방송통신정책연구 15-진흥-055

효과적인 초중고 SW온라인 교육체계 연구, 레파지토리와 평가를 중심으로

Effective Online Software education System for elementary and Secondary School

양병석/길현영/김윤명

2016. 01

연구기관 : 소프트웨어정책연구소

미래창조과학부
Ministry of Science, ICT and Future Planning

이 보고서는 2015년도 미래창조과학부 방송통신발전기금 방송통신 정책연구사업의 연구결과로서 보고서 내용은 연구자의 견해이며, 미래창조과학부의 공식입장과 다를 수 있습니다.

제 출 문

미래창조과학부 장관 귀하

본 보고서를 『**효과적인 초중고 SW온라인 교육체계 연구**』의 연구결과보고서로 제출합니다.

2015년 01월

연구기관 : 소프트웨어정책연구소
총괄책임자 : 양병석
참여연구원 : 길현영, 김윤명

목 차

제 1 장 한국 초중고 SW교육의 대안, 온라인 교육 ····· 1
제 1 절 초중고 SW 온라인 교육의 가치 ····· 1
1. 소프트웨어 중심사회의 도래 ····· 1
2. 소프트웨어 교육의 필요성 증가 ····· 2
3. 한국 초중고 SW교육의 대안으로서의 온라인 교육 ····· 7
제 2 절 SW 온라인 교육의 현황과 중점 연구 범위 ····· 12
1. 국내 SW 온라인 교육 현황 ····· 12
2. 온라인 SW교육의 가치체인 ····· 13
3. 온라인 SW교육 서비스의 제공 주요 서비스 ····· 15
4. 온라인 SW교육 콘텐츠 유형 ····· 21
5. 중점 연구 범위 ····· 23

제 2 장 SW 온라인 교육 서비스 사례 분석 ····· 25
제 1절. 소프트웨어 교육 콘텐츠 유형별 분석 ····· 25
1. 비쥬얼 프로그래밍 ····· 25
2. 텍스트 프로그래밍 ····· 41
3. 피지컬 컴퓨팅 ····· 58
제 2절. 사례 분석을 통해 본 시사점 ····· 63
1. 레파지토리 모델의 사례분석을 통해 본 시사점 ····· 63
2. 평가 모델의 사례분석을 통해 본 시사점 ····· 64

제 3 장 효과적인 초중고 SW 온라인교육 체계 ····· 66
제 1 절 초중고 SW온라인 교육을 위한 레파지토리 모델 ····· 66

1. 레파지토리 모델의 구성 요소 ····· 66
2. 레파지토리 모델의 프로세스 ····· 68
3. 효과적인 초중고 SW온라인 교육체계를 위한 레파지토리 모델 ····· 70
제 2 절 초중고 SW온라인 교육을 위한 평가 모델 ····· 71
1. 무엇을 평가 할 것인가? ····· 71
2. 무엇으로 평가를 할 것인가? ····· 74
3. 효과적인 초중고 SW온라인 교육체계를 위한 평가 모델 ····· 77

제 4 장 결론 및 시사점 ····· 78

참고문헌 ····· 79

부 록 ····· 81
1. OLC센터 ····· 81
2. Udacity ····· 82
3. CodeClub ····· 83

표 목 차

<표 1－1> 삼성전자의 비전공자 대상의 소프트웨어교육 ····· 1
<표 1－2> 삼성전자 개발(R&D)인력 현황 ····· 5
<표 1－3> 해외 주요 국가의 소프트웨어 교육정책 ····· 6
<표 1－4> 대한민국 소프트웨어 교육정책 ····· 8
<표 1－5> 국내 SW 온라인교육 현황 ····· 12
<표 1－6> 온라인 SW교육 가치체인의 단계 ····· 14
<표 1－7> 온라인 SW교육의 유형 ····· 21
<표 2－2> Hamster 사양 ····· 61
<표 2－3> 레파지토리에 저장되야 하는 요소들 ····· 63
<표 2－4> 큐레이션의 종류 ····· 63
<표 3－1> Computational Thinking 평가 Framework ····· 71
<표 3－2> 중학교 교육과정 정보 교과 내용 체계 중 일부 ····· 72
<표 3－3> 영국의 컴퓨팅 커리큘럼 ····· 72
<표 3－4> 미국의 컴퓨터 과학 커리큘럼 ····· 73
<표 3－5> 중등과정 성취기준 예시 ····· 74
<표 3－6> Comutational thinking 평가 Framework 요소별 산출물 ····· 75
<표 3－7> 평가주체별 평가방법 ····· 76

그 림 목 차

[그림 1－1] 인터넷에 연결되는 기기의 숫자 ····· 1
[그림 1－2] 소프트웨어 중심사회 ····· 2
[그림 1－3] 소프트웨어중심사회에서의 소프트웨어교육 ····· 4
[그림 1－4] 미국내 소프트웨어 인력 공급 예측 ····· 5
[그림 1－5] 글로벌 무크의 성장 ····· 9
[그림 1－6] 수준별로 다양하고 동적인 소프트웨어의 지식 ····· 10
[그림 1－7] 온라인 SW교육의 가치체인 ····· 13
[그림 1－8] 온,오프라인의 교육 통합검색을 지원하는 ThinkNew ····· 16
[그림 1－9] 지도검색을 지원하는 코드클럽 ····· 17
[그림 1－10] 소프트웨어중심사회 포털 ····· 18
[그림 1－11] 정해진 형태의 조작만 가능한 Junior SW ····· 20
[그림 1－12] 블록을 조합을 해봄으로서 문제해결을 시도 해볼수 있는 code.org ····· 20
[그림 1－13] 지식쇼핑 마인드스톰의 가격비교 결과 ····· 23
[그림 2－1] Code.org 온라인 사이트(http://code.org) ····· 26
[그림 2－2] code.org의 레파지토리 code studio ····· 26
[그림 2－3] code studio 의 최근 프로젝트 ····· 27
[그림 2－4] code studio 의 최근 스튜디오 ····· 27
[그림 2－5] code studio 의 큐레이트된 프로젝트 ····· 28
[그림 2－6] code studio 의 사람들이 리믹스 중인 프로젝트 ····· 28
[그림 2－7] code studio 의 사람들이 좋아하는 프로젝트 ····· 28
[그림 2－8] 20시간 소개 과정 ····· 29
[그림 2－9] 20시간 소개 과정 중 미로 퍼즐에 대한 설명 동영상 ····· 30
[그림 2－10] 미로퍼즐 미션 ····· 30
[그림 2－11] Scratch의 홈 화면 ····· 31
[그림 2－12] Scratch의 에디터 ····· 32

[그림 2-13] Scratch 탐험하기 ····· 32
[그림 2-14] Scratch 콘텐츠 분류 ····· 33
[그림 2-15] Scratch Wiki ····· 34
[그림 2-16] Scratch ED ····· 34
[그림 2-17] Scratch 프로젝트 프로젝트 페이지 ····· 35
[그림 2-18] Kodu Game Lab community 메인 페이지 ····· 36
[그림 2-19] Kodu gamelab의 Worlds ····· 37
[그림 2-20] 엔트리 홈 ····· 38
[그림 2-21] 엔트리의 프로젝트 제작 화면 ····· 38
[그림 2-22] 엔트리의 개인 갤러리 ····· 39
[그림 2-23] 엔트리의 학습하기 ····· 39
[그림 2-24] 코드아카데미의 홈 ····· 41
[그림 2-25] 코드아카데미 파이썬 학습화면 ····· 41
[그림 2-26] 코드아카데미 심화 학습 자료 ····· 42
[그림 2-27] 코드아카데미 교사자료 ····· 42
[그림 2-28] 코드아카데미 웹의 기초 과정 ····· 43
[그림 2-29] 코딜리티 메인화면 ····· 44
[그림 2-30] 코딜리티 프로그래밍 교육화면 ····· 45
[그림 2-31] 코딜리티 테스트 생성화면 ····· 45
[그림 2-32] 코딜리티 테스트 조건 설정 ····· 46
[그림 2-33] 코딜리티 채용 후보자 초대 ····· 46
[그림 2-34] 코딜리티 Lesson ····· 47
[그림 2-35] 코딜리티 Workspace ····· 47
[그림 2-36] 코딜리티 복잡도 PDF ····· 48
[그림 2-37] 코딜리티 Challenge Calcum 2015 ····· 48
[그림 2-38] 코다임 클라우드 코딩 서비스 구름(goorm) ····· 49
[그림 2-39] 구름(goorm)의 교육 자료 공유 서비스 ····· 50
[그림 2-40] 구름(goorm)의 자동채점 ····· 51

[그림 2－41] 구름(goorm)의 소스코드 유사도 검사 ····· 51
[그림 2－42] 구름(goorm)의 실시간 랭킹 ····· 52
[그림 2－43] 구름(goorm)의 답안 분석 기능 ····· 52
[그림 2－44] 프로젝트 오일러 ····· 53
[그림 2－45] Alospot ····· 54
[그림 2－46] DoveIt ····· 54
[그림 2－47] 깃허브의 소스 코드 ····· 55
[그림 2－48] 1~10 사이의 2의 배수 합 실행 결과 ····· 56
[그림 2－49] 1~10 사이의 2의 배수 합을 구현한 프로그래밍 소스 코드 ····· 56
[그림 2－50] 최적화된 1~10사이의 2의 배수 합을 구현한 프로그램 소스 코드 ····· 57
[그림 2－51] ODIY의 목적 ····· 58
[그림 2－52] ODIY의 자료현황 ····· 58
[그림 2－53] 주니어 SW ····· 59
[그림 2－54] 네이버 소프트웨어야 놀자 중 피지컬컴퓨팅 ····· 60
[그림 2－55] Hamster와 USB to BLE Bridge 동글 ····· 60
[그림 2－50] Hamster 주행 제어 ····· 61
[그림 3－1] 레파지토리 모델의 구성요소 ····· 66
[그림 3－2] 레파지토리 모델의 배포과정 ····· 68
[그림 4－1] OLC센터 온라인 강좌의 예 ····· 81
[그림 4－2] Udacity ····· 82
[그림 4－3] Udacity의 Nanodegree 프로그램 ····· 83
[그림 4－3] 코드클럽 첫 페이지 ····· 84
[그림 4－4] 코드클럽 코스 ····· 84
[그림 4－5] 코드클럽 교사 훈련 서비스 ····· 85

요 약 문

1. 효과적인 초중고 SW온라인 교육체계 연구, 레파지토리와 평가를 중심으로

2. 연구 목적 및 필요성

소프트웨어가 삶의 질을 향상시키고, 기업과 정부의 경쟁력에 지속적으로 영향을 끼치는 소프트웨어 중심사회가 도래하여 소프트웨어가 중요하다는 인식이 커지고 있다.

모든 영역에서 소프트웨어 활용이 증가됨에 따라 전공자가 아닌 경우에도 누구나 소프트웨어 지식이 필요하며, 전공자의 경우에는 더 높은 수준의 소프트웨어 역량이 필요하다.

이에 따라 세계적으로 소프트웨어 교육 열풍이 불고 있고, 대한민국에서도 초중고 소프트웨어 교육이 필수화되었다.

그러나 소프트웨어 교육을 위해 당장 투입할 자원이 부족한 상황에서, 지금까지 여러 교육 현장을 바꾸어 놓은 온라인 교육 방법들을 적용한 온라인 소프트웨어 교육이 하나의 대안이 될 수 있으며, 그 효과에 대한 연구가 필요하다.

3. 연구의 구성 및 범위

온라인 소프트웨어 교육은 다양한 콘텐츠를 가지고 있으며, 서비스가 소비자에게 전달되고 활용되기까지의 가치체인을 따라 여러 주요 기능이 존재한다.

본 연구는 그 중에서 비주얼프로그래밍, 텍스트프로그래밍, 피지컬컴퓨팅을 포함한 다양한 영역을 조사하고, 초중고 온라인 소프트웨어 교육에 시급하게 필요한 레파지토리와 평가에 관해 중점적으로 연구하였다.

4. 연구 내용 및 결과

국내외 사례 분석과 소프트웨어교육의 목적에 따른 레파지토리와 평가 모델을 분석하여 효과적인 초중고 온라인 소프트웨어 교육체계를 제안하였다.

먼저 레파지토리 모델은 사용자 그룹별로 권한이 차등되어야 하며, 콘텐츠 유형 중심이 아닌 프로세스 중심, 최종 산출물 저장공간이 아닌 형상관리도구, 그리고 커뮤니티 기반의 확산 모델이 필요하다,

평가의 모델은 코드 평가의 경우, 비교적 간단한데 수행 결과를 즉각적으로 제공해야 하고 단계적으로 난이도 있는 문제에 도전할 수 있도록 해야 한다. 문제해결능력을 평가하기 위해서는 프로젝트 기반 협업을 포함한 각종 활동들을 추적 조사해야 한다.

5. 정책적 활용 내용

이러한 레파지토리 모델과 평가 모델은 한국의 초중고 소프트웨어 교육에 필요한 온라인 교육자원 설계나 소프트웨어 교육시장 진입을 목표로 하는 기업들의 기초자료로 활용이 가능하다.

6. 기대효과

소프트웨어 교육이 필수화됨에 따라 기대와 우려가 모두 높아지고 있다. 특히 국내의 경우, 제대로 된 소프트웨어 교육이 실시될 수 있는지에 대한 부정적인 의견도 많다. 본 연구가 양질의 소프트웨어 교육을 초중고 학생들에게 효과적으로 제공하는 온라인 소프트웨어 교육 체계 구성에 실질적인 도움이 될 것으로 기대한다.

SUMMARY

1. Effective Online Software education System for elementary and Secondary School

2. Objective and Importance of Research

With the advent of software centric society where the software improves the quality of human life and continues to affect the competitiveness of enterprises and governments, the recognition that software is important is growing.

As practical use of software is increased in all areas, everyone needs knowledge of software even if he or she is not software major. Therefore, software professionals are requested for the higher level of software capabilities.

Accordingly, software education is a worldwide hot issue. In Korea, software training class became mandatory in elementary and secondary schools.

As an alternative of the situation that we can only invest limited resources for the software education, applying online education method, which has changed many scenes at the educational field, may be the solution to the current situation. Thus, the effectiveness of deploying online education method needs to be studied.

3. Contents and Scope of the Research

On-line software education has a variety of contents, and there are several key features along the value chain until the service is delivered to the consumers and be utilized by them.

This study is carried out for various areas including visual programming, textual programming and physical computing, and is focused on the repository and the

evaluation model that are urgently needed in on-line software education framework for schools.

4. Research Results

We carried out domestic and international case studies and analyzed the repository and the evaluation model for the purpose of software education.

Based on the analyses, we propose an effective on-line software education system for elementary and secondary school.

First, the repository model should be differently authorized for each user group. It should be process-oriented rather than content-type-oriented. It must not be just storage for final product. It must contain software configuration management system. The community-based diffusion model is also needed.

In the evaluation model, code evaluation is relatively simple. It is needed to provide immediate execution, and should organize courses stepwise according to the difficulty level of the problems. In order to evaluate the problem-solving skills, facilities of tracking various activities including project-based collaboration is needed.

5. Policy Suggestions for Practical Use

The repository and evaluation model of on-line software education can provide criteria for measuring on-line educational resources for software education for schools in Korea. It also can be used as basic information for companies that aim to enter the software education market.

6. Expectations

On software education class became mandatory course, all the expectations and concerns are increasing. In particular, there are many cases of negative feedbacks about whether the proper education of software can be implemented in Korea. This study is

expected to be a substantive help to configure the online software education system to effectively provide superior quality education to elementary and secondary students.

CONTENTS

Chapter 1. Effective Online Software education System for elementary and Secondary School

Chapter 2. Objective and Importance of Research

Chapter 3. Contents and Scope of the Research

Chapter 4. Research Results

Chapter 5. Policy Suggestions for Practical Use

Chapter 6. Expectations

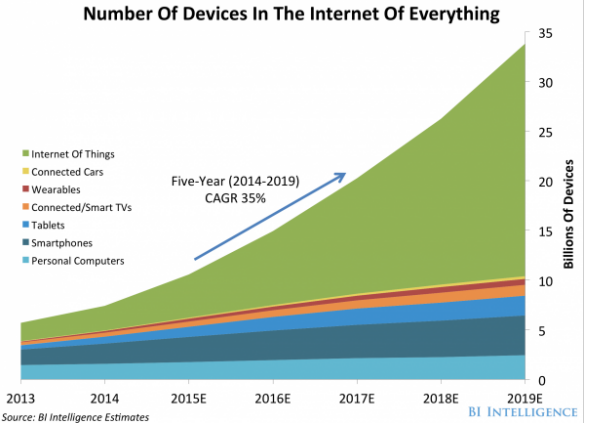
제 1 장 한국 초중고 SW교육의 대안, 온라인 교육

제 1 절 초중고 SW 온라인 교육의 가치

1. 소프트웨어 중심사회의 도래

소프트웨어가 개인·기업·정부의 전반에 광범위하게 사용되어 삶의 질을 향상 시키고 기업과 정부의 경쟁력이 지속적으로 제고되는 사회[1]인 소프트웨어 중심사회가 도래했다. 2020년이면 전세계에 50억 인구가 500억개의 사물이 연결될 것이라는 전망[2]이 나왔다. 이 연결된 모든 사물은 소프트웨어가 들어갈 것이고, 소프트웨어에 의해 통제되고 행동하게 된다. 과거에는 우리는 소프트웨어가 기존 프로세스의 효율성을 높이는 도구로서, 혹은 특정 소프트웨어 산업영역에서 독자적인 것으로 생각했다. 그러나, 이렇게 소프트웨어가 모든 기기와 모든 사회에 혈관처럼 파고들기 시작했고, 산업 자체를 바꾸는 혁신을 주도하기 시작했다.

[그림 1-1] 인터넷에 연결되는 기기의 숫자



자료: BI Intelligence Estimates

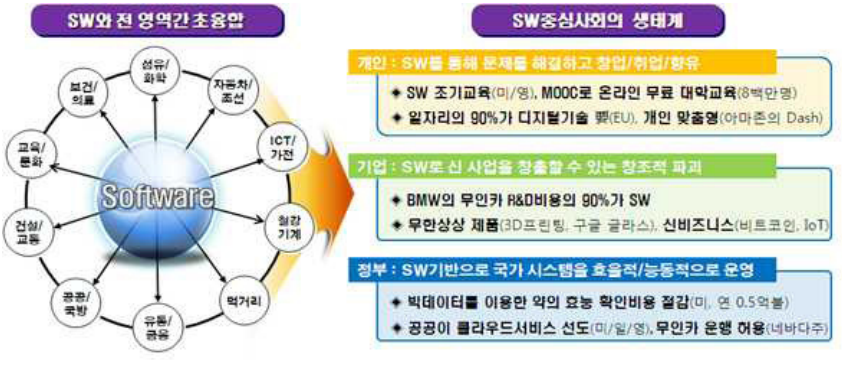
1) [2014-003] SW중심사회 – 의미와 대응 방향, 소프트웨어 정책연구소
2) World Economic Forum, By John Chambers, Jan 15 2014,

소프트웨어는 다양한 분야에서 핵심 경쟁력으로 떠오르거나, 기존 영역을 없애거나 통폐합 하고, 혹은 새로 만들어내는 파괴적인 혁신을 수행하기 시작했다.

> 2014년 한국, IT활용지수[3]는 55.7점으로 전년대비 2.7점 상승하여 "정보시스템을 통해 기업간 정보 공유 및 협업을 하는 단계"(기엄 간 협업 단계)를 기록하였으며, 전략적 경영 및 신사업 창출인 4단계(58.6점 초과)에 이르면 SW에 대한 수요는 폭발할 것으로 예상된다.

이로서 소프트웨어가 사회의 경쟁력까지 좌우하게 되었고, 개인의 소프트웨어적 역량도 국가적으로도 매우 중요한 사안으로 인식 되고 있다.

[그림 1-2] 소프트웨어 중심사회



자료 : 미래창조과학부, 소프트웨어 중심사회 실현 전략, 2014

2. 소프트웨어 교육의 필요성 증가

□ 기존 정보화 교육의 한계

이러한 SW중심사회에서는 모든 영역에서 SW활용과 응용이 증가됨에 따라, SW관련 전

3) IT활용지수 : 통계청 승인 국가 통계, 국내 기업의 IT활용 분야 및 수준, IT조직과 인력, IT인프라 구축 등을 파악한 지표

공자가 아닌 경우에도 SW를 일상적으로 만나고 이해하는 것을 요구 받는다. 물론, 소프트웨어 중심사회 이전에도 소프트웨어 교육은 정보화교육이라는 이름으로 존재하였다. 이는 기존의 업무를 디지털로 옮기고, 디지털 기기를 통해 더 빠르게 처리함을 목표로 하였으며 오프라인의 문서나 사진을 디지털로 바꾸어 수정하는 소프트웨어의 활용, 곧 한글[4]이나 포토샵[5] 같은 능력을 요구했었다. 인터넷이 등장한 후에도 크게 달라지지는 않았다. 인터넷은 웹문서를 편집하기 위한 웹에디터 같은 전문 소프트웨어 교육을 위주로 진행 되었다.

그러나, 사회는 문서는 인쇄 없이 디지털기기에서 유통되거나 직접 사용되었고, 사진은 소리와 동영상 그리고 양방향성을 가진 멀티미디어로 발전했다. 다양한 소프트웨어가 출현하고 그래픽인터페이스의 사용자 경험의 일관성이 증가하여 하나의 전문 소프트웨어 지식이 아닌 다양한 소프트웨어를 언제든 사용할 수 있는 소프트웨어 자체의 이해를 요구하기 시작했다.

또한 모든 산업과 사물 및 기기에서 디지털 데이터나 자료의 중요성이 증가하여 이러한 숫자를 단순히 더하고, 정리하던 엑셀의 수준이 아닌 자료의 연관도를 분석하고 판단하기 위한 데이터의 처리 능력이 요구되고 있다. 이러한 변화는 기존의 소프트웨어 활용 중심의 소프트웨어교육이 한계를 드러내기 시작했으며, 소프트웨어 역량을 가진이들과의 격차를 만들기 시작했다.

□ 대중을 위한 디지털 리터러시, 소프트웨어 교육

결국은 소프트웨어에 대한 이해는 기본적인 읽기, 쓰기처럼 디지털리터로시[6]의 일환으로 이야기되기 시작했다. 소프트웨어가 동작하는 기본적인 원리와 소프트웨어가 만들어지는 철학 등에 대한 교육은 새로운 소프트웨어를 만났을 때 별도의 교육에 들어가는 시간과 비용을 크게 줄여줄 수 있다.

