을 통해 알게 된 것은 무엇일까요?
7. 물의 순환 과정을 적어 봅시다.



[스크래치 프로그래밍 하기]

1. 등장하는 스프라이트 이름 짓기
2. 방송하기를 이용하여 프로그래밍 하기



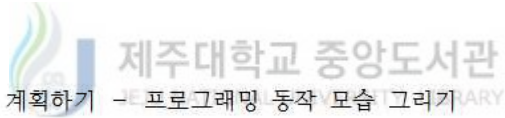
3. 무대 바꾸기

[나만의 프로그래밍하기]

1. 나만의 프로그래밍 주제 정하기

물을 아껴쓰자

나의 주제 :



2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

3. 프로그래밍 설명서 만들기

프로그램 제목	
프로그래머 이름	
만든 이유	
내가 선택한 지구 온난화의 원인.피해 장면	
등장하는 스프라이트	
프로그램 동작 방법	
여러분이 프로그래밍한 내용을 바탕으로 미래의 모습을 상상해 보고 적어봅시다.	

[정리하기]



1. 오늘 새로 알게 된 점을 적어봅시다.

2. 오늘 공부한 소감을 적어봅시다.



< 주제: 환경을 생각하는 에너지 >

[문제 탐색하기]

우리가 사용하는 석탄과 석유, 천연가스는 그 양이 정해져 있어 언젠가는 사라진다고 합니다. 또한 석탄과 석유를 사용하면 지구온난화를 유발하는 이산화탄소가 생기면서 지구의 온도가 점점 올라갑니다. 그렇게 되면 극 지방의 빙하가 녹아서 바닷물이 높이가 올라가고, 폭우나 폭설 등의 이상기후가 나타나고 있습니다.

에너지 부족과 환경오염을 막을 수 있는 신·재생 에너지에 대해서 알아보시다.



※ 화석 연료(석유, 석탄, 천연가스)와 달리 환경 친화적이며 제한이 없이 사용이 가능한 자원을 뜻함. 비슷한 용어로는 대체 에너지가 있습니다.

[과학 원리 탐구하기]

1. 북극곰과 투발루 국민들은 어떤 어려움을 겪고 있나요?
2. 신·재생 에너지를 개발하는 이유는 무엇인가요?
3. 신·재생 에너지에는 어떤 것이 있을까요?(인터넷 검색해보기)
5. 태양광 발전기는 어떻게 동작하나요?(인터넷 검색해보기)
6. 풍력 발전기는 어떻게 동작하나요?(인터넷 검색해보기)

7. 미래 사회에는 어떠한 제품들이 신재생에너지로 만들어질지 상상해 봅시다.(인터넷 검색해보기)

[스크래치 프로그래밍 하기]



1. 바람을 만나면 돌아가는 풍력발전기 프로펠러 프로그래밍 하기
스프라이트 이름 : 프로펠러



2. 태양빛을 만나면 에너지를 얻는 태양광발전기 프로그래밍 하기
스프라이트 이름 : 태양광발전기



3. 무대를 이용하여 태양, 바람을 방송하기로 조절하기
무대



4. 태양에서 태양빛을 태양광 발전기로 이동시키기



[나만의 프로그래밍하기]


1. 나만의 프로그래밍 주제 정하기

이산화탄소를 줄이기 위해 신·재생 에너지를 이용하여 에너지를 미래의 자동차, 가전제품을 프로그래밍 해봅시다.

나의 주제 :

2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

3. 프로그래밍 설명서 만들기

프로그램 제목	
프로그래머 이름	
만든 이유	
내가 선택한 신재생 에너지	
등장하는 스프라이트	
프로그램 동작 방법	
<p>여러분이 프로그래밍한 내용을 바탕으로 미래의 모습을 상상해 보고 적어봅시다.</p>	 <p>제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY</p>

[정리하기]

1. 오늘 새로 알게 된 점을 적어봅시다.
2. 오늘 공부한 소감을 적어봅시다.