특정 기술 서적을 읽는 방법을 공부하는 것이 기존의 정보화 시대의 교육이었다면, 소프

---

4) 現한컴(한글과 컴퓨터사)에서 만든 문서 편집 소프트웨어

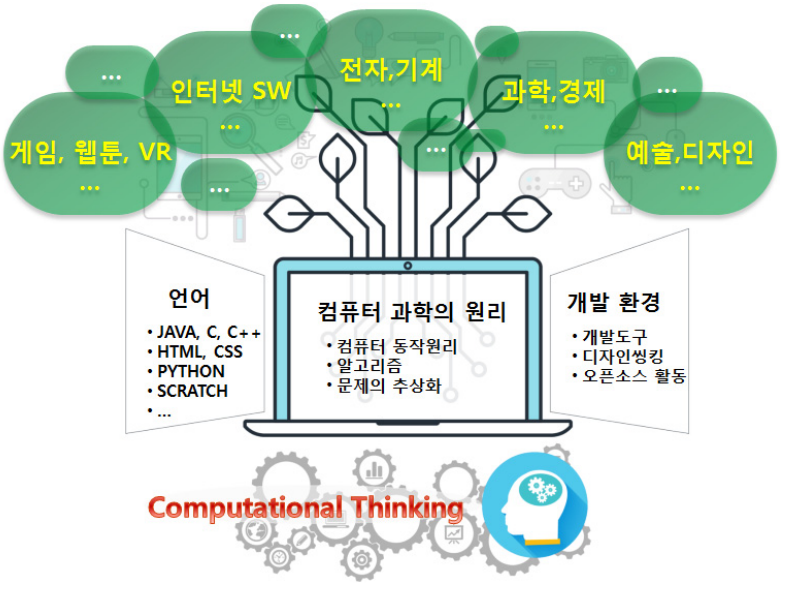
5) Adobe사에서 만든 사진 및 그래픽 편집 소프트웨어

6) 디지털리터러시 : 읽기, 쓰기처럼 스마트폰, 태블릿, 컴퓨터등의 디지털 기기들을 숙련되게 사용하는 기본 소양

트웨어 교육은 글을 배우는 것이다. 우리는 국문학자나 작가가 아니지만, 글을 읽는 것이 중요하다는 것을 이미 알고 있다. 작가는 아니지만 우리는 필요에 따라서 다양한 분야에서 글을 쓰고 있다.

글이 등장하면서 모든 분야에서는 혁신이 일었다. 모든 분야의 일하는 방법이 바뀌었고, 글을 통해 어떻게 해당 분야를 개선하고 혁신할지를 고민했다. 이제 소프트웨어도 마찬가지다. 소프트웨어를 일상적으로 사용하면서 소프트웨어를 통해 모든 분야를 혁신시킬 수 있다. 이것이 모든 이들에게 소프트웨어 교육이 필요한 이유라고 할 수 있으며, 이에 따른 소프트웨어 교육에 대한 필요성이 증가하고 있으며 비전공자를 대상으로 한 소프트웨어교육 사례 또한 증가하고 있다.

[그림 1-3] 소프트웨어중심사회에서의 소프트웨어교육



자료 : 직접 작성, 소프트웨어정책연구소

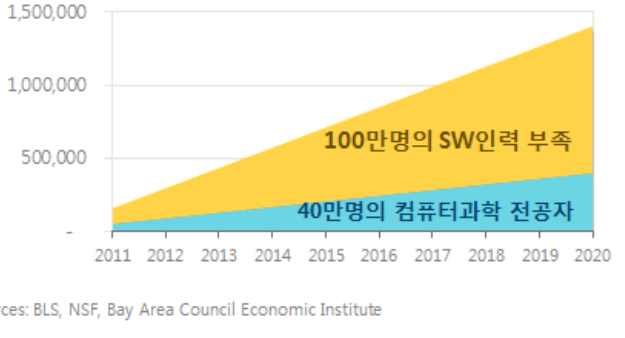
〈표 1-1〉 삼성전자의 비전공자 대상의 소프트웨어교육

그 룹	SCSA (Samsung Convergence Software Academy)	SCSC (Samsung Convergence Software Course)
설명	공채에서 인문계생을 선발하여, SW교육 후 입사	비전공자를 SW과목을 수강시키고, 부전공 학위 부여하는 교육과정
첫 시행	2013년 3월	2014년 3월
모집인원	년 400명	5년간 5000명 (학발 학기당 80명 이내)
혜택	삼성전자 삼성 SDS정규직 입사	삼성전자 공채지원 우대
SCSC 개설 대학	서울대, 연세대, 고려대, 성균관대, 서강대, 한양대, 중앙대, 숭실대, 이화여대 등 25개 대학	

☐ **양질의 SW전문인력 부족으로 인한 효과적인 SW전문인력 양성 필요성 증가**

소프트웨어의 전문인력이 부족한 것은 이미 현실이다. 가트너는 전세계 소프트웨어 산업의 규모가 2000년에 1200조원에서 2015년 1700조원으로 성장할 것으로 전망하였으며, 2020년 미국에만 140만명의 소프트웨어 인력이 필요하나, 100만명의 인력이 부족할 것으로 예상되고 있다.

[그림 1-4] 미국내 소프트웨어 인력 공급 예측



자료 : Code.org

국내에서도 전통적 소프트웨어 산업 뿐만 아니라, 삼성전자 같은 제조업에서도 이미 소프트웨어 인력 비용은 지속적으로 높아지고 있었다. 삼성전자의 경우 지속적으로 소프트웨어 인력을 충원해 2013년에 약 7만명의 R&D인력 약 60%인 4만명이 소프트웨어 인력이고, 이중 절반 정도가 국내 소프트웨어 인력으로 2만명의 소프트웨어 인력이 종사하는 것으로 나타났다.

또한 스마트카와 커넥티드카가 주목 받기 시작한 2016년은 자동차산업에서의 소프트웨어 기술 확보와 인력의 공격적 충원이 진행 중[7]이다.

〈표 1-2〉 삼성전자 개발(R&D)인력 현황

년도	개발 (R&D)	하드웨어	소프트 웨어	국내	해외	소프트 웨어 인력비율	국내 소프트 웨어 인력비율
2011년	55,320명	27,431명	27,889명	18,493명	9,396명	50.4%	66.3%
2012년	60,495명	26,866명	33,629명	18,754명	14,875명	55.6%	55.8%
2013년	69,230명	28,724명	40,506명	20,097명	20,409명	58.5%	49.6%
2014년	70,398명	29,871명	40,527명	21,015명	19,512명	57.6%	51.9%

자료 : 지속가능경영보고서 2014, 2015 삼성전자

한국직업능력개발원에 따르면[8] 2014~2018년 기간중 IT전문인력의 신규수요는 누계 기준으로 19만 3천 5백명에 이를 전망이며 신규채용 중 소프트웨어가 73.1%로 가장 높았으며 그 중 정보통신전문학사 인력이 53.8%로 다른 기술 분야대비 높은 비중을 차지했으며, HW부분은 초과공급을 예상한데 반해 SW부분은 거의 모든 부분에서 초과수요를 예상했다.

이에 발 맞추어 정부에서는 국내의 2019년까지 소프트웨어 중심대학 20개를 선정하여

7) 소프트웨어, 자동차 '소용돌이' 중요성 커지고 인력도 이동 (2015-12-31) 전자신문

8) 2014 ICT 전문·융합(SW)인력 실태분석 및 전망 (2015

최장 6년간 연평균 20억원을 지원하여 소프트웨어전문인력을 5천 500여명을 양성할 계획을 발표 했으며, 2015년 10월에는 가천대·고려대·경북대·서강대·성균관대·세종대·아주대·충남대의 8개 대학을 선정해서 사업을 진행중[9]에 있다.

3. 한국 초중고 SW교육의 대안으로서의 온라인 교육

□ 국내외 소프트웨어 교육 정책

SW교육은 모든 국민을 대상으로 한 소양교육과 전문 SW인력양성을 위한 교육으로 양과 질면에서 모두 높은 필요성을 요구받고 있다. 이에 대한 일환으로 세계 각국은 앞다투어 소프트웨어 교육을 의무화 혹은 강화를 진행하고 있는 중이다.

〈표 1-3〉 해외 주요 국가의 소프트웨어 교육정책

국 가	주요 내용
미국	·'K-12 컴퓨터 과학 표준 (K-12 Computer Science Standards)발표 ('11년) ·아칸소주는고등학교 정규과목에 coding 수업 포함 의무 법제화 ·워싱턴, 텍사스, 켄터키 등은 고등학교 제2외국어 대신 coding 선택
영국	·초·중등학교 정규 교과과정에 SW 필수 교육 ·만 5세부터 알고리즘과 디지털 콘텐츠 제작 및 활용 기술 습득 ·1,500여개 학교에서 방과 후 코딩 프로그램 'Code Club' 운영 · ICT(활용중심)에서컴퓨팅(코딩중심)과목으로개편 ('13년)
이스라엘	·'94년부터 소프트웨어 과목을 정규과목에 포함 ·고등학교 이과생들을 대상으로 3년간 450시간을 SW수업에 할애 ·대입시험에서 90%가 '컴퓨터과학' 선택 ·컴퓨터과학 과목에 프로그래밍 내용을 45% 이상 구성
중국	·중·고등학교 정규 교과과정에 SW 필수 교육 (초등학교는 선택) ·전국 37개 NPSS(National Pillot School of Software) 운영, 약 6.4만명재학중 ·정보기술 과목 전체 시간 수의 70% 이상이 실습
기타	·(일본) '정보' 과목을 고등학교에서 필수과목으로 지정('09년) ·(인도) '프로그래밍'과목을 초중등에서 필수과목으로 지정('13년), ·(에스토니아) 2015년부터 모든 초등학생이 SW교육을 받게 됨 ·(핀란드) 2016년 9월부터 모든 초등학생이 SW교육을 받게 됨

9) SW중심대학 추진 계획 (2015-7) 미래창조과학부

국내에서도 2014년 7월 소프트웨어 중심사회 실현전략보고회에서 소프트웨어교육 의무화를 통해 논리적 사고·문제분석 능력을 증진하는 미래형 창의인재 양성 계획을 발표를 했고, 지속적으로 소프트웨어 교육에 대한 정책을 추진해왔다.

〈표 1-4〉 대한민국 소프트웨어 교육정책

구분	주요 내용		
초등학교	·실과 ICT 단원을 SW기초 소양단원으로 확대 구성 · 17시간 이상 필수 교육	·(미래부)SW영재교육원 ·(미래부)SW창의캠프운영	·(미래부/교육부) 2018년까지 선도학교 900개 확대운용
중학교	· '과학/기술, 가정/정보' 교과군 신설 · 정보 과목을 SW교육 중심으로 개편 · 정보 과목을 필수로 34시간 이상 교육		
고등학교	· 정보 과목을 일반 선택으로 전환 운영 · SW 마이스터고 신설	· (교육부) 중,고생 대상 정보보호영재교육원 운영	

□ 소프트웨어 교육 자원 부족

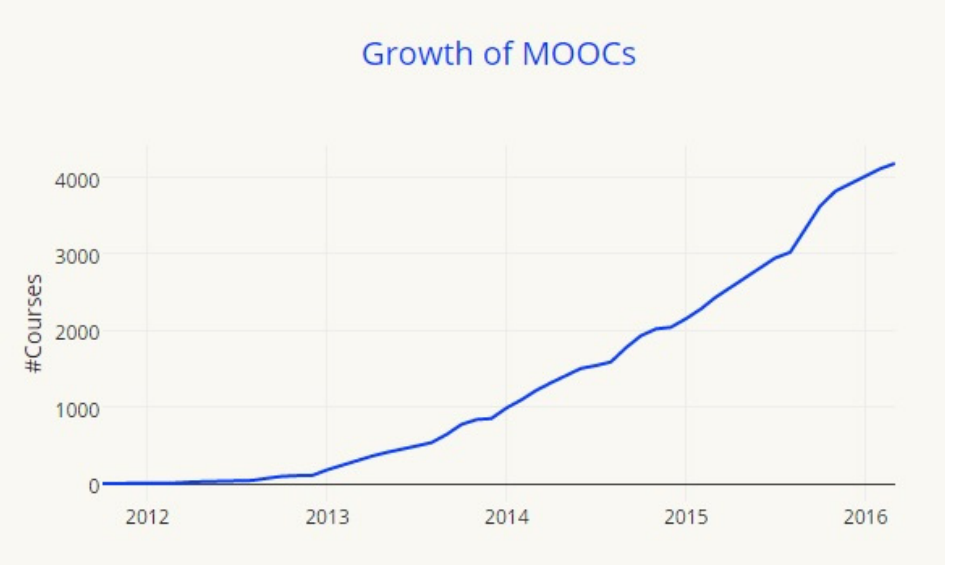
그러나 소프트웨어교육에 의욕적인 정부정책에도 불구하고, 중학교 '정보·컴퓨터' 자격증 소지 전체 교원은 1,739명이며, 이 중 '정보' 과목 담당 교사는 933명으로 교육 인력 부족과, 확보된 시수 중 상당 부분은 기존의 사이버윤리등의 내용을 포함하고 있어 문제해결능력을 키우기에는 충분하지 못한 시수 등이 지적되고 있다.

그 외에도 소프트웨어 교육을 위한 교육자료 부족, 지역별 격차, 교내 인프라 부족 등 한국의 소프트웨어 교육이 넘어야 할 산은 매우 많아 자칫하면 형식적인 교육으로 전락할 우려가 많아지고 있다.

□ 온라인 교육과 무크(MOOC), 혼합형 학습(Blended learning)

2015년은 MOOC가 이슈화된 해이기도 하다. 코세라, 에덱스, 퓨처런 등 다양한 MOOC 서비스가 출현한 이례로 3500만명 정도가 MOOC를 이용[10]했다고 한다.

[그림 1-5] 글로벌 무크의 성장



자료 : By The Numbers:MOOCS in 2015,, CLASS CENTRAL

대규모 공개 온라인 강좌인 무크(MOOC·Massive Open Online Course)는 온라인 교육이 오프라인 교육의 보조제가 아닌 독립적인 대체재로서 동작하고, 비용을 줄이고 오프라인의 교육 효과를 높일 수 있음이 증명되었다.

또한 이러한 온라인과 오프라인에 걸친 교육은 혼합형 학습(Blended Learning)은 데이터에 근거한 다양한 시도들이 이루어짐으로서 일방향의 오프라인 교육에도 변화를 가져오고 있다. 그 중 하나의 대표적인 운동 중 하나는 수업의 방식을 온라인 교육을 통해 바꾼 거꾸로 교실(Flpped Learning)이다. 국내에서는 2015년 3월에 KBS1을 통해 "거꾸로교실의 마법 - 1000개의 교실"이라는 방송을 통해 전해져 많은 학부모와 교사들 사이에서 화제가 되었고 거꾸로교실을 실천하는 교사 커뮤니티인 "미래교실네트워크"[11]는 매우 활발한 활동을 전개하고 있다.

10) By The Numbers:MOOCS in 2015,, CLASS CENTRAL

11) 거꾸로 교실을 확산하고자 하는 교사들의 커뮤니티 (http://www.futureclass.net/)

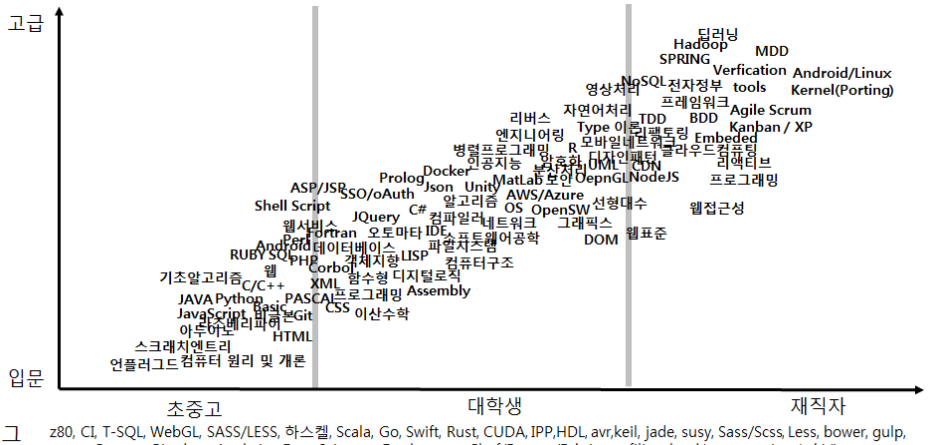
☐ SW교육의 문제점의 대안으로서의 온라인 교육

이러한 온라인 교육의 혁신은 소프트웨어 교육의 문제점의 대안으로도 가능하다. 온라인 교육은 하나의 콘텐츠가 매우 많은 이들에게 전달이 가능함을 MOOC를 통해 확인했으며, 거꾸로 교실을 통해 학교 안에서는 실습과 활동중심의 교육을 집에서는 개념학습등을 진행할 수 있음을 확인할 수 있다. 또한 소프트웨어 교육의 가장 큰 문제점이라고 할 수 있는 지역간 격차와 학교의 기자재의 문제도 집에 보유하고 있는 개인 PC나 개인의 스마트폰 단말을 이용하여 많은 부분을 극복할 수 있겠다고 할 수 있다.

☐ 심화 소프트웨어 학습으로서의 온라인 교육의 가능성

또한 소프트웨어의 지식 특성 상 빠르게 변화하고, 다양한 분야들이 존재하며 이런 소프트웨어 학습에는 나이 제한이나 자격 조건이 따로 있지 않다. 때문에 소프트웨어에 재능을 보이는 일부 아이들의 경우, 스스로 앱이나 웹서비스를 개발할 수 있는 수준에 오르는 경우도 혼하다.

[그림 1-6] 수준별로 다양하고 동적인 소프트웨어의 지식



그 외 z80, CI, T-SQL, WebGL, SASS/LESS, 하스켈, Scala, Go, Swift, Rust, CUDA, IPP, HDL, avr, keil, jade, susy, Sass/Scss, Less, bower, gulp, grunt, Opencv, Big data, Analytics, Data Sciences, Deployment Chef/Puppet/Fabric, profiling, load/stress testing, LabView, PLC(programmable logic controller), Ladder, MPS, AVR, ARM, ST(Structured Text), SFC(Sequential Function Charts), IL(Instruction List), FBD(Function Block Diagram), Robotino, CDN, , 보안, Atomic, CoreOS, Ubuntu Core

자료 : 직접 정리, 소프트웨어정책연구소

이는 소프트웨어교육이 다른 교육과 달리 교사가 학생들 보다 지식면에서 월등히 앞서

서 교육을 한다는 것이 불가능함을 의미하기도 한다. 이러한 학생들은 향후 소프트웨어 전문인력으로서의 충분한 자질을 가지고 있고, 이미 무언가를 만들어 보고 싶은 것이 구체적으로 있을 수도 있으나 교사가 모든 것을 가르쳐 줄 수가 없다. 따라서 스스로 성장할 수 있는 기회와 동기를 제공하고, 교사는 옆에서 방향을 설정하고 조언을 줄 수 있는 환경적인 부분이 필요하다. 이러한 부분에서는 온라인이 큰 역할을 할 수 있다.

실제로 전문 소프트웨어 인력들도 온라인에서 오픈소스나 커뮤니티 활동을 통해 소통하고 함께 학습하고 성장하고 있다.

□ 소프트웨어의 협업을 경험해 볼 수 있는 온라인 교육

또한 온라인은 소프트웨어가 가지고 있는 가장 기본적인 특질인 협업에 대한 경험과 학습에도 용이하다. 많은 교육이 개인이 특정 지식을 머릿속에 넣고 이것을 확인하는 과정이라면 다른 이들의 소프트웨어에 연결하고 함께 만드는 과정을 통해 창의적 문제해결능력을 학습하는 경험적 교육이 절실하며 이를 온라인 교육을 통해 시도해볼 수 있다. 전세계의 공개 소프트웨어가 올라오는 깃허브는 전세계 소프트웨어의 놀이터이자 학습장으로 사용되고 있는 것도 이러한 맥락이라 할 수 있다.

제 2 절 SW 온라인교육의 현황과 중점 연구 범위

1. 국내 SW 온라인 교육 현황

앞서 온라인 교육이 국내 SW교육이 안고 있는 문제점의 대안이 될 수 있음을 이야기했다. 그러면 국내의 온라인 SW교육 현황을 좀 살펴볼 필요가 있겠다.

〈표 1-5〉 국내 SW 온라인교육 현황

서비스명	대상	특징	제공 콘텐츠
Junior SW	초중등	·NIPA, 경인교대 주관 ·동영상 및 플래시 강좌	· 3개 언어 · 피지컬 컴퓨팅 4개 · SW기초 이론 2개
소프트웨어야 놀자	초중등	·네이버 주관 ·엔트리(Entry) 소재로 한 짧은 영상 강좌와 교재 제공	· 1개 언어 · SW교양 2개 · 피지컬 컴퓨팅 1개 (예정) · SW기초 이론 1개 (예정)
OLC	대학생 재직자	·NIPA와 KOSSA 주관 ·강좌 및 행사 동영상 제공	· 언어 9개 · 심화 과정 13개
ODIY	초중등	·초, 중등 대상 ·한국과학창의재단 주관 ·초소형 컴퓨터 학습 목적	· 피지컬 컴퓨팅 3개 · SW교양 1개
생활 코딩	일반인	·개인이 운영하는 비영리 사이트 ·포스팅 형식의 공개 강좌	· 언어 9개 · 심화 분야 6개
아이티동스쿨	일반인	·인터넷 강의 전문 사이트 ·다양한 강사진	· 언어 10개 이상 · 컴퓨터관련 자격증 강의 제공 · 패키지 강의 5개
TOPCIT	일반인	·ICT 산업 전문 강사진	· SW, DB, Network 등 기초 지식 관련 동영상 · IT 비즈니스 관련 동영상
구름	초중등 대 학 생 재직자	·클라우드 코딩 서비스	· 다양한 SW 교육 환경 제공 (C/C++, Java, Python 등 총 9종 언어) · 프로그래밍 시험 서비스 및 자동채점 기능 제공 · 소스코드 유사도 검사 기능

현재 국내의 온라인 SW교육의 현황을 보면 과거 정보처리기사의 자격증 위주의 동영상 강좌를 제공하는 다수의 온라인 학원사이트들이 존재하고 있으며, 대부분의 서비스들은

동영상 위주의 지식전달형 서비스임을 알 수 있다.

앞서 말한 소프트웨어 교육에서 중요한 경험을 제공하는 곳은 초중등을 대상으로는 네이버에서 주관한 소프트웨어놀자가 엔트리와 일반인을 대상으로는 클라우드 코딩 서비스를 제공하는 구름 정도만 있어, 국내에서는 아직 온라인을 이용한 소프트웨어 교육을 전면적으로 확대하기에는 아직 많은 어려움이 있을 수 있으며, 그 내용적인 측면에서도 더 살펴볼 필요가 있을 수 있다.

2. 온라인 SW교육의 가치체인

먼저 온라인 SW교육을 이용하게 되는 학습자의 가치체인을 살펴보겠다. 가치체인이 중요한 이유는 해당 시장이 지속적으로 성장하기 위해서는 시장이 완결된 구조로 선순환을 하며 성장해야 하기 때문이다.

[그림 1-7] 온라인 SW교육의 가치체인

자료: 직접 작성, 소프트웨어정책연구소

〈표 1-6〉 온라인 SW교육 가치체인의 단계

단계	설명	사례
1.인식 및 동기유발	SW교육에 대한 동기를 유발 하는 단계	캠패인 및 광고
2.정보 검색	교육 정보를 찾는 단계	검색 서비스
3.선택 및 추천	개인에 적합한 교육 정보를 선택	추천 및 개인화 서비스
4.학습 및 평가	교육을 수행하고 검증하는 단계	학습 및 교수 서비스
5.공유 및 활용	학습 결과물에 대한 활용 및 공유	소셜 학습, 갤러리

□ 인식 및 동기 유발

학습자는 소프트웨어 교육에 대한 필요성을 확인하는 “인식” 단계부터 시작한다. 현재 소프트웨어 교육에 대해 생소한 많은 이들을 위해 집중해야 되는 부분이기도 하다. 이를 위해서는 다양한 캠패인이나 광고 활동을 진행해야 하며, 사회적 분위기 조성이 중요한 부분이다.

□ 정보 검색

두 번째는 정보“검색” 단계이다. 정보가 많지 않은 경우에는 카테고라이징이나 집중화로 해결할 수 있다. 그러나 정보가 다양해지고 많아지게 되면 검색이 매우 중요해진다. 또한 온라인의 많은 사용자가 포털의 통합검색이나 모바일 마켓플레이스의 검색창을 통해 정보를 찾는다. 이 때 사용되게 되는 것도 “검색”이다. 이 때 원하는 필요로한 정보를 찾기 어렵다면 애써 확보한 학습자의 동기는 급격하게 떨어진다.

□ 선택 및 추천

세 번째는 “선택” 및 추천 단계이다. 학습자의 개인에게 적합한 교육정보가 추천되어져야 한다. 현재 온라인 SW교육의 경우에는 대다수가 일반인들을 대상으로한 교육이다. 따라서 초중고의 학생들에게는 초중고에 필요로한 온라인 SW교육정보를 확충해야 한다.

☐ 학습 및 평가

네 번째 학습 및 평가는 실제 학습을 하는 단계이다. 앞서 말한데로 실제 온라인 환경에서 직접 소프트웨어를 구동하고 만져보면서 학습을 하고, 피드백을 받음으로서 지속적인 활동을 통해 개선하고 문제점을 알고 학습해 나가는 과정을 거친다. 현재 한국의 많은 교사들의 고민은 온라인 상에 학습이 가능한 교육 서비스들이 많지만, 실제 이를 통해 어떻게 아이들을 평가하고 학습에 실질적인 도움을 줄것인지에 대한 해답은 부족한 상황이다. 본 보고서에서는 이 부분에 대해 뒤에 집중적으로 다룰 예정이다.

☐ 공유 및 활용

다섯 번째, 공유 및 활용은 학습을 통해 만들어진 결과물들을 공유하고, 이를 이용해 다른이들이 재사용을 통해 활용 혹은 학습을 하게 하는 것이다. 오늘날은 소셜네트워크와 같은 온라인 플랫폼을 통해 오프라인의 친구관계를 통해 확산이 가능하여 서로의 동기를 강화하는 효과도 가져갈 수 있으며, 이를 통해 교육효과를 증진할 수 있다.

3. 온라인 교육 서비스의 제공 주요 서비스

두 번째는 온라인 SW교육의 사이트들이 제공하는 주요 서비스 형태에 대해 알아보자. 대부분의 사이트들은 꼭 하나의 형태의 서비스만 제공하는 것은 아니며, 각 사이트들이 내세우고 있는 특징에 따라 강조하고 있는 서비스가 다르다.

☐ 교육정보 제공

먼저 교육정보를 제공하는 서비스이다. 해당 정보는 사이트내의 정보 뿐만 아니라, 다른 사이트 정보, 교수법 및 수업자료, 오프라인에서 일어나는 수업정보등 다양한 정보들을 포함하고 있다.

이러한 교육정보 제공은 주로 메뉴나 카테고라이징을 통한 분류와 태그나 검색 기반의 검색 서비스로 이루어진다. 교육 수요자의 진입장벽을 낮추고, 교육 수요자들의 흥미를 끌게하는 것이 중요하다.

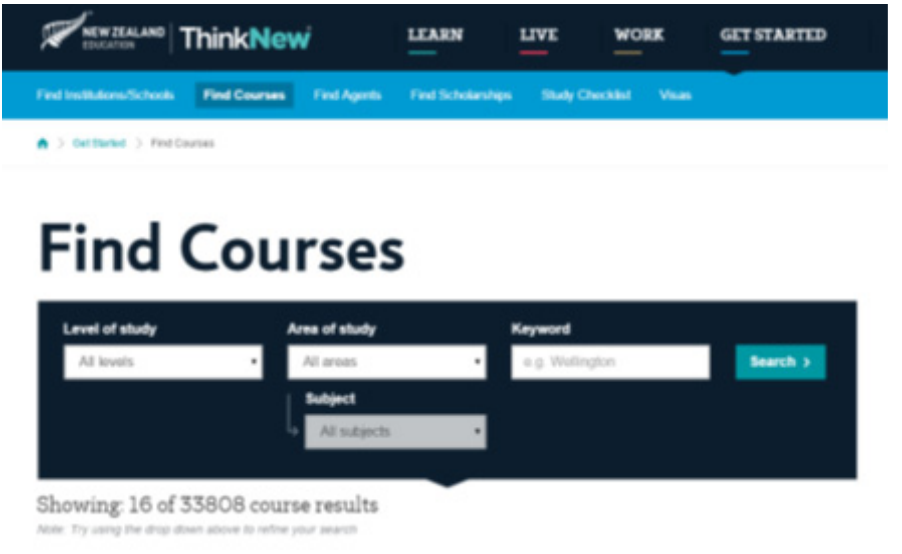
이러한 정보는 교사와 같은 교육공급자와 학생과 같은 교육수요자가 매우 요구하는 바가 다르며, 두 사용자층이 뒤섞이는 것은 서로의 활동을 방해하는 경우가 많아 두 개는 주

로 분리되어 제공되고 있다.

먼저 교육공급자의 경우, 수업정보를 공유할 수 있는 커뮤니티나 배포사이트 형태로 운영된다. 이러한 교육정보의 제공서비스에 가장 큰 이슈는 배포에 적합하지 않는 교육용 자료를 상호 교환하고자 하는 요구사항이 많아 외부로 공개된 형태로 운영되지 않는다. 많은 소프트웨어교육 서비스들이 교육수요자의 페이지를 앞에 내세우면서도, 공급자용 페이지를 별도로 두고 있는데 교사들이 소프트웨어교육의 품질을 결정하는 주요한 요인이기 때문이다.

교육수요자는 수요자에게 동기를 제공하거나, 적합한 교육 정보를 제공하는 것을 목적으로 한다. code.org의 Hour of code같은 경우, 아이들이 좋아할만한 콘텐츠나 유명인들을 내세워 동기유발에 적극적으로 나서고 있다. 검색의 개념으로서는 온,오프라인의 교육정보를 검색할 수 있는 뉴질랜드의 ThinkNew이 있으며, 다양한 오프라인의 소프트웨어 교육 클래스를 검색할 수 있는 CodeClub[12]이 이러한 유형에 해당한 다고 할 수 있다.

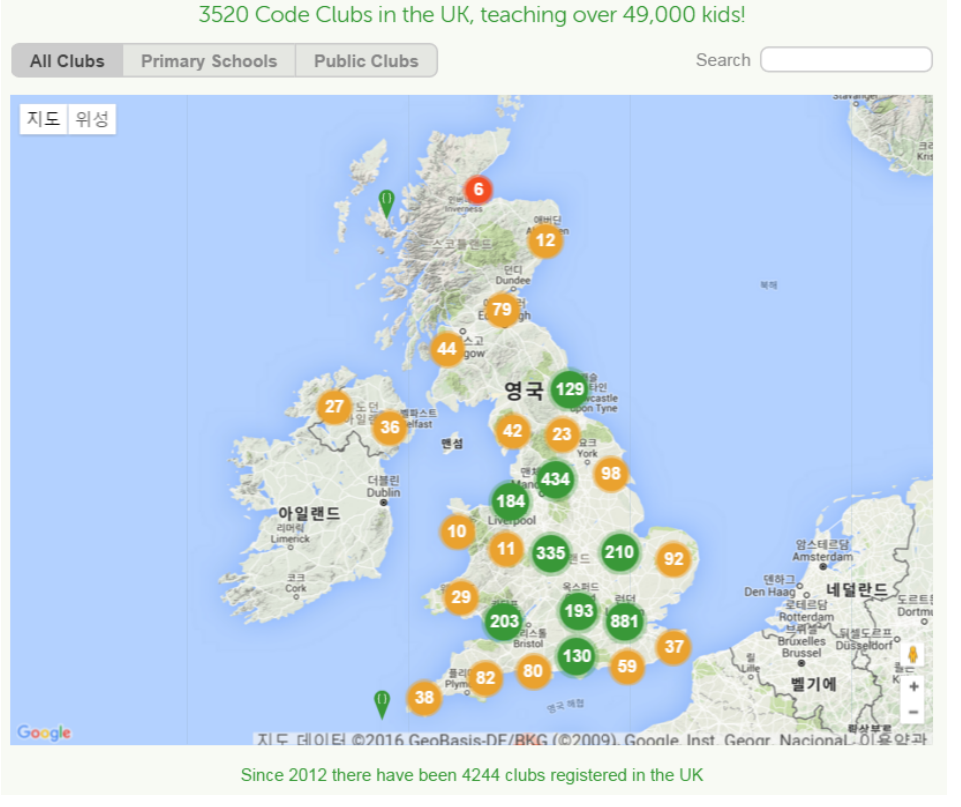
[그림 1-8] 온,오프라인의 교육 통합검색을 지원하는 ThinkNew



자료: ThinkNew 스크린샷

12) [참고] 부록, 1. CodeClub

[그림 1-9] 지도검색을 지원하는 코드클럽

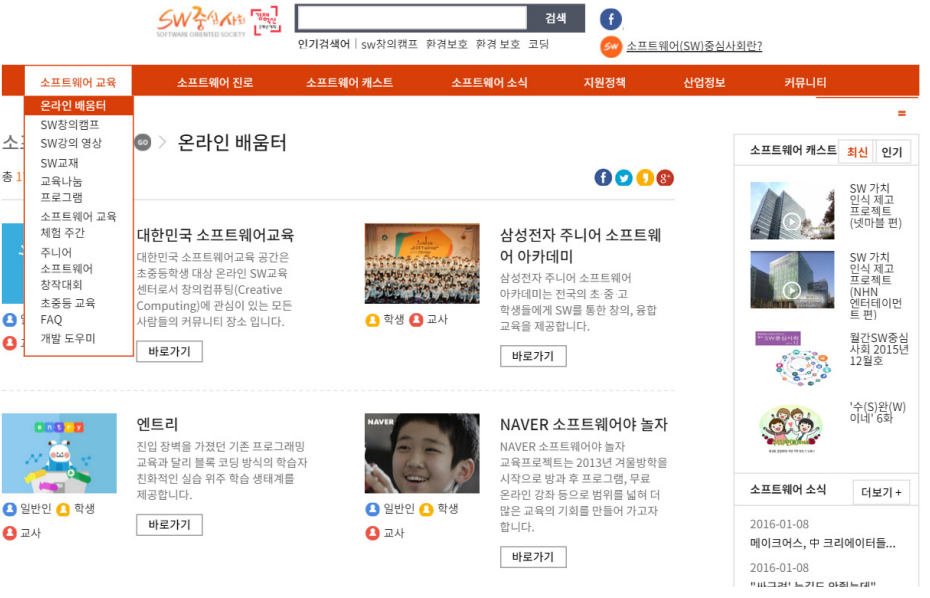


자료 : CodeClub 스크린샷

특히 현재 한국의 소프트웨어교육의 실정 상, 소프트웨어 교육에 대한 인식이 낮고 교육정보가 매우 부족한 상황으로, 미래창조과학부는 소프트웨어중심사회(http://softeware.kr) 포털의 첫 번째 메뉴에서 이러한 소프트웨어 교육정보들을 제공하고 있다. 그러나 소프트웨어 교육정보 외에도 산업 정보등의 지나치게 다양한 정보들이 모여 있어 교육 정보를 원해 방문한 사용자에게는 혼란을 주기 쉽도록 되어있다.

또한 콘텐츠가 수요자와 공급자에 의해 자생적으로 주고받는 플랫폼 형태가 아닌 사이트운영자의 정보배포처의 형태로 운영되고 있다보니 정보의 역동성과 품질 유지의 지속성이 어려운 형태이다. 향후 소프트웨어 교육을 주제로 하고 민간 교육전문단체 중심의 자생적인 운영을 하는 전문 온라인 서비스 플랫폼이 필요로 하지 않을까 싶다.

[그림 1-10] 소프트웨어중심사회 포털



자료: 소프트웨어중심사회 포털 스크린샷

□ 동영상 강좌

동영상 강좌의 온라인 교육 사이트는 이러닝의 가장 일반적인 형태이다. 교사의 강좌를 온라인으로 옮긴 형태로 오프라인에서 하던 강좌를 녹화하여 제공하므로, 실질적으로 가장 제작에 대한 특별한 고민이 없이 제작이 가능하다.

MOOC와 같은 서비스가 가장 발전한 동영상 강좌 서비스의 한 형태이다. MOOC는 edX, Udacity[13], Coursera 등이 있으며, 유·무료의 다양한 공개강좌들을 제공하고, 인증을 제공하는 곳 들이 있다. 이러한 대단위의 서비스에서는 동영상 인프라의 안정적인 운영에 큰 비용이 소요된다. 특히 전국의 초중고 학생들을 모두 이용하는 동영상 강좌 서비스를 만든다는 것은 어려운 일이다.

콘텐츠 제작면에서는 단순히 오프라인의 동영상 강좌를 녹화해서 업로드하는 것만으로

13) 참고 : [부록] 2.Udacity

는 온라인에서는 크게 효과가 떨어진다. 눈을 맞추고 강의를 진행해도 클래스의 많은 아이들이 잠이 드는데, 하물며 언제든 다른 브라우져를 띄울 수 있는 온라인에서의 집중유도는 쉬운 일이 아니다. 또한 스마트폰이 보급된 요즈음은 온라인 콘텐츠를 이동시간이나 짜투리 시간에 소비하는 경향이 짙다. 때문에 온디멘드 서비스들은 비교적 짧은 시간에 정보를 전달하기 위해 내용을 20분미만 이나 수분 이하로 자르고 작은 모바일 화면에도 시청이 가능하도록 텍스트 가독성 등을 신경을 써서 개발을 하고 있다. 이러한 콘텐츠는 기존의 강의 동영상을 그대로 녹화해서 올리는 대신 온라인 배포용으로 별도로 개발을 해야 한다.

□ 인터렉티브한 체험 학습 서비스

인터렉티브 학습은 사용자가 온라인의 미션을 수행하여 피드백을 통해 학습하는 형태이다. 문자나 영상의 읽기나 시청 뿐 아니라 마우스나 키보드로 체험적인 서비스를 제공하는 서비스를 제공한다. 한국에서는 주로 플래시 같은 기술을 이용하여 콘텐츠가 제공 되었다. 가장 광범위하게 사용된 인터렉티브 기능은 앞서 설명한 내용의 이해도를 확인하는 퀴즈를 푸는 형태의 학습이 많았다.

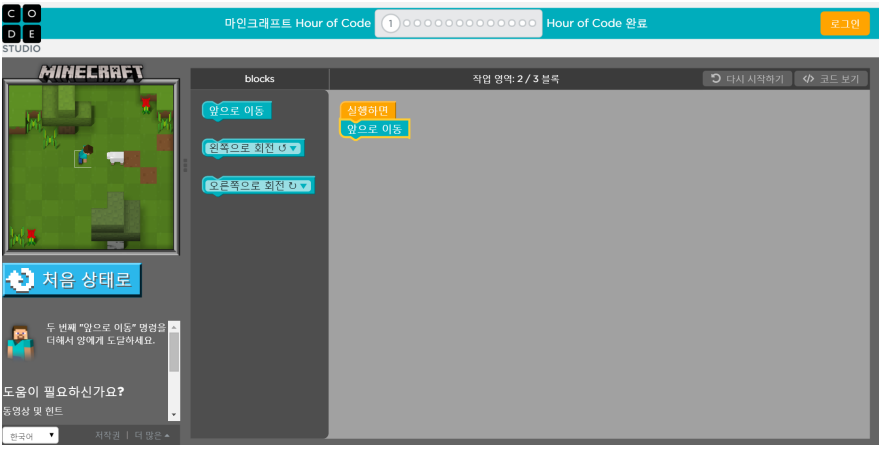
소프트웨어 교육의 경우에는 주로 활용 측면의 교육이 많아 특정 소프트웨어의 사용법을 가르치기위해 소프트웨어와 유사한 환경을 플래시로 보여주면서 간단한 기능을 실행해 보는 소프트웨어의 활용 교육이 다수가 있었다.

그러나, 소프트웨어의 학습은 주어진 정해진 미션을 정해진 방법으로 풀기 보다는 다양한 방법의 문제해결과 실습을 통한 직접 경험이 중요함에 따라 실제 프로그래밍이나 실습이 가능한 환경에서 미션을 수행하는 형태로 인터렉티브 학습 서비스들이 등장하고 있으며, 소프트웨어 교육이 갖는 목적인 문제해결능력을 위해서는 문제를 다양한 방법의 접근의 시도의 경험이 더 중요하므로 이러한 실습과 자유도가 높은 체험을 통해서만 목표한 바에 달성이 가능하다. Code.org나 엔트리의 경우 동영상 개념 강좌 이후, 미션을 수행하고 힌트를 주는 형태의 인터렉티브한 학습 서비스를 제공하고 있다.

[그림 1-11] 정해진 형태의 조작만 가능한 Junior SW

자료: Junior SW강좌 스크린샷

[그림 1-12] 블록을 조합을 해봄으로서 문제해결을 시도 해볼수 있는 code.org



자료: Code.org 스크린샷

4. 온라인 소프트웨어 교육 콘텐츠 유형

온라인 소프트웨어 교육을 콘텐츠의 기준으로 분류하면 비쥬얼 프로그래밍, 텍스트 프로그래밍, 피지컬 컴퓨팅 정도로 나눌 수 있다. 콘텐츠의 형태에 따라 산출물도 크게 차이나고 난이도와 장단점이 다르다. 어떤 형태가 가장 최선의 형태라기보다는 추구하는 바가 달라 나타난 형태라고 할 수 있다.

<표 1-7> 온라인 SW교육의 유형

유 형	특 징	사 례
비쥬얼 프로그래밍	명령어 기능을 수행하는 코드를 가진 블록을 조합하여 프로그래밍 직관적인 그래픽을 사용하여 간편하고 사용하기 쉽다.	스크래치(Scratch) 엔트리(Entry) 코두(Kodu)
텍스트 프로그래밍	코드를 작성하여 프로그래밍 컴퓨터 구조와 라이브러리를 알아야 프로그래밍 구현 가능	파이썬(Python) C, C#, Java
피지컬 컴퓨팅	하드웨어와 융합한 형태로 SW를 이용하여 하드웨어를	로봇 프로그래밍 아두이노(Arduino)

□ 비쥬얼 프로그래밍

비쥬얼 프로그래밍의 장점은 시각적으로 이해하기 쉽다는데 있다. 비쥬얼 프로그래밍의 대표주자는 해외에서는 MIT에서 개발한 스크래치, 국내에서는 엔트리 교육연구소에서 개발한 엔트리가 있다. 비쥬얼 프로그래밍은 가시성을 목표로 만들어진 프로그래밍 방법으로서 프로그램 구조를 시각적으로 이해하기 쉽게 만들기 위함이 었다. 일반적으로 특정 알고리즘이나 동작을 의미하는 심볼들의 조합으로 프로그래밍을 짜는 형태로, 플로우차트나 간트차트 UML을 이용한 프로그래밍 도구가 여기에 해당된다. 비쥬얼 프로그래밍은 전문 소프트웨어 엔지니어들에게는 선호되지 않는다. 먼저 키보드만으로 빠른 개발을 원하는 프로그래머들에게 선호되지 않고, 완성된 프로젝트는 성능 문제를 야기하거나, 아주 복잡한 프로젝트의 경우 디스플레이 화면에 보기가 어렵다는 단점들이 존재한다.

그러나, 이 시각적 이해의 우수성은 교육용으로 적합하다. 여기에 착안해서 개발된 MIT의 스크래치는 마치 블록을 조합해서 프로그래밍을 할 수 있도록 만들어졌고, 이를 벤치마크하여 개발 후 독자적인 개량을 하고 있는 것이 국내의 엔트리다.

현재 이러한 비쥬얼 프로그래밍 도구들은 이해가 용이하다는 장점 때문에 초등학교 정도의 저학년이나 입문자의 학습 도구로 유용하다. 그러나 일반적인 프로그래밍 언어가 포괄하고 있는 객체지향이나 다양한 디자인패턴과 링킹과 같은 프로그래밍의 모든 개념을 포괄하고 있지는 못하다. 비쥬얼프로그래밍의 목적은 일반적으로는 소프트웨어교육에 흥미를 느끼도록 하게 하고, 소프트웨어 동작의 기본적인 원리를 이해시키고자 함이기 때문에 요구사항이 강력하지 않기 때문이며, 소프트웨어교육에서 전문교육이 아닌 소양교육으로 진행하는 경우에 적합한 도구이다.

□ 텍스트 프로그래밍

텍스트 프로그래밍은 가장 일반적인 프로그래밍 방법이다. 컴퓨터 프로그램의 언어로 가장 잘 알려진 C나 JAVA등이 여기에 해당되며, 실제 전문 소프트웨어 엔지니어들의 동일한 경험을 요구한다. 매우 강력하며 거의 모든 소프트웨어의 개념을 학습할 수 있다. 그러나, 텍스트 프로그래밍의 언어는 해외에서 만들어져 특별한 언어를 제외하고는 거의 모든 언어는 영어다. 매우 단순한 코드라 하더라도 영어의 장벽으로 인해 한국의 학습자에게 느껴지는 벽은 매우 높은 편에 해당한다. 또한 초기 프로그래밍 환경 셋팅에 어려움을 겪는 경우가 많고, 컴파일이라는 과정을 거쳐 개발이 진행되는데 소프트웨어에 대한 깊은 이해를 요구한다.

하지만, 텍스트 프로그래밍은 사용자가 만들기 원하는 대부분의 소프트웨어를 만들 수 있는 유일한 수단이며, 전문 소프트웨어 교육을 위해서는 실제 코딩을 해보는 것이 가장 중요하고 유효한 학습 수단이다.

□ 피지컬 컴퓨팅

피지컬 컴퓨팅은 프로그래밍의 활동만을 하는 것이 아닌 만든 소프트웨어가 로봇 이나 기계장치를 통해 실제 동작하는 것을 확인하는 활동이다. 비쥬얼 프로그래밍의 일종인 스크래치나 엔트리와 연동한 아두이노 프로그래밍, 레고의 마인드스톰 같은 교구들이 이러한 예이다.