스크래치로 만드는 과학 이야기



< 주제: 태양계를 탐험해 보자 >

[문제 탐색하기]

밤하늘에 반짝이는 별들을 본 적이 있나요? 둥근 보름달을 보며 소원을 빌어본 적이 있나요? 우주는 아름답고 신비한 공간인 동시에 앞으로 우리가 탐험하고 도전해야 할 대상입니다. 오늘은 우리 지구가 속한 태양계에 대하여 공부하면서 우주에 대한 도전을 시작해 봅시다.

※ 태양을 중심으로 행성, 위성, 소행성, 혜성들로 이루어진 집단을 '태양계'라고 합니다.

[과학 원리 탐구하기]

1. 태양계의 구성요소에는 무엇이 있는지 조사해 봅시다.

- 1) 태양 :
- 2) 행성 :
- 3) 위성 :
- 4) 소행성 :
- 5) 혜성 :

2. 태양계의 행성의 종류와 특징에 대하여 조사해 봅시다.

행성	특징(크기, 구성요소, 위치 등)
수성	
금성	
지구	
화성	
목성	
토성	

천왕성	
해왕성	

3. 태양계 행성을 크기가 큰 것부터 차례대로 나열해 보세요

※ 지구의 반지름이 1일 때 태양과 각 행성의 반지름

명칭	반지름	명칭	반지름	명칭	반지름
태양		지구	1	토성	
수성		화성		천왕성	
금성		목성		해왕성	

4. 태양에서 가까운 것부터 순서대로 행성을 나열해 보세요

※ 태양에서 지구까지의 거리를 1로 보았을 때 태양에서 각 행성까지의 거리

명칭	거리	명칭	거리	명칭	거리
수성		화성	1.0	천왕성	
금성		목성		해왕성	
지구		토성			

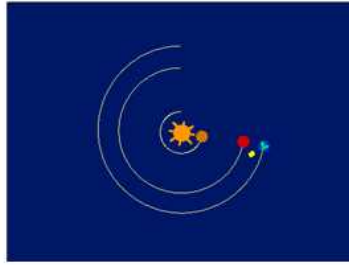
5. 태양계 행성은 어떻게 움직일까요?

- (1) 공전의 뜻 :
- (2) 태양계 행성은 어떻게 공전하고 있는가?
- (3) 지구가 태양을 한 바퀴 공전하는데 걸리는 시간은?

※ 행성이 태양 주위를 한 바퀴 공전하는데 걸리는 시간

명칭	시간	명칭	시간	명칭	시간
수성	0.24년	화성	1.88년	천왕성	84.02년
금성	0.62년	목성	11.86년	해왕성	164.77년
지구	1년	토성	29.46년		

[스크래치 프로그래밍 하기]



<p>1. 태양 주위를 공전하는 행성</p> <pre> 클릭되었을 때 x: 0, y: 30 쪽으로 가기 -90도 방향 보기 색 내리기 색의 색 [] 로 정하기 색의 크기 [1] 로 정하기 무한 반복 1도 돌기 0.5만큼 움직이기 </pre>	<p>2. 지구 주위를 공전하는 위성(달)</p> <pre> 클릭되었을 때 무한 반복 스프라이트2 위치로 가기 5도 돌기 1만큼 움직이기 </pre>
---	---

[나만의 프로그래밍하기]

1. 나만의 프로그래밍 주제 정하기
 오늘 공부한 내용을 바탕으로 태양 주위를 공전하는 행성의 모습을 프로그래밍 해봅시다.
2. 프로그래밍(이야기) 계획하기 - 프로그래밍 동작 모습 그리기

3. 프로그래밍 설명서 만들기

프로그램 제목	
프로그래머 이름	
만든 이유	
사용된 과학원리	
등장하는 스프라이트	
프로그램 동작 방법	
이야기 만들기	

[정리하기]

1. 오늘 새로 알게 된 점을 적어봅시다.

2. 오늘 공부한 소감을 적어봅시다.