피지컬 컴퓨팅의 장점은 물리적인 장치를 동작 시킴으로서 아이들이 만지고 동작함에

따라 높은 흥미를 유도할 수 있다는데 있다. 보이지 않는다는 소프트웨어의 특징은 아이들에게 이해하기 어려운 부분일 수 있으므로 실제 물리적으로 보이는 것들을 소프트웨어를 통해 움직여 봄으로서 학습효과를 높일 수 있다.

피지컬 컴퓨팅은 소프트웨어 중에는 가장 난이도가 높은 임베디드 프로그래밍이다. 이는 소프트웨어와 하드웨어의 이해를 모두 필요로 한다. 소프트웨어가 하드웨어에서 어떻게 동작하는 지를 이해해야 하는 것이 핵심이다.

그러나, 피지컬컴퓨팅의 활동의 단점은 하드웨어 위주로 흐르기 쉽다. 제한된 시간에 움직이는 로봇을 만들기 위해서는 정해진 목표대로만 따라해보게 되는데, 이때 대부분의 시간을 하드웨어를 조립하는데 시간을 소모하기 쉽다. 결국, 부족한 시간에 소프트웨어는 정해진대로 입력만을 요구하여 소프트웨어를 고쳐보고, 물리적 기계가 피드백을 하면서 소프트웨어가 어떤 역할을 하는지를 이해할만한 시간이 없는 경우가 많다.

또한 피지컬 컴퓨팅은 비용의 문제가 있다. 최근에는 아두이노나 라즈베리파이 같은 범용 소프트웨어 교구들이 나와 값이 매우 저렴해졌으나, 여전히 구입해야 하는 비용은 필요로 하며 어떤 소프트웨어 교구는 매우 비싸서 초,중,고의 학교에는 부담이 될 수 있다.

[그림 1-13] 지식쇼핑 마인드스톰의 가격비교 결과

자료: 네이버지식쇼핑 스크린샷

5. 중점 연구 범위

이상, 온라인 SW교육에 대한 전체적인 현황을 살펴 보았다. 소프트웨어 교육에 온라인 교육은 매우 유효할 뿐 만 아니라, 이미 다양한 유형의 콘텐츠가 서비스되고 있다. 그러나 초,중,고를 연령 계층을 어우르면서, 앞서 말한 다양한 유형의 콘텐츠들을 포괄하고, 소프트웨어 교육이 목표한 창의적 문제해결 인재양성을 할 수 있는 유연한 온라인 소프트웨어 교육 체계는 아직 없는 상황이다. 그 중 국내의 경우, 온라인 소프트웨어교육에 시급하게

요구되는 바는 부족한 교육자원을 가지고도 많은 학생들을 효과적으로 가르치고, 수준을 측정할 평가방법과 지속적으로 소프트웨어교육을 개선하고 교육자와 학생들이 만들어내는 산출물들을 저장하고 관리하면서 학습을 할 수 있는 레파지토리 공간의 문제가 있다. 뒤이어서는 다양한 사례 분석을 통해, 온라인 소프트웨어 교육 체계 중 가장 밑바탕이 되는 레파지토리와 그 위에서의 활동들을 어떻게 평가할지에 대한 평가모델에 대해 이야기를 하고, 초중고를 대상으로한 이상적인 온라인 소프트웨어 교육체계의 레파지토리와 평가 모델에 대한 이야기를 하겠다.

제2장 온라인교육 서비스의 사례 분석

제1절 소프트웨어 교육 콘텐츠 유형별 분석

소프트웨어 교육서비스들은 콘텐츠의 형태에 따라 크게 영향을 받는다. 다양한 소프트웨어 교육 콘텐츠들이 있으나, 교육용으로 많이 사용되는 콘텐츠는 입문용으로 선호되는 비쥬얼프로그래밍과 보다 본격적으로 소프트웨어를 만들어보는 텍스트프로그래밍이 있으며, 최근에는 물리적인 기계에 탑제되는 소프트웨어를 통한 학습을 하는 피지컬컴퓨팅이 있다.

각각의 프로그래밍 사례를 분석하여, 해당 사례의 레파지토리와 평가모델의 시사점을 알아보겠다.

1. 비쥬얼 프로그래밍

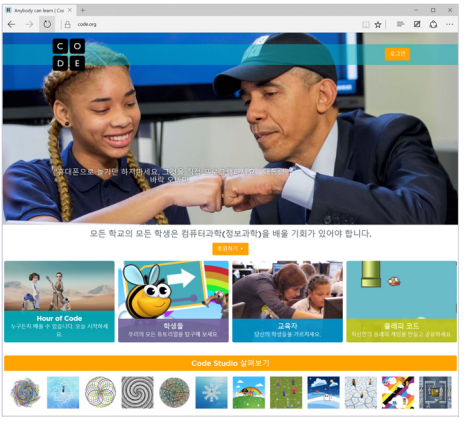
□ Code.org (http://code.org)

Code.org는 하루에 한 시간 코딩을 통해 모든 이들이 컴퓨터를 쉽게 느낄 수 있도록 하자는 미국의 대표적인 Computational Thinking 캠페인 Hour of Code를 주관하는 비영리기관으로 영유아와 초등학생들을 대상으로 입문 수준의 비쥬얼 프로그래밍 교육을 제공하며 아래와 같은 특징들을 가지고 있다.

- 부분적으로 한글을 지원하며, 인종, 나이, 성별과 상관없이 보다 많은 학교들에서 컴퓨터과학(정보과학)을 가르칠 수 있도록 하는 확산 운동을 추진하고 있다.
- Google, Microsoft, Infosys, Facebook 등 유수의 기업과 Bill Gates, Reid Hoffman 등 200여 기부자가 거액을 기부하여 운영 중에 있다.
- 모든 학생이 컴퓨터과학(정보과학)을 배울 기회를 얻게 되면 이를 통해 현실에서의 문제 해결 능력, 논리와 창의성을 키워주어 21세기의 생활과 진로탐색에 있어서 새로운 경험들과 성공들을 만들어 냄을 목표로 한다.
- Hour of Code는 K-12뿐 아니라 보다 다양한 연령대가 코딩을 쉽게 경험하도록 동영상과 튜토리얼로 이루어진 커리큘럼을 실행환경인 Code Studio와 함께 제공한다.

- 전세계 180개 이상의 40여개 이상의 언어로 준비되어 보급되는 운동으로 1억명 이상이 경험한 것으로 보고되었다.

[그림 2-1] Code.org 온라인 사이트(http://code.org)

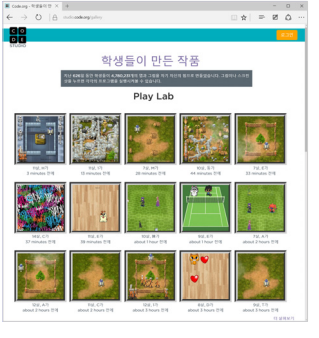


자료: Code.org 스크린샷

○ **레파지토리 측면의 code.org**

- Code.org는 code studio라는 학생들이 구현한 앱과 그림을 저장하고 공유하는 레파지토리 서비스를 제공한다. 해당 그림이나 스크린 샷을 클릭하면 실행 결과를 볼 수 있다.

[그림 2-2] code.org의 레파지토리 code studio



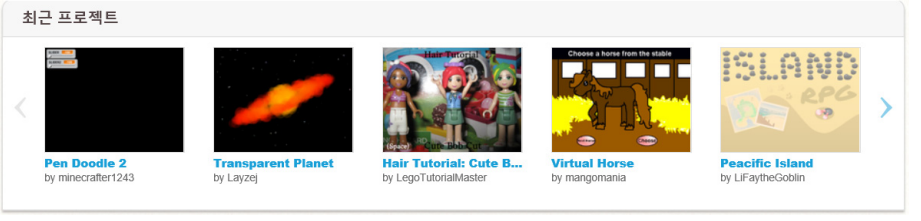
자료: Code.org 스크린샷

- codestudio는 다음과 같은 항목들이 존재한다.

· 최근 프로젝트

학생들이 최근에 제작한 스크래치 프로젝트를 등록하는 기본 공간이다.

[그림 2-3] code studio 의 최근 프로젝트

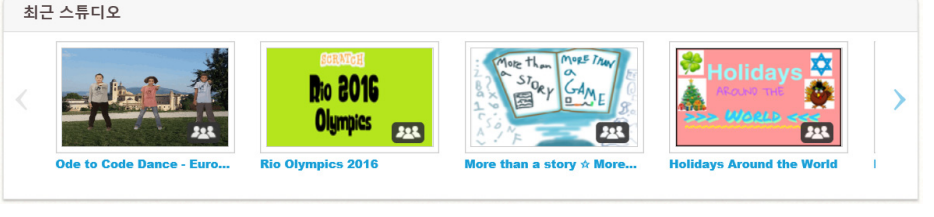


자료: Code.org 스크린샷

· 최근 스튜디오

학생들이 제작한 스크래치 프로젝트를 등록하고 다른 학생들과 의견 교환을 통하여 더욱 발전 방향으로 진행할 수 있는 공간이다.

[그림 2-4] code studio 의 최근 스튜디오

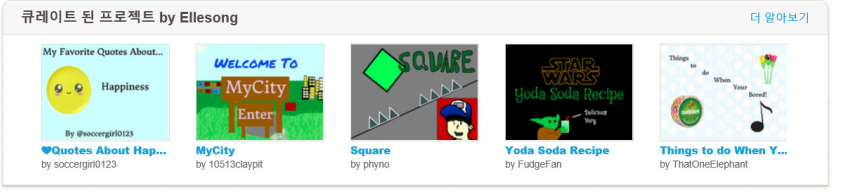


자료: Code.org 스크린샷

· 큐레이트된 프로젝트

Manager 및 Curator 들이 등록 스크래치 프로젝트에 대하여 의견을 등록하는 공간이다.

[그림 2-5] code studio 의 큐레이트된 프로젝트

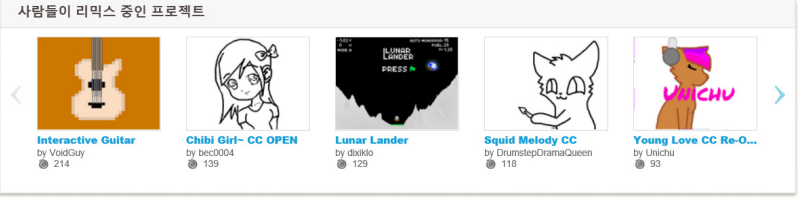


자료: Code.org 스크린샷

· 사람들이 리믹스중인 프로젝트

등록된 스크래치 프로젝트를 공유하여 새로운 아이디어를 접목하고, 수정하여 더욱 발전 방향으로 진행할 수 있는 공간이다.

[그림 2-6] code studio 의 사람들이 리믹스 중인 프로젝트

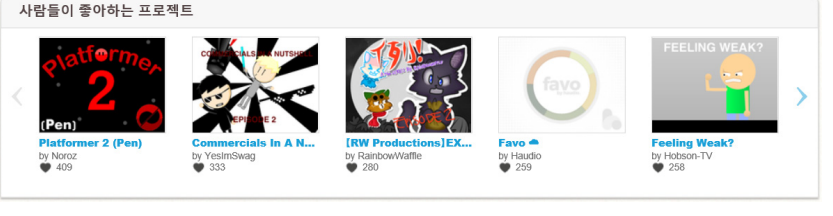


자료: Code.org 스크린샷

· 사람들이 좋아하는 프로젝트

학습자들로부터 많은 추천을 받은 스크래치 프로젝트들을 모아둔 공간이다.

[그림 2-7] code studio 의 사람들이 좋아하는 프로젝트



자료: Code.org 스크린샷

위 그림 에서 ♥(이 프로젝트가 좋아요) 버튼을 클릭하면 추천수가 증가하여 레파지토리를 통해 흥미를 가질만한 사용자 자료를 공유하고, 추천을 통해 상호독려를 하는 효과를 목적으로 한다.

- coder.org의 codestudio의 항목들을 자세히 살펴보면, 학습자들의 최근 프로젝트와 스튜디오를 노출하여 최근 정보를 개제하고, 큐레이트된 프로젝트를 통해 추전 정보를 제공하고, 소프트웨어의 특징인 재사용이 빈번한 리믹스 중인 프로젝트, 그리고 마지막으로 명시적으로 추천을 많이 받은 사람들이 좋아하는 프로젝트를 보여주고 있다. 이는 레파지토리에 올라온 자료들을 **시스템에 의한 추천은 시간(최근 프로젝트, 최근 스튜디오), 사용빈도(리믹스 중인 프로젝트)로 사람에 의한 추천은 운영자 큐레이션, 사용자 추천순** 으로 하고 있다.

  이는 사용자로 하여금 다양한 정보들을 여러 가지 관점에서의 좋은 콘텐츠를 쉽게 찾아볼 수 있게 하는 일반적인 방법으로 효과가 증명된 방식으로 레파지토리에서 콘텐츠를 어떻게 노출할지에 대한 참고가 될 수 있다.

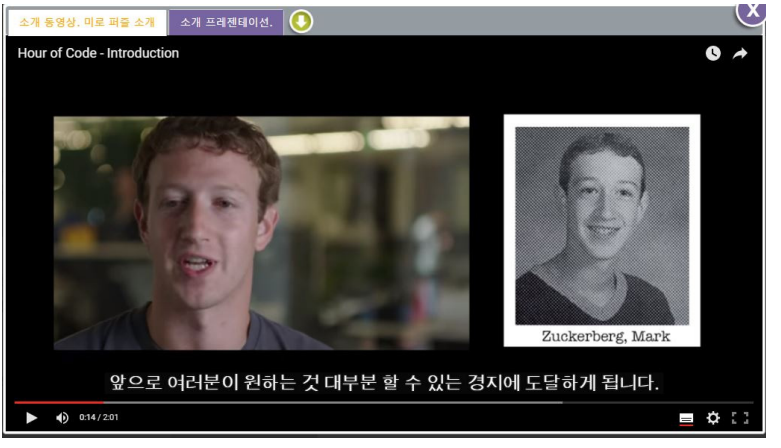
○ **평가 측면의 code.org**

- 또한, Hour of Code는 K-12뿐 아니라 보다 다양한 연령대가 코딩을 쉽게 경험하도록 **동영상과 튜토리얼로 연결된 커리큘럼을 실행환경인 Code Studio와 함께 제공하여 연속성**을 가지고 있다.

[그림 2-8] 20시간 소개 과정

자료: Code.org 스크린샷

[그림 2-9] 20시간 소개 과정 중 미로 퍼즐에 대한 설명 동영상



자료: Code.org 스크린샷

- **컴포넌트 블록들을 움직여 코딩을 하고 제출하면 해당 결과의 맞고 틀림에 대한 피드백이 즉각적으로 주어지며 틀린 경우, 추가로 힌트가 주어진다. 평가의 목적이 맞고 틀리고의 점수의 합산이 아닌 틀린 행동을 통해 맞는 답안이 나오도록 유도하는 데에 목적**을 가지고 있으며, **단계적 미션**을 제공하므로 큰 어려움 없이 진행이 가능하다.

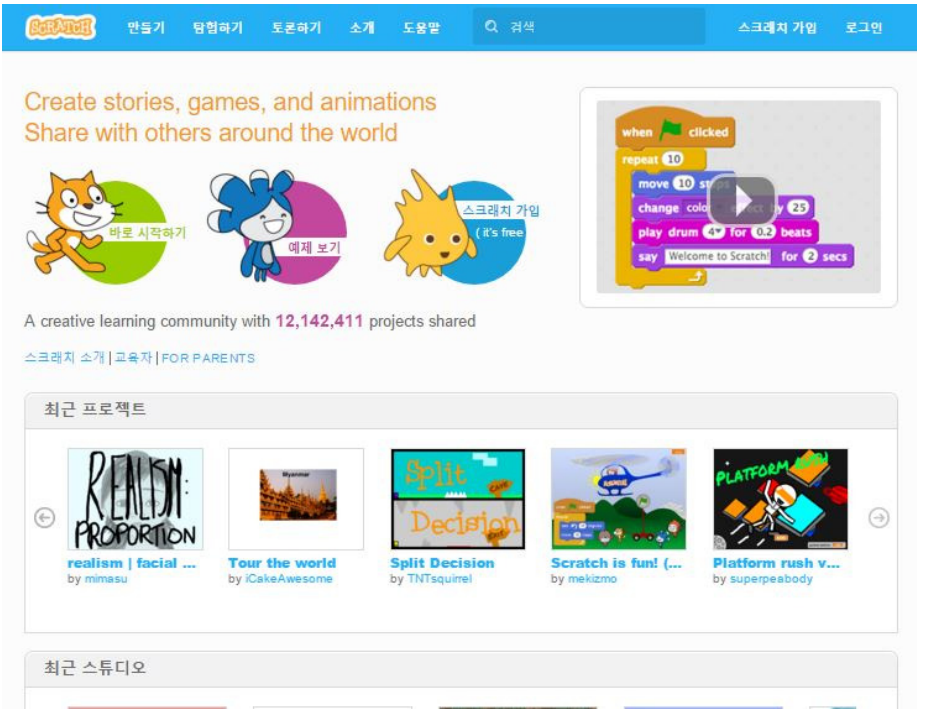
[그림 2-10] 미로퍼즐 미션

자료 : Code.org 스크린샷

□ Scratch (https://scratch.mit.edu/)

Scratch는 2007년 MIT 미디어랩에서 개발한 블록형 프로그래밍 언어이며, 현재 Lifelong Kindergarten Group 에서 운영하는 프로젝트로서 8세부터 16세까지 아동을 대상으로 설계됐으며, 작은 명령 단위인 블록 조각을 조립해 프로그래밍 할 수 있도록 만들어 졌다. Scratch는 어린이를 위한 대표적인 소프트웨어학습 도구로서 아래와 같은 특징을 가지고 있다.

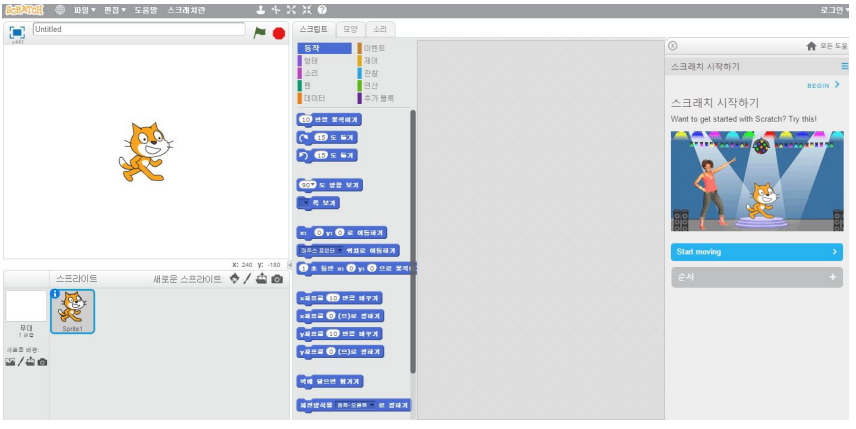
[그림 2-11] Scratch의 홈 화면



자료 : Scratch 스크린샷

- 기본적인 변수와 제어문을 사용할 수 있으며, 스프라이트를 객체로 사용해 각각 다른 명령을 넣어 프로그래밍할 수 있다.
- 오프라인 개발환경, 클라우드 저장, 사운드 편집, 그래픽 편집, 스프라이트 복사 기능 등이 있다.

[그림 2-12] Scratch의 에디터

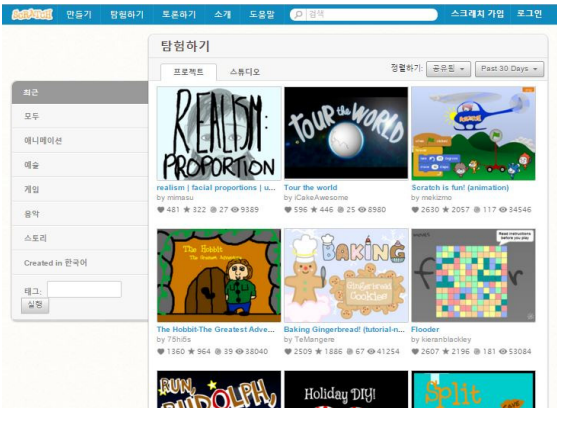


자료 : Scratch 스크린샷

○ **레파지토리 측면의 Scratch**

- 프로젝트 에디터로 다양한 인터랙티브 스토리, 게임, 애니메이션 등을 만들 수 있고, 탐험하기 메뉴를 통해 만들어진 프로젝트를 스크래치 온라인 커뮤니티에 공유 가능하며 2015년 12월 현재 1천200만 여개의 프로젝트가 공유되고 있다.

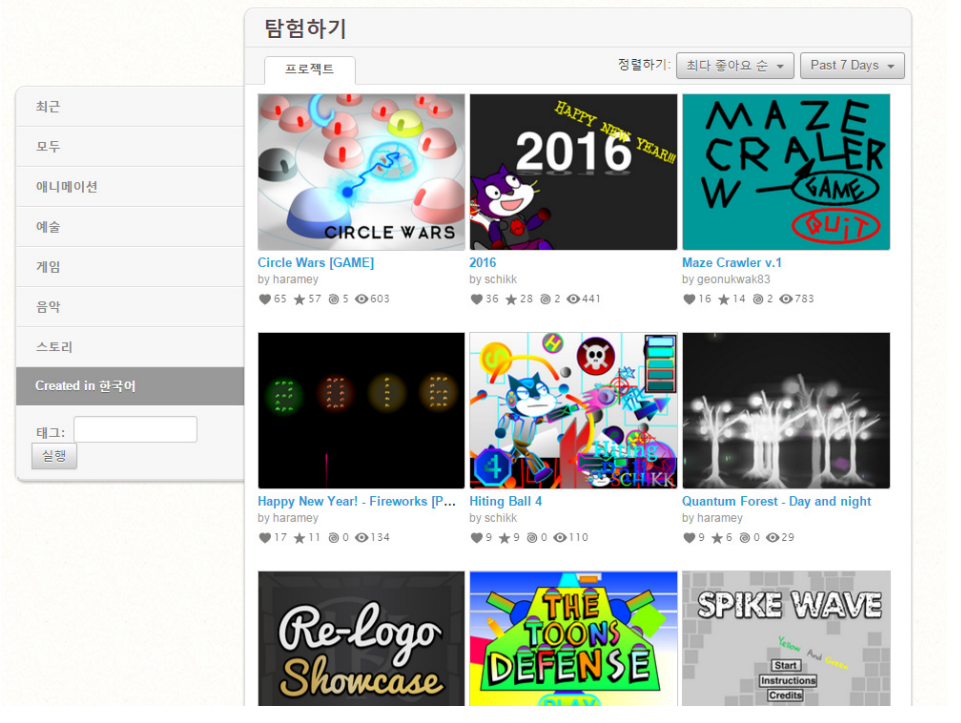
[그림 2-13] Scratch 탐험하기



자료 : Scratch 스크린샷

- 공유된 프로젝트는 다른 사용자에 의해 **좋아요, 댓글, 조회수** 등을 통해 피드백을 얻을수 있으며, **다른 사용자가 프로젝트의 소스를 확인하여 고쳐볼 수 있다.**
- 메인페이지에서의 프로젝트들을 추천해주는 방식은 codestudio와 동일하게 제시하고 있다.
- 탐험하기 페이지는 프로젝트들을 분류별로 보여준다. 애니메이션, 예술, 게임, 음악, 스토리, 그리고 한국어로 개발된 프로젝트들을 보여주도록 되어 있어 콘텐츠의 형식을 기준으로 분류되어있다.

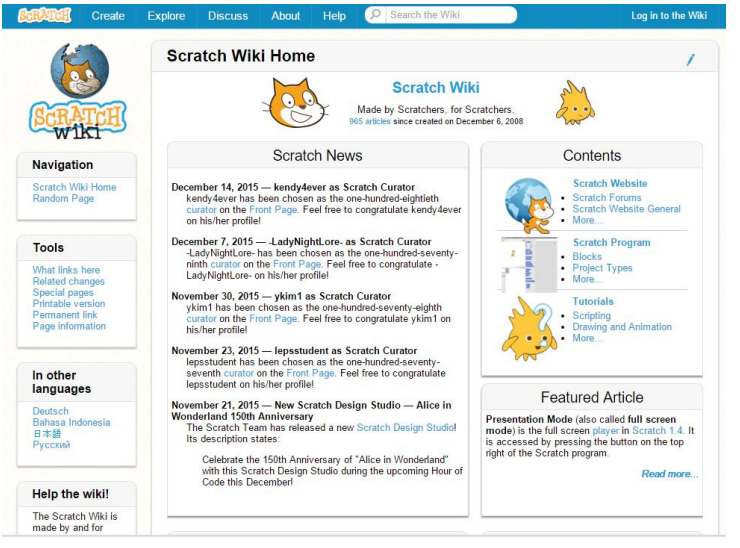
[그림 2-14] Scratch 콘텐츠 분류



자료 : Scratch 스크린샷

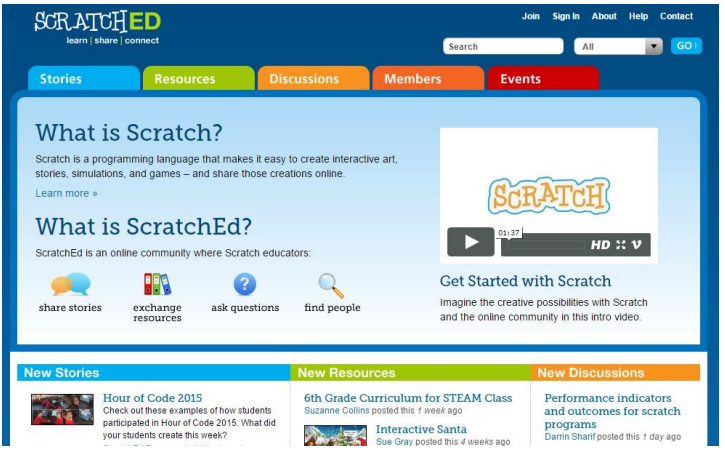
- 스크래치 위키에는 다양한 내용과 수준의 스크래치와 관련된 글과 튜토리얼 등도 포함하고 있어서 **교육자료실을 제공**한다.

[그림 2-15] Scratch Wiki



자료 : Scratch 스크린샷

[그림 2-16] Scratch ED



자료 : Scratch 스크린샷

- 교육자들을 위한 온라인 커뮤니티인 ScratchEd는 다양한 교육 이야기, 토론, 스크래치 교육과정 가이드와 같은 유용한 자료를 공유하고 있어서 **교육제공자를 위한 페이지와 정보를 따로 제공하고 있다.**

○ 평가 측면의 Scratch

- (추천과 댓글) 스크래치는 학습도구와 레파지토리 형태의 서비스로 별도의 평가시스템은 제공하고 있지 않다. 하지만, 레파지토리 내에 추천과 댓글 시스템을 통하여 자신과 타인에 대한 결과물의 사용자 반응을 알 수 있다. 이를 기반으로 다른 이들이 내 결과물을 어떻게 생각하는지를 학습자가 알수 있게 했다.

[그림 2-17] Scratch 프로젝트 프로젝트 페이지

자료 : Scratch 스크린샷

□ KODU GameLab (http://www.kodugamelab.com/)

KODU는 간단한 비주얼 프로그래밍 언어를 통해 어린이들이 PC와 Xbox 용의 게임을 만들 수 있는 Microsoft사의 도구로서, 디자인이나 프로그래밍 스킬이 없는 학생들에게 창의성과 문제 해결, 스토리텔링, 프로그래밍 등의 경험을 하도록 하고 있다.

PC 용 KODU Game Lab은 무료이며, Xbox 용 KODU Game Lab은 5$에 유료구매가 가능하며, 2013년 중반이후 사용자의 급격한 증가가 시작되어 2015년 12월 현재 등록된 사용자는 915만명에 이를 정도로 인기가 있으며 생활코딩 과 OLC 주니어에도 강좌가 실려 있다.

한국에서는 한글화가 미비된 점과 게임에 대한 부정적인 인식으로 인해 미국이 43.6%, 영국 12.3% 한국사용자는 1.83%인 16만에 불과한 실정이다. 하지만, 게임성이 높기 때문에 어린이들에게는 매우 높은 호응을 이끌어 낼 수 있으며, CodeStudio나 Scratch대비 더 많은 강력한 기능과 인터렉티브한 경험을 제공하여 국내 소프트웨어교육에 좋은 도구로 활용될 수 있다.

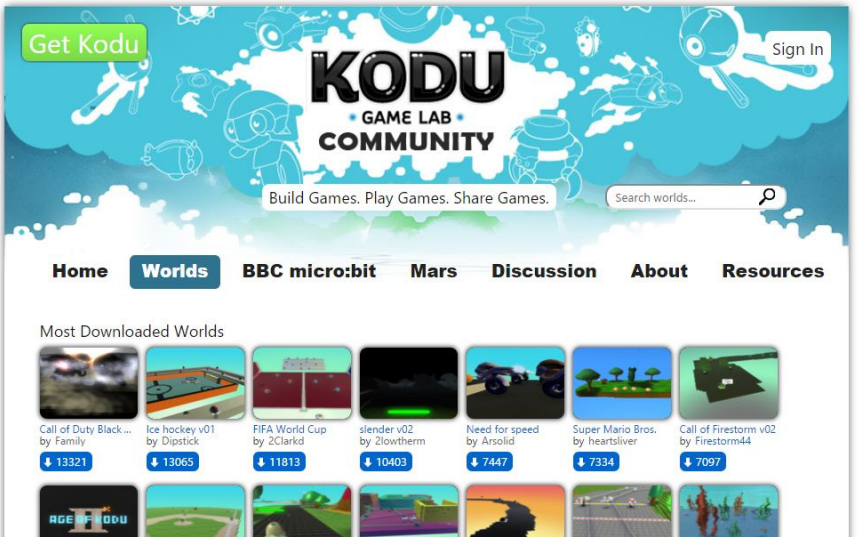
[그림 2-18] Kodu Game Lab community 메인 페이지

자료 : Kodu gamelab 스크린샷

○ **레파지토리 측면의 Kodu gamelab**

만들어진 결과물을 공유하는 Worlds라는 커뮤니티 기능이 있으며, 1200만개의 공유 프로젝트가 있으며 이에 대한 코멘트 포스트는 6천200만건에 이른다.

[그림 2-19] Kodu gamelab의 Worlds



자료 : Kodu gamelab 스크린샷

○ **평가 측면의 Kodu gamelab**

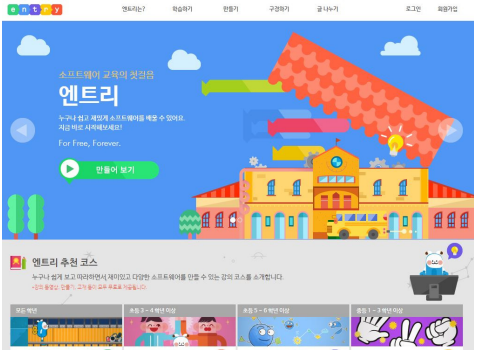
평가에 대한 별도의 기능을 제공하고 있지않으며, **댓글**을 통한 커뮤니티 기능만 제공하고 있다.

□ Play-Entry (http://play-entry.org/)

스크래치를 벤치마킹하여 개발한 비쥬얼 프로그래밍 SW학습도구로서, EBS[14]와 NAVER의 공동기획 '소프트웨어야 놀자'의 기본 플랫폼으로 사용되고 있다.

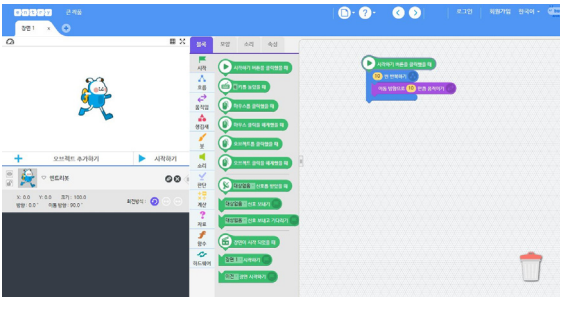
인터넷 연결 없이 수행되는 오프라인 버전도 제공하며, 엔트리봇 센서보드를 통해 엔트리에서 만든 프로그램과 아두이노를 연결 가능하고, 프로그램에서 에러가 날 때 어디서 에러가 났는지 제시하는 피드백 가능하며 블록이 이벤트의 어느 부분에 사용된 것 추적하는 다양한 기능을 제공한다.

[그림 2-20] 엔트리 홈



자료 : 엔트리 스크린샷

[그림 2-21] 엔트리의 프로젝트 제작 화면



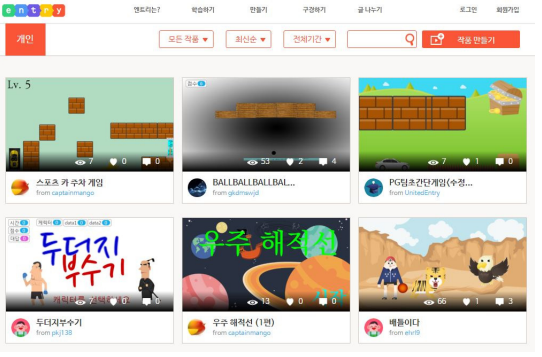
자료 : 엔트리 스크린샷

14) http://www.ebs.co.kr/main

○ **레파지토리 측면의 Play-entry**

[구경하기]메뉴에서는 엔트리를 통해 제작한 작품을 다른 사람들과 공유할 수 있고, 다른 사람과 **협업하여 공동 창작**도 가능한 기능을 제공하고 있다.

[그림 2-22] 엔트리의 개인 갤러리



자료 : 엔트리 스크린샷

○ **평가 측면의 Play-entry**

[학습하기]메뉴 에서 간단한 프로그래밍으로 미션을 해결하는 게임과 소프트웨어의 원리를 설명하는 동영상 등을 제공하여 **단계적 성취**를 할 수 있도록 하고 있다.

[그림 2-23] 엔트리의 학습하기

자료 : 엔트리 스크린샷

□ 비쥬얼 프로그램 사례 분석의 시사점

○ 레파지토리

- 비주얼 프로그래밍 도구의 프로젝트 결과물을 공유하는데 주로 초점을 두고 있다.
- 좋은 결과물들을 사용자가 쉽게 찾아볼 수 있도록, 운영자의 추천이나 기계적 추천방법을 사용 중이다.
- 비쥬얼 프로그래밍 도구에 대한 설명과 활용에 대한 교사용 자료실은 별도로 제공하고 있다.

○ 평가

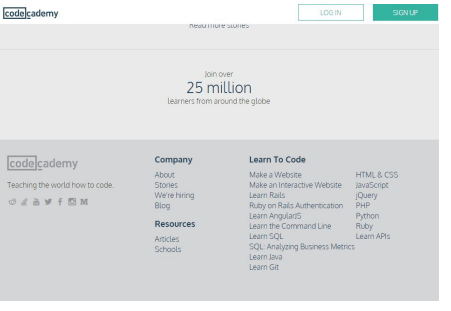
- 비주얼 프로그래밍 도구들은 초등학생이하의 어린이부터 중학교정도의 입문자를 타깃으로 하는 도구가 대부분이다.
- 소프트웨어의 기본 원리를 설명하고 이를 경험적으로 체득하게 하려는 것이 주목적으로 산출물에 대한 객관적 평가를 시도하지 않는 경향이 있다.
- Hour of Code의 문제 기술과 step by step 수행과정 그리고 그에대한 즉각적인 피드백이 유용한 것으로 보인다.
- 스크래치와 엔트리 KODU 등에서 이루어 지는 프로젝트의 공유나 커뮤니티 활동 등을 통한 피어 평가는 매우 중요한 요소로 인식된다.
- 프로젝트에 대한 코멘트가 제작자에게 얼마만큼 긍정적인 피드백으로 작용하는지, 메달이나 배지, 좋아요 등의 랭킹 시스템이 얼마나 해당 커뮤니티를 통해 알 수 있다.

2. 텍스트 프로그래밍

□ Codecademy (https://www.codecademy.com/)

무료로 프로그래밍 입문 교육을 제공하는 온라인 교육기관으로 교수자가 별도 동영상을 통해 가르쳐주기보다 직접 실습하면서 코딩을 배우는 Learn by doing 방식 활용하고 있는 것이 특징이다.

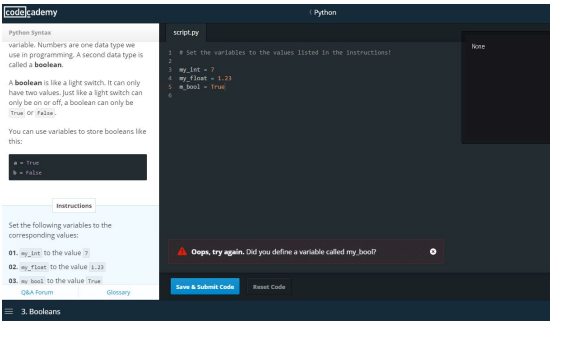
[그림 2-24] 코드아카데미의 홈



자료 : 코드아카데미 스크린샷

학습화면은 왼쪽창에는 설명과 명령, 가운데는 직접 코드를 입력할 수 있는 에디터가 있고 실행 버튼을 누르면 오른쪽에서 결과물을 확인할 수 있도록 되어있으며, 단계마다 자세한 설명과 힌트가 주어지며 에디터에 수강생이 코드를 잘못 입력하면 오류 메시지로 알려준다.

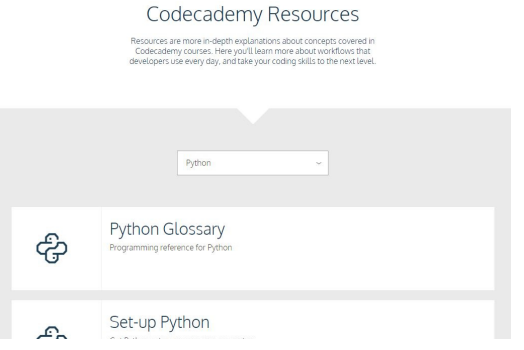
[그림 2-25] 코드아카데미 파이썬 학습화면



자료 : 코드아카데미 스크린샷

중간에 수강을 멈추더라도 이전에 입력된 답안 등 전체적인 히스토리가 저장이 되며, HTML과 CSS 파이썬, PHP, j쿼리, 자바스크립트, 루비 등 다양한 입문용 강좌와 심화강좌가 제공된다.

[그림 2-26] 코드아카데미 심화 학습 자료

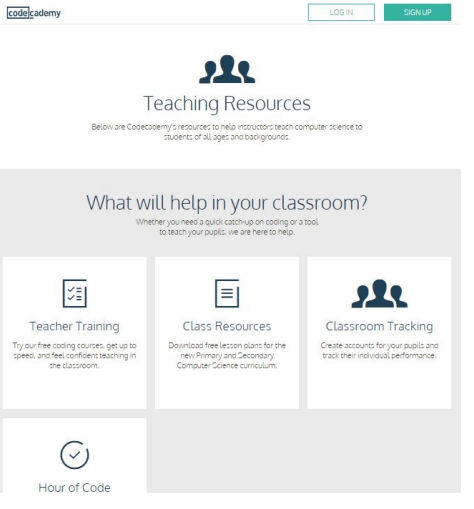


자료 : 코드아카데미 스크린샷

○ **레파지토리 측면의 CodeAcacdemy**

학생들에게 컴퓨터 과학을 가르치는 **교사들을 위해 리소스**를 제공하고 있다.

[그림 2-27] 코드아카데미 교사자료



자료 : 코드아카데미 스크린샷

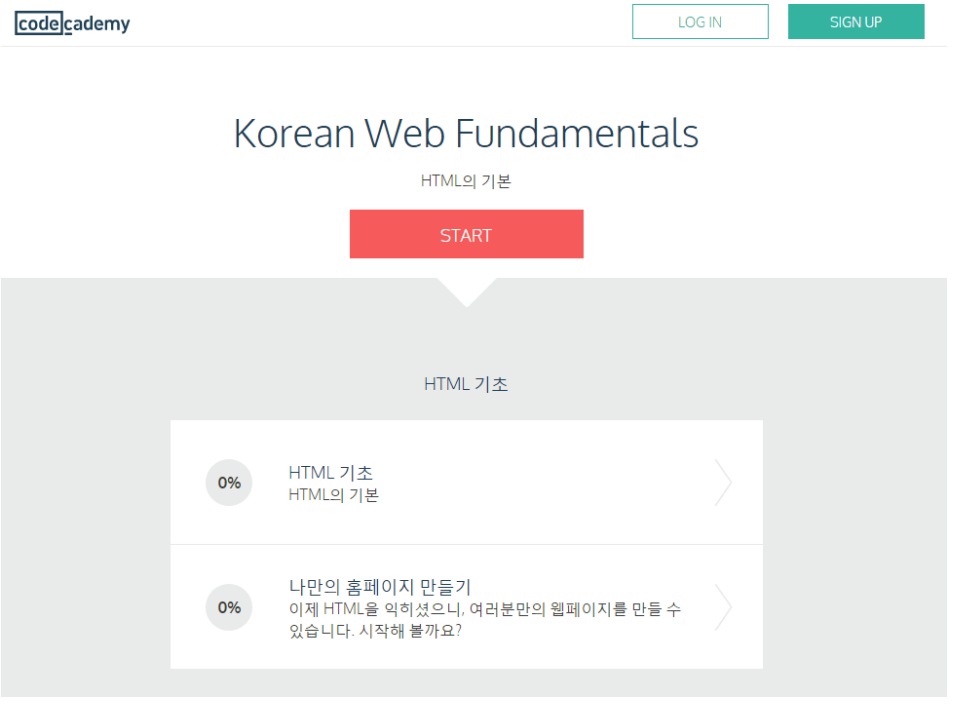
○ **평가 측면의 CodeAcacdemy**

지시를 정확하게 수행하는가에 대해 **즉각적인 피드백**이 입문교육의 평가 방법으로 적당하며, **진도관리**를 통해 학습자가 학습 동기를 가지도록 하는 기능이 우수한 것이 특징이다. 또한, 일부의 서비스를 한국어로 제공중 (google에서 'codecademy 한글'로 검색) 이다.

https://www.codecademy.com/en/tracks/korean-web

https://www.codecademy.com/ko/tracks/python-ko 등

[그림 2-28] 코드아카데미 웹의 기초 과정

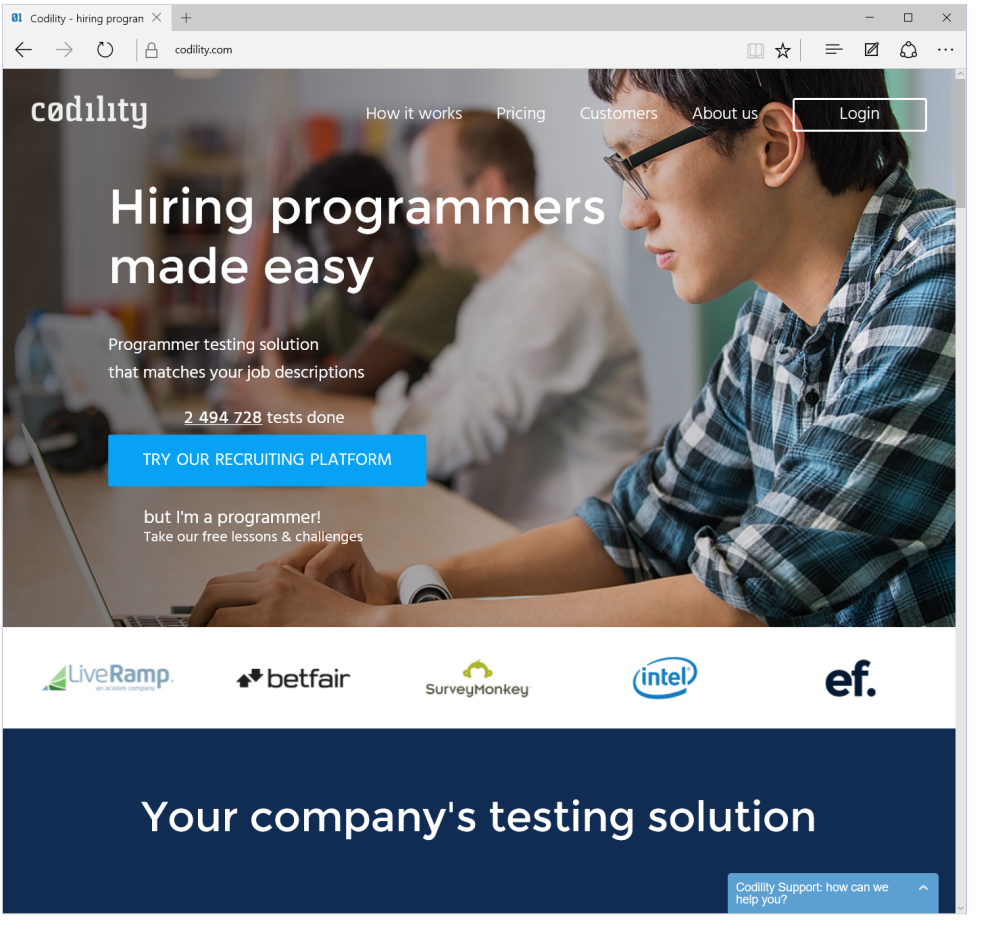


자료 : 코드아카데미 스크린샷

□ Codility (https://codility.com/)

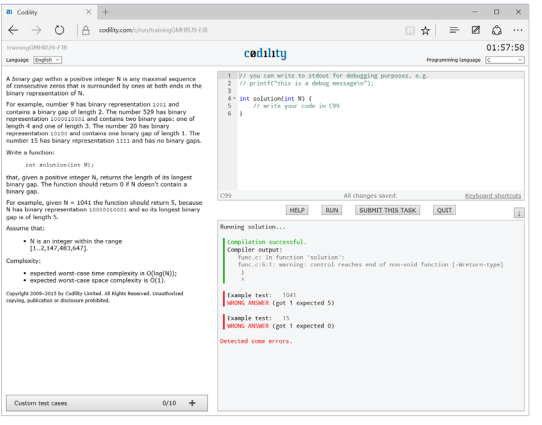
Codility.com는 초·중급자를 대상으로 다양한 프로그래밍 언어(C, C++, C#, Go, Java, JavaScript, Lua, Objective-C, PHP, Pascal, Perl, Python, Ruby, Scala, VB.NET)에 대한 코딩 테스트와 알고리즘을 공부하고 문제풀이를 할 수 있는 프로그래밍 교육서비스를 제공하며 평가 결과를 통하여 구인과 구직에 도움을 주는 서비스이다.

[그림 2-29] 코딜리티 메인화면



자료 : 코딜리티스크린샷

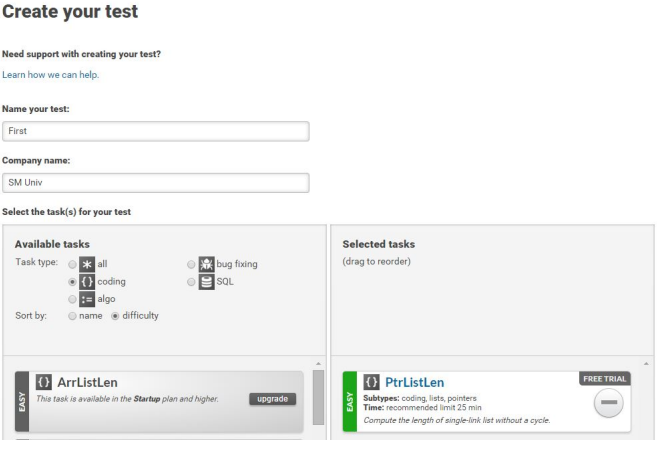
[그림 2-30] 코딜리티 프로그래밍 교육화면



자료 : 코딜리티 스크린샷

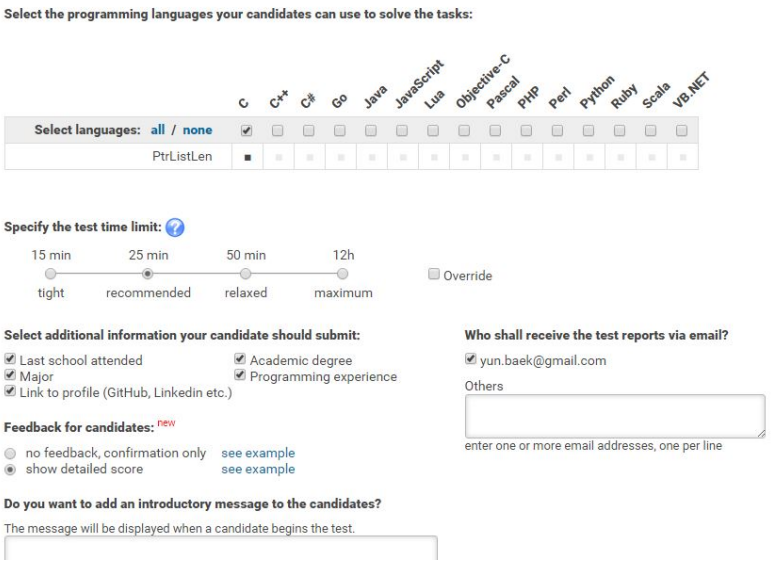
코딜리티는 문제를 내는 이가 교사보다는 채용담당자의 기준으로 되어있다. 채용 담당자가 프로그램 테스트를 만들고, 각종 구비서류 등의 조건을 설정한 후, 후보자들에게 메일을 보내 이 테스트의 수행에 초대를 하면 그 결과를 바탕으로 인터뷰 대상자를 선택하여 테스트를 수행한다.

[그림 2-31] 코딜리티 테스트 생성화면



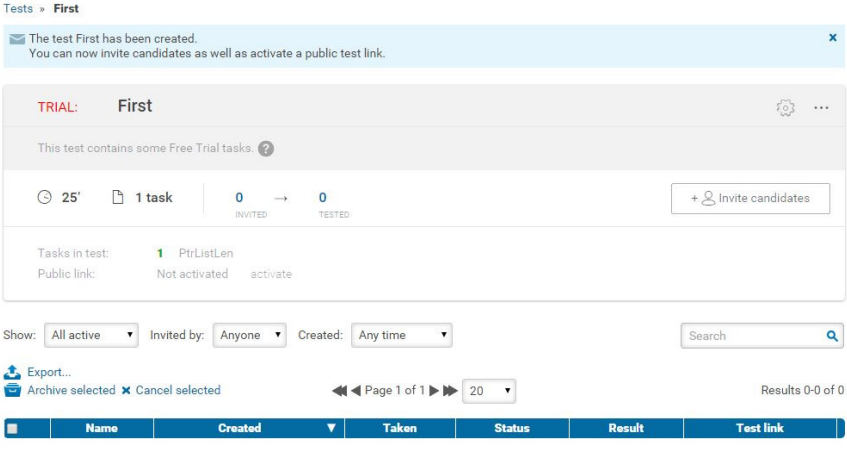
자료 : 코딜리티 스크린샷

[그림 2-32] 코딜리티 테스트 조건 설정



자료 : 코딜리티 스크린샷

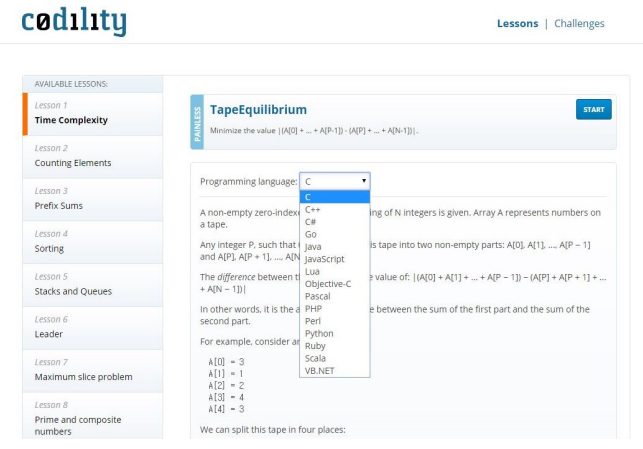
[그림 2-33] 코딜리티 채용 후보자 초대



자료 : 코딜리티 스크린샷

테스트 요청 메일을 받은 구직자에게는 본 테스트를 응시하기 전에 궁금한 부분을 해결하도록 FAQ를 제공하는데, 어떤 형태로 코드를 작성해야 하는지를 평가와 시험의 과정 등을 자세하게 설명하고, 본 테스트 이전에 모의 테스트를 제공한다.

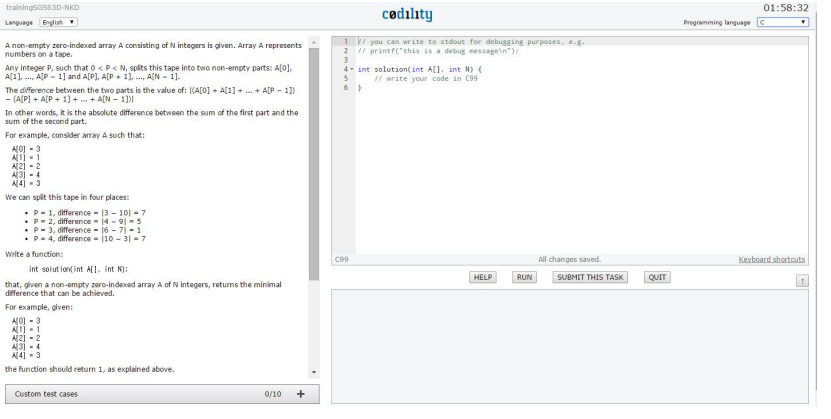
[그림 2-34] 코딜리티 Lesson



자료 : 코딜리티 스크린샷

또한 한편으로는 취업을 원하는 프로그래머들이 학습할 수 있는 환경을 제공한다.

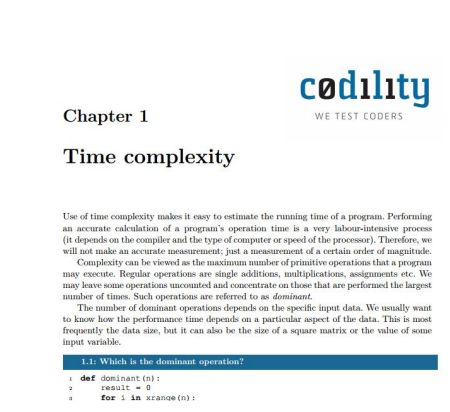
[그림 2-35] 코딜리티 Workspace



자료 : 코딜리티 스크린샷

Web에서 코드를 작성하고 컴파일 결과를 확인하며 코드를 제출하게 되면 시간 복잡도와 공간 복잡도를 서버에서 채점하고 등급을 알려주어 코드의 품질을 측정해주는 것도 특징이다.

[그림 2-36] 코딜리티 복잡도 PDF



codility
WE TEST CODERS

Chapter 1

Time complexity

Use of time complexity makes it easy to estimate the running time of a program. Performing an accurate calculation of a program's operation time is a very labour-intensive process (it depends on the compiler and the type of computer or speed of the processor). Therefore, we will not make an accurate measurement; just a measurement of a certain order of magnitude.

Complexity can be viewed as the maximum number of primitive operations that a program may execute. Regular operations are single additions, multiplications, assignments etc. We may leave some operations uncounted and concentrate on those that are performed the largest number of times. Such operations are referred to as *dominant*.

The number of dominant operations depends on the specific input data. We usually want to know how the performance time depends on a particular aspect of the data. This is most frequently the data size, but it can also be the size of a square matrix or the value of some input variable.

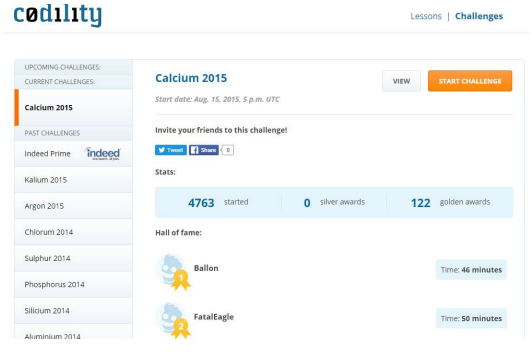
1.1: Which is the dominant operation?

```
1 def dominant(n):
2     result = 0
3     for i in xrange(n):
```

자료 : 코딜리티 스크린샷

프로그래머로서 이해하고 있어야 하는 개념들을 터득할 수 있는 교제와 충분한 예제를 함께 학습하도록 하고 있으며, 수시로 Challenge가 열리며 결과에 따라 Hall of fame에 게시되어 경쟁심을 자극하도록 되어있다.

[그림 2-37] 코딜리티 Challenge Calcum 2015



codility
Lessons | Challenges

UPCOMING CHALLENGES:
CURRENT CHALLENGES:
Calcium 2015
PAST CHALLENGES
Indeed Prime
Kalium 2015
Argon 2015
Chlorum 2014
Sulphur 2014
Phosphorus 2014
Silicium 2014
Aluminium 2014

Calcium 2015
VIEW
START CHALLENGE
Start date: Aug. 15, 2015, 5 p.m. UTC
Invite your friends to this challenge!
Tweet
Share 0
Stats:
4763 started
0 silver awards
122 golden awards
Hall of fame:
Ballon
Time: 46 minutes
FatalEagle
Time: 50 minutes

자료 : 코딜리티 스크린샷

○ **레파지토리 측면의 코딜리티**

코딜리티는 평가를 목적으로 만들어진 서비스로서 별도의 레파지토리 기능은 제공하지 않는다.

○ **평가 측면의 코딜리티**

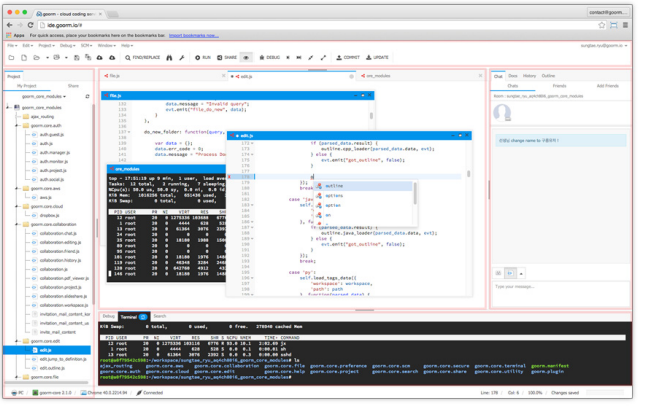
온라인 SW 교육의 개인 능력 평가에 있어서는 갖추어야 할 도구와 시스템을 모두 가지고 있는 서비스로 자동평가라는 측면에 최적화되어 있다고 할수 있으나, 코드의 재사용, 협업, 프로젝트라는 측면은 결여 되어 있다는 점이 단점이다.

□ 코다임(http://codigm.com/)

코다임(codigm, http://codigm.com/)은 언제 어디서나 SW를 배우고, 개발할 수 있는 클라우드 코딩 서비스인 구름(goorm)을 개발하고 서비스하고 있다[4].

클라우드 코딩 서비스 구름은 구름 EDU를 이용하여 초등학교/중학교/고등학교/대학교를 비롯한 다양한 교육 기관에서 프로그래밍 언어를 제공하고 있는데 특히 구름은 실습용 컴퓨터의 성능 노후화 및 한 대의 컴퓨터를 여러 명이 사용하여 유지 관리가 어려운 환경, 그리고 컴퓨터마다 파편화된 실습용 개발 도구의 버전 등은 일관된 프로그래밍 교육이 어려운 환경에서 SW 교육에 유용하다[5].

[그림 2-38] 코다임 클라우드 코딩 서비스 구름(goorm)



자료 : 코다임 스크린샷

구름 EDU를 활용하면, 온라인에서 자동으로 코딩 문제를 출제하고 채점을 해줄 뿐만 아니라, 수업 시간 이외에도 학교·학원·집 구분 없이 언제 어디서나 항상 일관된 교육 환경을 제공한다. 강의자 도구를 지원하여 오프라인 수업에서도 활용 가능하며, 실제 교과서와 연동되어 온·오프라인에서 연속성 있는 코딩 교육이 가능하다. 구름의 특징은 다음과 같다.

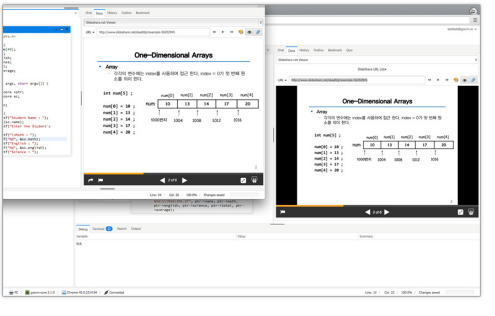
- 손쉬운 강의 교재 제작 가능 (기존 강의안 활용, PDF, 슬라이드 쉐어 연동, 신규 제작을 위한 기능 제공)
- 다양한 SW 교육 환경 제공(C/C++, Java, Python 등 총 9종 언어)
- 손쉬운 학생 관리를 위한 교수자 모드 제공 (학생 등록, 성적 관리, ..)
- 프로그래밍 시험 서비스 및 자동채점 기능 제공
- 소스코드 유사도 검사 기능 제공

구름EDU는 대학 등의 오프라인강좌에서 벌어지는 실습 강의 상황 시나리오에 특화된 서비스이나, 개인 학습자가 특정한 프로그래밍 언어 또는 프로그래밍 개념을 공부하고자 할 경우에는 도움이 되는 콘텐츠는 아직 부족한 편으로 지속적으로 보강중에 있다.

○ 레파지토리 측면의 코다임

코다임(codigm, http://codigm.com/)은 교육 자료 공유서비스는 다른 개발자들과 팀을 조직하고 협업할 수 있는 기능을 제공하여 **슬라이드/PDF 공유, 채팅 기능, 소스 코드의 실시간 동시 편집**이 가능하다.

[그림 2-39] 구름(goorm)의 교육 자료 공유 서비스

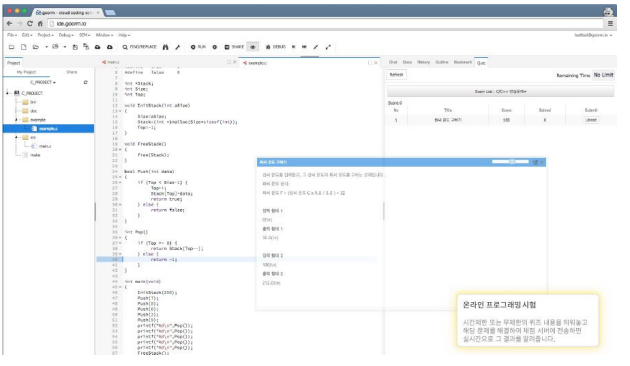


자료 : 코다임 스크린샷

○ **평가 측면의 코다임**

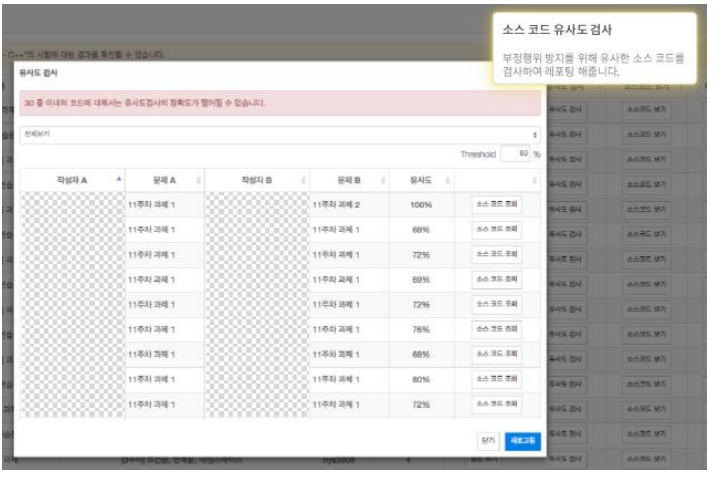
구름은 시간과 장소에 구애 받지 않고 웹브라우저와 인터넷 연결만으로 프로그래밍 학습을 진행할 수 있으며, 프로그래밍 테스트를 **자동 채점**할 수 있으며, **유사도 평가**를 통해 베끼기 여부를 알 수 있다.

[그림 2-40] 구름(goorm)의 자동채점



자료 : 코다임 스크린샷

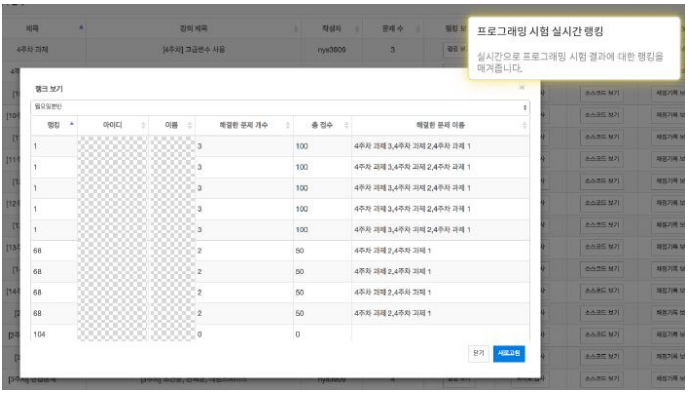
[그림 2-41] 구름(goorm)의 소스코드 유사도 검사



자료 : 코다임 스크린샷

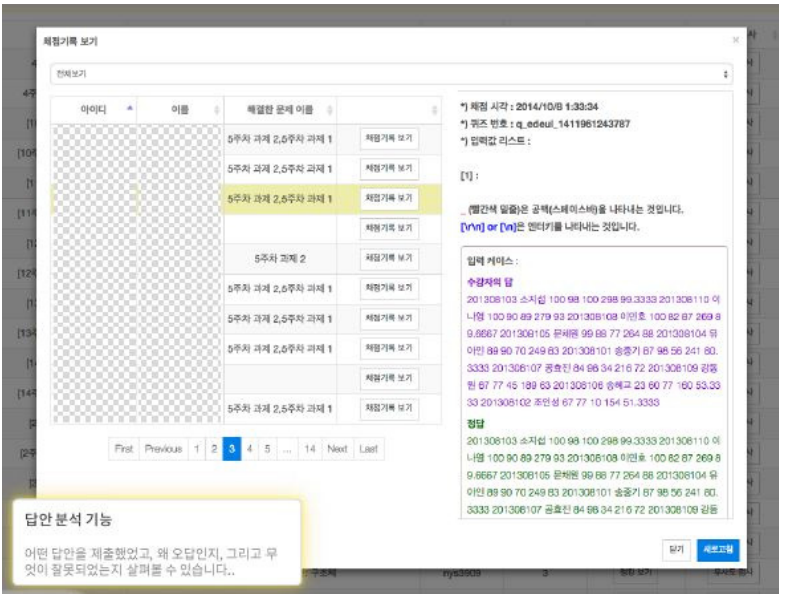
또한 평가 결과의 **랭킹**과 **분석결과**를 제공 하여여 학습평가 기반의 지속적 학습 동기를 제공한다.

[그림 2-42] 구름(goorm)의 실시간 랭킹



자료 : 코다임 스크린샷

[그림 2-43] 구름(goorm)의 답안 분석 기능



자료 : 코다임 스크린샷

□ Project Euler (https://projecteuler.net/)

○ **평가 측면의 프로젝트 오일러**

수학문제들을 프로그래밍으로 해결하는 퀴즈 풀이 사이트 https://projecteuler.net/로서 http://euler.synap.co.kr/ 에서 한글번역 판을 서비스 중이다. 특징은 자신이 원하는 언어로 문제를 풀고 답 입력하면 채점하여 결과를 제공한다. 소프트웨어 교육 측면에서는 수학이나 다른 과목과의 **융합 교육**에 참고하면 좋다.

[그림 2-44] 프로젝트 오일러

Project Euler @ kr

Problems (현재 113 문항)

요약문을 클릭하시면 자세한 내용을 볼 수 있고, 로그인하면 답안을 제출할 수 있습니다.

번호	문제 요약	푼 사람
1	1000보다 작은 자연수 중에서 3 또는 5의 배수를 모두 더하면?	3356
2	피보나치 수열에서 4백만 이하이면서 짝수인 항의 합	2625
3	가장 큰 소인수 구하기	1882
4	세자리 수를 곱해 만들 수 있는 가장 큰 대칭수	1589
5	1 ~ 20 사이의 어떤 수로도 나누어 떨어지는 가장 작은 수	1574
6	1부터 100까지 "제곱의 합"과 "합의 제곱"의 차는?	1568
7	10001번째의 소수	1361
8	1000자리 숫자 안에서 이어지는 5자리 숫자의 곱 중 최대값은?	1166
9	$a + b + c = 1000$ 이 되는 피타고라스 수	1199
10	이백만 이하 소수의 합	984
11	20×20 격자에서 연속된 네 숫자의 곱 중 최대값	765

자료 : 프로젝트 오일러 스크린샷

□ 알고 스팟(Algospot) (https://algospot.com/)

○ **평가 측면의 알고 스팟**

다양한 알고리즘 대회의 문제를 토대로 프로그래밍을 연습하는 사이트로서 다양한 언어로 문제 풀이가 가능하며 문제별 채점결과(속도, 크기) 랭킹을 제공한다. 게시판, 질문과 답변, 튜토리얼을 지원하고 있어 코딩 테스트 준비에 유용하다.

[그림 2－45] Alospot

자료 : Algospot 스크린샷

□ 더블릿(Dovelet) (www.dovelet.com/)

○ 평가 측면의 더블릿

회원이 되려면 소정의 금액을 지불해야 하는 유료 알고리즘 트레이닝 사이트

C, Java, Python, C++, Perl로 문제 풀이가 가능하며 1~30계단, 옥상으로 단계를 구분해서 운영하며 QnA를 제공한다.

[그림 2－46] Dovelt

자료 :Dovelt 스크린샷

□ 깃허브 (http://github.com)

○ **레파지토리 측면의 깃허브**

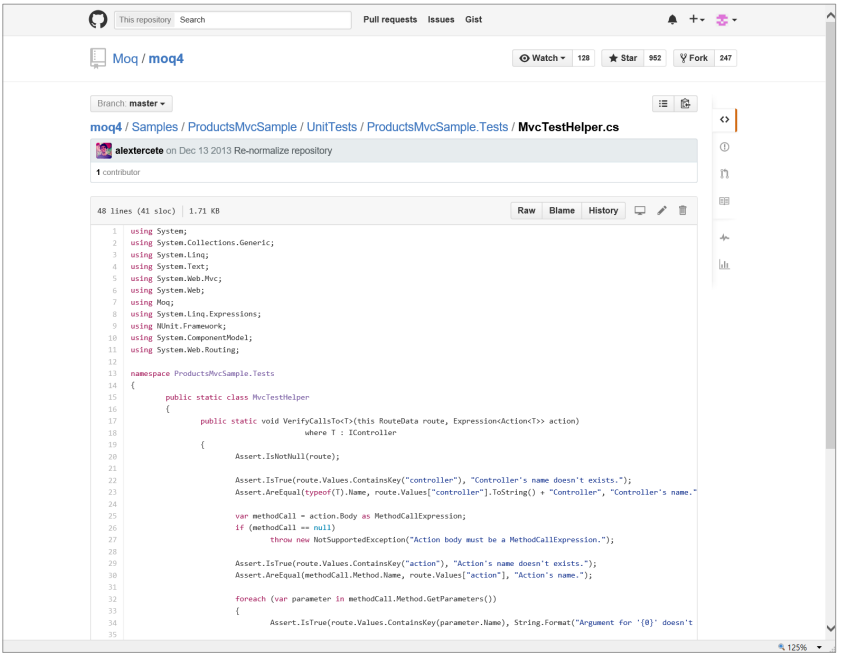
텍스트 프로그래밍 레포지토지로 대표적인 예는 GitHub(https://github.com/)라고 할 수 있다. GitHub는 git을 사용하는 프로젝트를 지원하는 웹 기반의 호스팅 서비스로서, GitHub에서는 다음과 같은 중요한 요소를 가진다.

- 버전 관리
- 소스 코드 저장
- 협업(Collaboration)

○ **Github를 통한 협업 및 개선**

GitHub에서 개발자는 아래 그림과 같이 프로젝트를 만들고, 웹 페이지를 통해 열람, 수정 등이 가능하다.

[그림 2-47] 깃허브의 소스 코드

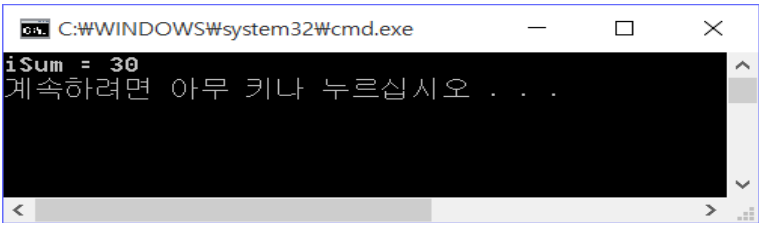


자료 :깃허브 스크린샷

위 그림 에서 fork 부분의 숫자 247은 fork 횟수이다. fork는 프로젝트 복사 회수를 의미한다. 개발자가 fork 하면 개발자의 계정에 현재 작성된 버전이 그대로 복사가 되고, 수정한 버전을 생성하여 계속 사용할 수도 있다. 또한, [Pull request] 메뉴를 클릭하여 rails 관리자에게 수정한 부분을 반영해 달라고 요청할 수 있다. [Watch] 메뉴는 프로젝트의 수정을 지속적으로 관찰하고 수정이 발생하면 notification을 받을 수 있다.

GitHub를 SW 교육에 활용한 예를 들어 보자. 그림 3.23과 같이 1~10사이의 2의 배수 합을 구하는 프로그램을 구현하는 예를 들어보자.

[그림 2-48] 1~10 사이의 2의 배수 합 실행 결과



자료 :직접 작성

위 그림과 같은 결과를 얻기 위해 한 학생이 아래 그림과 같이 소스코드를 작성하였다.

[그림 2-49] 1~10 사이의 2의 배수 합을 구현한 프로그래밍 소스 코드

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApplication1
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             int iSum = 0;
14             for(int i=0; i <= 10; i++)
15             {
16                 if (i % 2 == 0) iSum += i;
17             }
18             Console.WriteLine("iSum = {0}", iSum);
19         }
20     }
21 }
```

자료 :직접 작성

다른 학생이 위의 그림의 소스코드 14~17 라인을 아래의 그림과 같이 최적화 할 수 있다.

[그림 2-50] 최적화된 1~10사이의 2의 배수 합을 구현한 프로그램 소스 코드

```
14              for(int i=0; i <= 10; i+=2)
15              {
16                  iSum += i;
17              }
```

자료 :직접 작성

위 그림과 같이 최적화 한다면 for문의 실행 횟수를 1/2로 줄일 수 있고, if (i % 2 == 0)문을 제거하여 프로그램의 실행 속도를 높일 수 있게 된다.

□ 텍스트 프로그램 사례 분석의 시사점

○ 레파지토리

- 레파지토리에 올려져 있는 프로젝트를 직접 코딩하거나 수정할 수 있는 기능을 제공하는 것이 학습에 도움이 된다.
- 텍스트 프로그램의 레파지토리는 협업과 버전관리를 가능한 많은 기능들이 구현된 사례가 많았다.
- 그러나 학습을 목적으로 만들어진 것이 아니었기 때문에, 학습 자료나 학습자와 교사의 의견등을 담을 공간에 대한 고려는 많지 않아 이를 고려할 필요가 있다.

○ 평가

- 텍스트 프로그램은 사용자 층에 따라 난이도가 크게 차이가 났다.
- 수행 결과에 따른 결과를 즉각적으로 피드백하고, 평가 결과에 대한 분석을 통해 지속적으로 학습할 수 있도록 연결해주는 것이 중요하다.

3. 피지컬 컴퓨팅

□ ODIY(Opensource DIY) (http://opensource.kofac.re.kr/index.do)

오픈 소스기반의 초소형 컴퓨터와 이를 활용하는 방법에 대해 쉽게 배울 수 있는 온라인 학습사이트로 과학창의재단이 운영하고 있다.

[그림 2-51] ODIY의 목적

자료 : ODIY 사이트 스크린샷

커뮤니티 활동 보다는 커리큘럼 위주로 운영되고 있으며, 커리큘럼은 아두이노에 대한 동영상 강의가 대부분을 차지하고 있다.

[그림 2-52] ODIY의 자료현황

전체(77건)

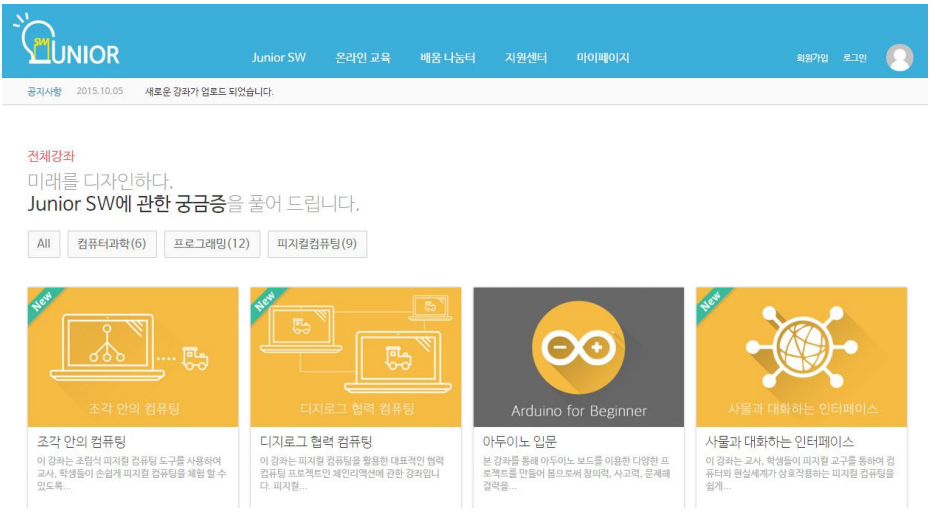
아두이노 (50건)	라즈베리파이 (6건)	비글보드 (4건)	기타 (12건)	프로젝트공유 (5건)
초급 (15건) 중급 (17건) 고급 (18건)	초급 (6건) 중급 (0건) 고급 (0건)	초급 (4건) 중급 (0건) 고급 (0건)	초급 (12건) 중급 (0건) 고급 (0건)	아두이노 (1건) 라즈베리파이 (0건) 비글보드 (1건) 기타 (3건)

자료 : ODIY 사이트 스크린샷

□ 주니어SW (https://koreasw.org)

초·중등생 대상 온라인 소프트웨어 교육센터로 창의컴퓨팅(Creative Computing) 커뮤니티를 표방하며 정보통신산업진흥원(NIPA)과 미래인재연구소(InFIT)가 공동 운영하고 있으며, 온라인 교육의 한 카테고리로 피지컬 컴퓨팅을 운영 중에 있으며, 동영상 강좌를 제공한다.

[그림 2-53] 주니어 SW

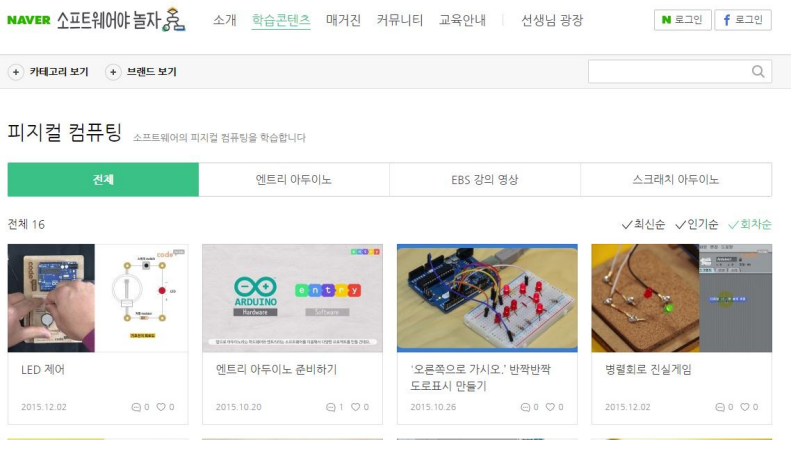


자료 : 주니어SW 스크린샷

□ NAVER 소프트웨어야 놀자 (http://playsw.naver.com/)

초· 중등 학생을 위한 대표적인 온라인 SW 교육 사이트인 소프트웨어야 놀자 (http://playsw.naver.com/main)는 컴퓨팅사고력, 스크래치수업, 아두이노수업 등의 소프트웨어 관련 교재와 엔트리를 활용한 실습 가능 동영상 강의와 SW 관련 된 ebs•BBC동영상들을 제공하고 있으며, 선생님들을 위한 교육지도서의 자료를 제공하여 수업시간에 활용이 가능하며, .학습콘텐츠의 카테고리 중 피지컬 컴퓨팅을 제공하고 있다.

[그림 2-54] 네이버 소프트웨어야 놀자 중 피지컬컴퓨팅



자료 : 네이버 소프트웨어야 놀자

□ 피지컬 컴퓨팅 사례 분석의 시사점

○ 레파지토리

- 피지컬 컴퓨팅은 HW와 SW가 융합된 형태로 HW를 제어하는 소스 코드가 동반되어야 한다. 기본적으로 텍스트 프로그래밍의 레파지토리에 HW의 동작을 확인할 수 있는 동영상이 필요하며, 프로젝트를 통해 만들어진 코드를 포함한 산출물들을 저장할 공간이 필요하다.

※ Hamster 사례

[그림 2-55] Hamster와 USB to BLE Bridge 동글

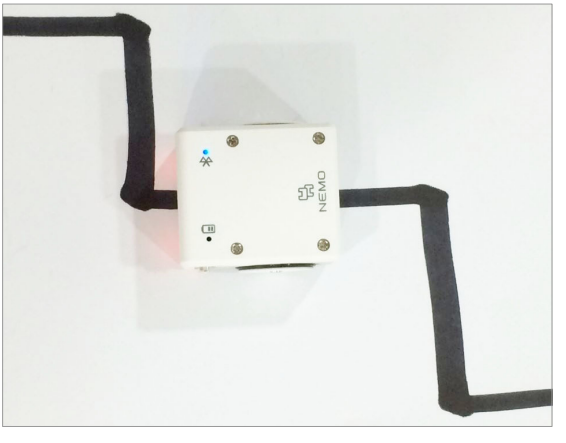
자료 : 구글 이미지 검색

〈표 2-2〉 Hamster 사양

항목	특 징
통신방식	블루투스 4.0 BLE(접속 범위 15m 이내)
배터리	충전 30분 연속 동작 평균 1시간 대기 최대 12시간
충전단자	micro USB
PC 연결	USB 동글을 통한 시리얼 통신
스마트 폰 연결	블루투스 4.0 BLE 연결
센서	전방 거리 센서(근접 센서) 바닥 센서 / 3축 가속도 센서 조도 센서 / 온도 센서 / 배터리 전압 센서

USB to BLE Bridge 동글은 블루투스 4.0 BLE 패킷을 시리얼 통신 패킷으로 변환해 주는 장치이다. PC와 Hamster를 블루투스로 연결할 때는 페어링 과정을 통한다. 페어링 과정은 쉽고 빠르며 여러 대의 Hamster전원이 켜져 있어도 자신의 Hamster만을 찾아 연결한다.

[그림 2-50] Hamster 주행 제어



자료 : 주니어SW 스크린샷

아래의 Hamster 주행을 제어하는 SW 프로그래밍은 파이썬, 스크래치, 엔트리, 자바스크립트, 자바, C/C++, 등의 다양한 프로그래밍 언어로 구현할 수 있다. 아래는 Hamster 주행을 제어하는 파이썬 프로그래밍 언어로 구현한 예이다

```
# author: Kwang-Hyun Park (akaii@kw.ac.kr)

# simple movement
from roboid.hamster import *

robot = Hamster()

# move forward
robot.write(Hamster.LEFT_WHEEL, 50)
robot.write(Hamster.RIGHT_WHEEL, 50)
wait(500)

# move backward
robot.write(Hamster.LEFT_WHEEL, -50)
robot.write(Hamster.RIGHT_WHEEL, -50)
wait(500)

# stop
robot.write(Hamster.LEFT_WHEEL, 0)
robot.write(Hamster.RIGHT_WHEEL, 0)

# don't forget to dispose
disposeAll()
```

○ **평가**

- 피지컬 컴퓨팅의 경우에는 프로젝트를 수행한 과정과 결과물에 대한 동영상과 설계, 구현에 대한 문서로 평가할 수 있지만 이를 자동화하는 것은 어렵다.
- 프로젝트에 대한 각종 산출물을 바탕으로 커뮤니티의 투표와 코멘트를 중요 피드백으로 사용할 수 있으나 현재까지 활성화 되고 있지 않아서 이를 활성화 하기 위한 운영방안이 필요하다.

제 2 절 사례 분석을 통해 본 시사점

1. 레파지토리 모델의 사례 분석을 통해 본 시사점

□ 레파지토리에 저장되야 하는 결과물

레파지토리에 필요한 자료들은 교사용의 교육자료와 학습자가 학습을 통해 만드는 산출물로 나눌 수 가 있다. 교육자료의 경우, 콘텐츠의 유형에 따라 크게 달라지지는 않으며 문서, 동영상의 다양한 교육 리소스가 해당된다.

그러나, 학습자가 만드는 산출물의 경우에는 콘텐츠의 유형에 따라 달라진다.

〈표 2-3〉 레파지토리에 저장되야 하는 요소들

콘텐츠 유형	교사 교육자료	학습자 산출물
비쥬얼 프로그래밍	동영상, 문서	프로젝트
텍스트 프로그래밍		소스코드 + 프로젝트
피지컬 컴퓨팅		소스코드 + 동영상

□ 댓글과 추천, 큐레이션

일반적인 레파지토리와 달리, 교육을 위한 레파지토리에는 교사와 친구들의 반응이 매우 중요하다. 이러한 상호평가의 요소를 강화하기 위해서는 댓글과 추천과 같은 기능과 함께 양질의 프로젝트를 노출하는 큐레이션이 중요하다.

〈표 2-4〉 큐레이션의 종류

추천자	결과물
시스템	시간순, 재사용빈도, 추천순, 카테고리
관리자	운영자, 교사 추천

□ 최종 시사점

SW교육을 위한 레파지토리는 교사와 학생들을 위한 별도의 공간이 필요하며, 프로젝트의 활동을 기록하는 것이 중요하다, 따라서 최종 결과물만 저장하는 형태가 아닌 형상관

리와 협업을 지원하는 Github의 모델이 필요하며, 학습자들에게 적합한 프로젝트와 자료들을 적절히 추천하고 노출하는 큐레이션과 운영이 필요하다.

2. 평가 모델의 사례 분석을 통해 본 시사점

□ 즉각적이거나 단계적인 자동평가

온라인상의 평가는 특정 시점의 평가가 아닌 즉각적인 피드백 형태로의 자동평가가 이루어진다. 이는 학습자의 수준을 정하기 보다는 학습자 스스로의 문제점을 파악하여 지속적으로 개선하기 위함이다. 특히 초중고의 초급자의 경우, 제시되는 명령을 얼마나 잘 따라 가는가에 초점을 두고, 학습 의욕을 북돋우어주는 것이 바람직하다.

또한 이러한 즉각적인 평가는 특정 단계별 미션을 수행하는 형태로 이루어져 갑작스러운 난이도 상승으로 인한 학습에 대한 흥미를 잃지 않도록 해야 한다.

□ 유사도 평가

유사도 평가는 학생들의 베끼기를 확인하기 위함에 있다. 그러나, 문제의 성격에 따라 비슷한 코드가 나올 수 있는 가능성은 높다. 따라서 절대적인 수치보다는 교사의 업무를 줄여주는 보조적인 수단으로 사용되야 하며, 교사들도 이를 알고 있어야 한다.

□ 상호평가

프로젝트의 댓글이나 추천도 평가의 요소에 들어갈 수 있다. 이는 학습자의 적극도를 측정하고, 프로젝트의 정성적인 품질을 측정할 수 있다. 또한 학습자의 동기강화에도 효과적이다. 단순히 자기 수준을 측정하는 것을 넘어, 상호 대결을 통한 자극을 줄 수 도 있다.

□ 재사용 평가

재사용은 프로젝트의 품질을 평가하고, 소프트웨어가 가진 속성을 잘 이해하는 것이라고 할 수 있어 이를 평가의 요소로 사용할 수 있다.

□ 레파지토리 기반의 활동

온라인 상의 평가는 프로젝트나 미션들을 수행하는 사용자의 행동에 기반한 데이터로 기반하여 평가를 할 수 있다. 이 때 레파지토리는 이러한 데이터를 기록하는데 이용 될 수 있다.

□ 상급자를 위한 콘텐츠, 현장에서 만날 수 있는 실제적인 문제

초중고 생이라고 해도 온라인 학습을 통해서 충분히 고급 프로그래머 수준에 이를 수 있다. 교사들이 이를 모두 가르쳐주기에는 사실상 어렵다. 때문에 상급자를 위한 고급 프로그래밍 기술에 대한 콘텐츠와 평가에 대한 부분이 필요하다.

- 요구되는 출력만 만족하는 프로그램이 아니고 강력한 조건들을 달성하는 정도를 수치화 할 필요가 있다.

□ 최종 시사점

온라인 SW교육에서 모든 종류의 사용자에게 수행 결과에 대한 즉각적인 피드백은 필수적이다. 이를 위해 웹기반의 온라인 수행 환경이 갖춰져야 하고 자동 수행 평가가 이루어지는 것이 바람직하다.

초급자를 위해서는 Codecademy처럼 단계적인 학습을, Codility서비스와 같은 객관적인 능력 평가에 있어서는 갖추어야 할 도구와 시스템이 필요하며, 구름EDU처럼 코드의 재사용, 협업, 프로젝트라를 수행할 필요가 있겠다. 피지컬 컴퓨팅 프로젝트의 경우, 대한 각종 산출물을 바탕으로 커뮤니티의 투표와 코멘트를 중요 피드백으로 사용할 수 있고, 이를 활성화 하는 것이 중요하다.

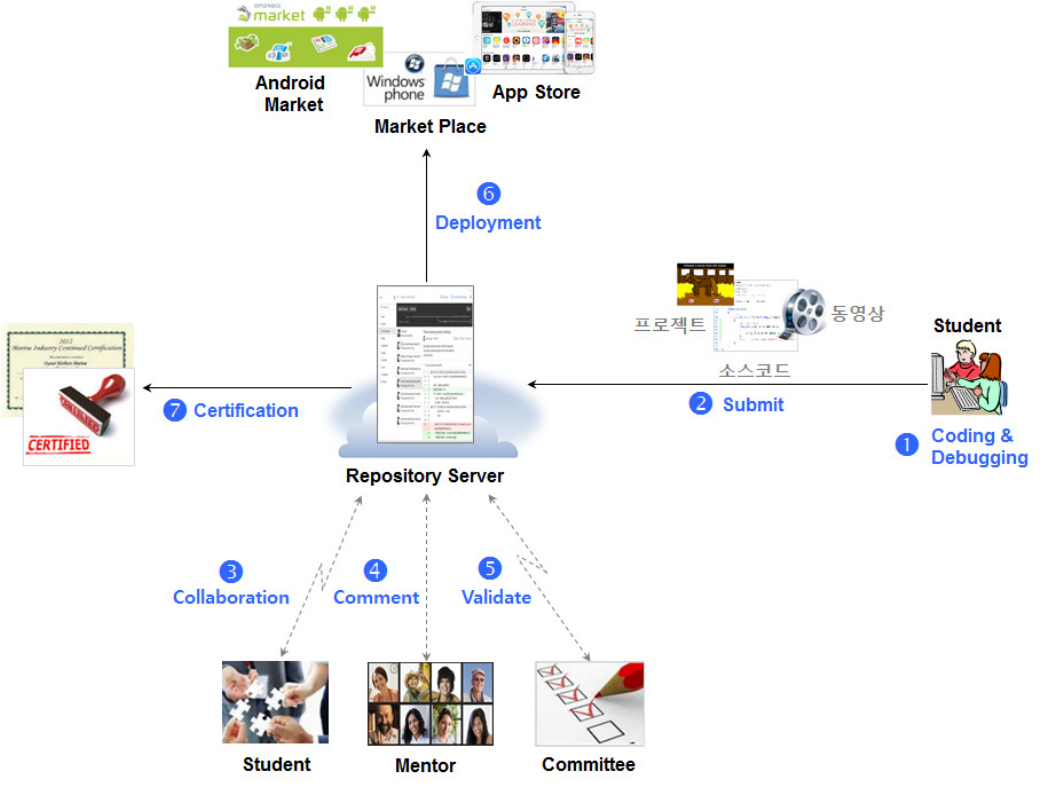
제3장 효과적인 초중고 SW온라인 교육체계

제1절 초중고 SW 온라인 교육을 위한 레파지토리 모델

앞서 이야기한 시사점에 기반한 효과적인 초중고 SW온라인 교육체계에 대한 모델을 제안한다. 그 중 먼저 가장 기반이 될수 있는 레파지토리는 아래와 같다.

1. 레파지토리 모델과 구성 요소

[그림 3-1] 레파지토리 모델의 구성요소



자료 : 직접작성

□ 레파지토리 서버

레파지토리 서버는 SW교육에 대한 자료들과 학습 활동들과 프로젝트가 저장되고 기록되는 곳이다. 이를 중심으로 모든 교육에 대한 데이터들이 누적되고 공유될 수 있다.

□ 학생과 동료 학생

학생은 교육을 온라인 SW교육 서비스를 받고, 프로젝트를 수행한다. 이때 레파지토리 서버를 기반으로 프로젝트 수행을 동료 학생들과 함께 하거나, 서로의 프로젝트들에 대해 피드백하고 재사용할 수 있다.

□ 멘토, 교사

멘토나 교사는 프로젝트를 코치하거나 개선방향에 대해 조언을 줄 수 있다. 학생들의 프로젝트를 고치거나 댓글을 다는 것이 가능하다.

□ 전문가 위원

수업내에서 만든 프로젝트가 온라인SW교육 서비스나 마켓플레이스와 같은 공개된 곳에 공개될때의 검수를 수행한다. 이때 개인정보 문제, 보안의 위협이나, 윤리적 문제에 대한 부분을 확인할 수 있다.

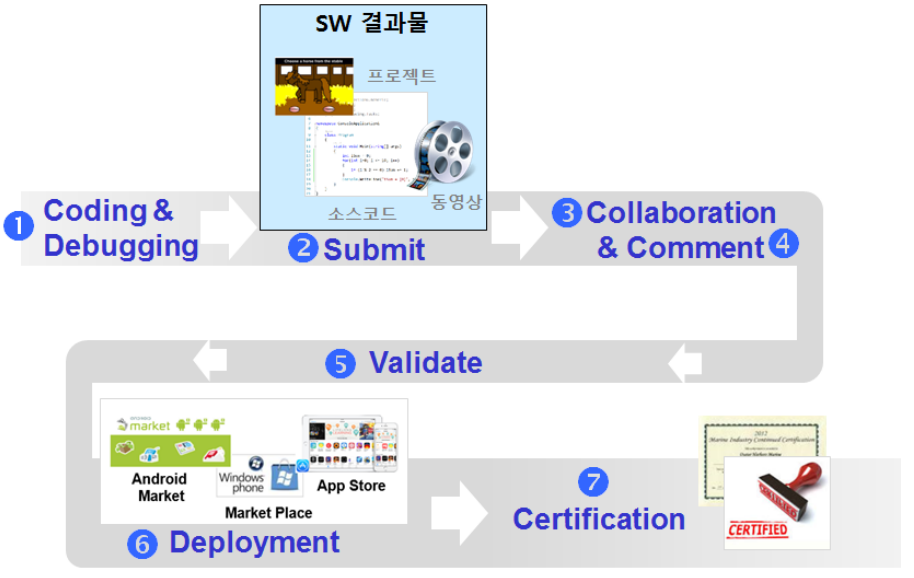
□ 마켓플레이스, 갤러리

온라인 SW교육을 통해 만들어진 결과물을 공개된 갤러리나 구글플레이와 같은 외부 마켓플레이스로 공개할 수 있게 한다.

2. 레파지토리 모델의 프로세스

레파지토리에 기반한 프로젝트 수행 및 배포는 아래와 같은 과정을 겪을 수 있다.

[그림 3-2] 레파지토리 모델의 배포과정



자료 : 직접작성

☐ Coding & Debugging

학생들이 소스를 코딩하고, 디버깅을 반복하는 과정이다. 앞에 설명한 클라우드 코딩 서비스 구름(goorm)과 같이 웹에서 직접 코딩을 하고 피드백할 수 있는 환경 제공이 필요하다.

☐ Submit

Coding & Debugging 과정의 결과물인 소스코드, 프로젝트, 동영상 등을 레파지토리 서버에 업로드하는 과정이다. 이 과정에서는 스크래치, 엔트리 등과 같이비쥬얼 프로그래밍 결과물은 스크래치, code.org의 레파지토리와 같은 기능 제공이 필요하다. 하지만 Java, Python 등과 같은 텍스트 프로그래밍 결과물은 클라우드 코딩 서비스 구름(goorm)과 codility.com의 소스코드 테스트와 같은 환경 제공이 필요하다. 또한, 로봇, 아두이노 등과

같은 피지컬 컴퓨팅 결과물은 스크래치, code.org의 레파지토리와 같은 기능과 클라우드 코딩 및 소스코드 테스트와 같은 환경이 융합된 기능을 제공하는 환경이 필요하다.

□ Collaboration

Collaboration 과정은 레파지토리 서버에 업로드 된 소스코드, 프로젝트, 동영상에 대하여 동료나 선•후배들이 참여하여 협업을 하거나, 버전 업그레이드 하는 과정이다. 이 과정은 다른 개발자들과 팀을 조직하고 슬라이드/PDF 공유, 채팅 기능 뿐만 아니라 소스 코드의 실시간 동시 편집의 협업 기능이 필요한 과정이다.

□ Review

레파지토리 서버에 업로드 된 소스코드, 프로젝트, 동영상에 대하여 교사 또는 멘토들이 Review하는 과정이다. 이를 통해 개선안을 내거나 추가적인 코치가 가능하다.

□ Validate

레파지토리 서버에 업로드 된 소스코드, 프로젝트, 동영상에 대하여 전문가 위원들이 검증하는 과정이다. 이 과정에서는 Apple의 AppStore, Microsoft의 Windows Store와 같이 등록한 소프트웨어에 대하여 전문가들이 검증하도록 한다.

□ Deployment

검증된 소스코드, 프로젝트, 동영상을 배포하는 과정이다. 레파지토리에 등록된 많은 소스코드, 프로젝트, 동영상을 검증하여 우수한 품질의 소스코드, 프로젝트, 동영상을 다른 개발자들이 사용할 수 있도록 공개 배포한다.

□ Certification

평가를 통하여 인증을 수여하는 과정이다. 이 과정에서는 Codelity.com에서 제공하는 코딩 테스트와 알고리즘 테스트와 같은 과정을 통과하고, Udacity.com에서 제공하는 Nanodegree 프로그램과 같은 교육 과정을 이수하면 인증을 해줄 수 있으며, 만들어진 프로젝트를 외부 공개 마켓플레이스에 배포할 수 있다.

3. 효과적인 초중고 SW온라인 교육체계를 위한 레파지토리 모델

□ 사용자 그룹별로 권한이 차등된 레파지토리 모델

- 레파지토리 공간에 대한 접근은 교육 공급자이냐 수요자이냐에 따라서 권한과 책임, 그리고 공유 가능한 자료의 범위가 달라야 한다.

□ 콘텐츠 유형 중심이 아닌 프로세스와 중심 모델

- 온라인 SW교육의 모든활동들을 포괄할 수 있는 레파지토리 구축을 위해서는 콘텐츠의 형태에 무관한, 프로세스과 중심의 레파지토리가 필요하다.

□ 최종 산출물 저장공간이 아닌 형상관리 도구

- 최종 결과물에 대한 저장공간이 아닌 해당 결과물이 나오기까지의 산출물과 사용자들의 활동이 기록된 형상관리 도구이어야 한다. 교육은 산출물이 목적이 아니라 교육 과정에서 얻고자 하는 교육활동과 효과가 목적이기 때문이다.

□ 커뮤니티 기반의 확산 모델

- 레파지토리를 이용하는 사용자들이 함께 모여 개선하고, 의견을 나누고 지속적으로 자료를 공유함으로서 발전하는 커뮤니티 형태의 모델을 지향해야 한다. 초중고 교육은 학교를 중심으로한 교육이 이루어지므로 커뮤니티의 역할은 매우 중요하다고 할 수 있겠다.

제 2 절 초중고 SW 온라인 교육을 위한 평가 모델

효과적인 초중고 SW온라인 교육을 위한 평가모델을 생각해보기 위해서는 먼저 SW교육에 어ㄸ ㄴ 것을 평가할 것인가부터 살펴보자.

1. 무엇을 평가 할 것인가?

□ SW 교육의 목적과 평가 프레임워크

SW교육의 목적은 현대사회의 중심 동력인 Computing 환경에 대한 적극적인 이해와 활용을 목적으로 한다. 인간의 생각을 추상화하고 이를 기계에 구체적으로 구현 - Computational Thinking(CT)의 평가 프레임워크는 아래와 같다.

〈표 3-1〉 Computational Thinking 평가 Framework[15)]

요소	설명
Computational concepts	프로그래밍의 기본 요소인 시퀀스, 루프, 조건 등의 개념을 이해함
Computational Practices	기본요소를 이용하여 SW를 구현하며 테스팅, 디버깅, 재사용, 추상화 등을 포함
Computational Perspectives	프로젝트 분석. 디자인, 인터뷰 등을 포함

□ 개정된 한국의 정보교과의 학습 내용

2015년 개정된 한국의 중학교 교육과정 중 정보교과의 내용체계에서 코딩과 밀접하게 연관되는 것은 아래 표와 같다.

15) 최형신, Computational Thinking 역량 계발을 위한 수업설계 및 평가 루부릭 개발, 한국정보교육학회논문지 제18권 제1호 pp.57-64, 2014년 3월

〈표 3-2〉 중학교 교육과정 정보 교과 내용 체계 중 일부

영역	핵심 개념	일반화된 지식	내용요소	기능
문제 해결과 프로그래밍	추상화	추상화는 문제를 이해하고 분석하여 문제 해결을 위해 불필요한 요소를 제거하거나 작은 문제로 나누는 과정이다.	•문제 이해 •핵심요소추출	비교하기 분석하기 핵심요소추출하기 표현하기 프로그래밍하기 구현하기 협력하기
	알고리즘	알고리즘은 문제 해결을 위한 효율적인 방법과 절차이다.	•알고리즘이해 •알고리즘표현추출	
	프로그래밍	프로그래밍은 문제의 해결책을 프로그래밍 언어로 구현하여 자동화하는 과정이다.	•입력과 출력 •변수와 연산 •제어 구조 •프로그래밍 응용	

이에 반해 미국과 영국은 초등학교 수준부터 우리보다는 적극적인 SW 교육 커리큘럼을 가지고 있다.[16)]

〈표 3-3〉 영국의 컴퓨팅 커리큘럼

구분	학년	교육내용
KS1	1-2	. 알고리듬과 프로그램의 이해 . 간단한 프로그램의 작성과 시험
KS2	3-6	. 목적에 맞는 프로그램 설계와 작성 . 문제를 작게 쪼개어 해결(디버깅) . 순차, 선택, 반복, 변수, 입출력 활용 . 알고리듬과 프로그램의 에러 검출과 수정
KS3	7-9	. 실세계 문제의 상태와 행위를 모델링 . 정렬과 검색, 프로시저 활용 . 2가지 이상의 프로그래밍 언어 사용 . 부울(AND, OR, NOT) 논리 이해

16) 정영식, 소프트웨어 제작 분야의 성취 목표, 교수학습 방법 및 평가 방법에 관한 연구, 한국정보교육학회논문지 제18권 제1호 pp.185-193, 2014년 3월

〈표 3-4〉 미국의 컴퓨터 과학 커리큘럼

구분	문제해결과 알고리듬	컴퓨터 프로그래밍
K-2	. 알고리듬의 정의 . 간단한 작업 과정 쓰기	. 컴퓨터는 프로그램에 의해 작동됨을 이해 . 컴퓨터 프로그램, 프로그래머, 프로그래밍 언어 개념
3-5	. 성공적인 작업을 위한 알고리듬 작성과 실행 . 논리연산자 AND OR 결과	. 프로그래밍 언어의 목적 . 간단한 순차 프로그램 작성 및 시험
6-8	. 컴퓨터의 기능은 명령을 실행한 것임을 이해 . 주어진 문제를 해결하기 위해 알고리듬 작성 . 효과적인 알고리듬 선택	. 알고리듬을 프로그램으로 작성 . 주어진 문제를 해결하기 위해 변수와 반복문 이용하여 프로그램 작성과 시험
9-10	. 문제해결 절차 이해 . 표준 설계도구 활용 . 주어진 알고리듬이 성공적인 해결여부판단	. 특정 사양에 맞는 프로그램의 작성과 시험, 실행 . 문제를 프로그램으로 작성 . 적절한 자료형 선택 . 구조화된 프로그램 작성

SW 교육 평가는 커리큘럼의 각 교육 영역에 대해 성취기준이 마련되고 기준에 대한 달성여부, 달성 정도를 파악하는 것이 평가의 중요 요소라고 할 수 있다.

〈표 3-5〉 중등과정 성취기준 예시

영역	성취 기준
다양한 알고리듬	. 다양한 데이터 정렬 방법을 이해하고 그 차이를 설명할 수 있다 . 다양한 데이터 검색 방법을 이해하고 그 차이를 설명할 수 있다 . 데이터 정렬 방법을 순서도로 표시하고 최적의 방법을 찾을 수 있다 . 데이터 검색 방법을 순서도로 표시하고 최적의 방법을 찾을 수 있다 . 순서도를 보고 오류를 찾거나 다음 동작을 예측할 수 있다 . 효율적인 방법으로 자료를 정렬과 검색하려는 자세를 갖는다
데이터 구조 생성	. 다양한 형태의 데이터 구조와 자료형을 이해할 수 있다 . 다차원 배열을 이용한 프로그램을 작성할 수 있다 . 간단한 데이터 구조를 설계하고 그것을 프로그램으로 작성할 수 있다 . 간단한 데이터 구조를 이용하여 특정 값을 입출력 할 수 있다 . 필요한 자료를 추상화 하고 구조화하려는 자세를 갖는다
소프트웨어 제작 관리	. 프로그램 속의 에러를 수정하기위해 논리적인 추론을 사용할 수 있다 . 소프트웨어의 개발 공정을 이해하고, 단계별로 필요한 문서를 작성할 수 있다 . 사용자 입력 값에 따라 결과가 달라지는 프로그램을 작성할 수 있다 . 디버깅 툴을 이용하여 프로그램을 수정할 수 있다 . 기계어의 의미를 알고 간단한 프로그램을 작성할 수 있다 . 자신이 만든 프로그램을 비판적으로 개선하려는 자세를 갖는다

2. 무엇으로 평가를 할 것인가?

□ 산출물 기준의 평가

교육활동에서 생산되는 각종 자료는 평가의 기준으로 활용 될 수 있다. 앞서 이야기한 Computational Thinking의 평가 Framework별로 평가가 가능한 산출물은 아래와 같으며, 수준별로 성취영역을 구분하고, 산출물을 기반으로 평가를 진행 할 수 있다.

〈표 3-6〉 Comutational thinking 평가 Framework 요소별 산출물

성취 영역	평가 목적	산출물 (자동화 가능성)
Computational concepts	(개념학습)Compuatianl Thinking의 개념의 이해정도를 평가	• 객관식 (높음) • 주관식 테스트 (보통) • 에세이 (낮음)
Computational Practices	(프로그래밍 도구 활용) 프로젝트에 필요한 프로그래밍 도구의 학습 성취도 평가	• 실습 수행시간 (높음) • 실습 수행 중 행동 (보통)
	(코딩) 실제 프로그램을 한 결과를 기준으로 한 평가	• 소스코드 (보통) • 실행결과 (높음) • 시간 복잡도 (높음) • 공간 복잡도 (높음) • 읽기가능정도 (보통)
Computational Perspectives	(분석, 설계 능력) SW를 설계하고 분석해낼 수 있는 능력에 대한 평가	• 구술설명 (낮음) • 문서작성 (낮음)
	(협업 능력) 팀단위로 SW를 만드는데 필요한 협업 능력 평가	• 스펙기술서 (낮음) • 스토리보드 (낮음) • 역할분담 (낮음) • 간트 차트 (보통)

그러나 산출물은 객관적인 수치화가 가능한 부분은 자동화가 가능하나, 분석,설계, 협업 능력 같은 높은 수준의 측정을 위해서는 자동화의 도움을 받기 어려우며, 주관적인 평가가 많이 개입되는 산출물들의 경우에는 점수기반의 수준 측정 보다는 Pass, Fail정도의 가벼운 평가가 중요하며, 평가보다는 가이드와 코치가 더욱 중요하다. 자동화가 어려운 부분은 산출물 기준 보다는 평가자의 자질이 더욱 더 중요하다.

□ 평가자별 분류

이러한 경우, 산출물 보다도 더욱 더 중요한 것은 평가 주체에 의존하는 주관적인 평가이다. 평가자들의 역할은 학생들의 점수를 메겨, 줄을 세우기보다는 스스로 학습을 지속적으로 할 수 있도록 도움을 주는 취지의 접근이 중요하다.

교사, 전문가, 학습자 그룹이 될 수 있으며 자동평가가 가능한 부분은 별도로 구분해 보았다.

〈표 3-7〉 평가주체별 평가방법

평가주체	평가 방법
교사	• 주관식 테스트, 에세이 • 실습 중 태도 코멘트 • 프로그램 분석 결과 평가 • 프로젝트 산출물 평가
전문가 멘토	• 교사의 역할을 온라인상에서 batch 혹은 interactive로 분담 가능
학습자간 피어평가	• 개념학습 토론 • 프로그램 수행 • 코드의 읽기 가능 정도 • 프로그램 분석 토론 • 프로젝트 산출물 평가
시스템 (자동평가)	• 완전 자동 평가 - 객관식 테스트 (문제은행 방식) - 도구를 이용한 실습 완성 시간 (온라인 도구의 사용자 트래킹) - 코드 실행 결과 (웹기반의 실습 환경) • 부분 자동 평가 - 도구 사용자의 행동 - 코드 퀄리티 ※ 실행시간, 변수 사용, 분기나 반복의 사용, 코딩 컨벤션, 블록의 사용

가중 주요한 평가자는 교사가 될 수 있다. 그러나, 국내 초중고 학교의 교사의 수는 매우 부족한 상황으로, 온라인을 통한 전문가 멘토 풀을 확보하여 이를 보조하고, 시간과 공간의 문제를 해결하는 것을 추천한다. 또한 객관성의 증가를 위해 1명의 평가자가 아닌 다수의 크로스 평가가 이루어짐이 좋다.

학습자간 피어평가는 친구들간의 평가를 통해 동기를 강화하고, 협업등의 능력을 증가하는데 도움이 될 수 있다.

자동평가의 경우에는 가장 기본적인 것은 객관적으로 측정가능한 수치는 완전자동평가가 가능하며, 주관적이거나 비정형적인 데이터에 기반하는 경우, 사용자의 행동이나 평가자들의 피어 평가를 기반으로 하여 반자동화를 해야 한다.

3. 효과적인 초중고 SW온라인 교육체계를 위한 평가 모델

□ 비교적 간단하고 효과적인 코드 평가

○ 중요한 것은 코드(code)

- 온라인 SW 교육에 있어 가장 중요한 산출물은 '코드'가 되야 한다.
- '코드'가 주어진 문제를 해결하는지를 즉답적인 평가가 진행되어야 하며, 이를 위한 활동들을 추적평가해야 한다.

○ '코드' 를 위해서는

- Hour of code나 Codecademy에서 인상적으로 보았던 친절하고도 순차적인 지침과 그에 대한 즉각적인 피드백은 초급자들에게 꼭 필요하다할 수 있다.
- 지침을 따라가는 것을 통해 '코드'가 만들어 지고
- '코드'를 만드는 사람과 이해하는 사람이 늘어나기 위해서는 기본적으로 많은 초급자용 콘텐츠가 필요하다.

□ 복잡하나 달성해야하는 프로젝트 평가

○ '코드' 보다 한단계 더 발전한 평가 수단은 프로젝트라고 할 수 있다.

- 주어진 문제를 푸는 연습을 쌓아, 스스로 문제를 찾아 해결하는 것을 프로젝트라고 할 수 있으며, 여기에는 분석, 설계, 피드백등의 코딩외의 추가적인 활동들이 포함된다.
- 또한 팀단위의 협업에 대한 측정도 이를 통해 가능하다.

○ 협업은 온라인 SW교육의 경우, 동시접속의 어려움이 존재하므로 레파지토리와 커뮤니티 등을 활용하여 비동기적인 활동에 대한 추적과 평가가 필요하다.

○ 코드평가에 대비하여 자동화가 어려우며, 상대적으로 교사의 능력만으로 프로젝트를 평가하기에는 쉽지 않다. 따라서 피어평가의 활용을 극대화 할 수 있는 온라인 시스템 설계와 멘토 등의 외부평가에 대한 방안도 구상되어야 한다.

□ 결론

비주얼 프로그래밍을 막 벗어난 혹은 그런 경험도 없는 초보자들이 순차적인 지침과

친절한 피드백을 가지는 환경에서 '코드'를 작성하고 그것에 즉각적으로 실행되어 문제 해결과 프로젝트 중심의 흥미와 역량을 측정할 수 있는 평가가 가장 시급하게 필요하다고 할 수 있다.

제4장 결론 및 시사점

이상 초중고 온라인 SW교육의 레파지토리와 평가 모델을 살펴보았다. 레파지토리는 온라인 SW교육에서 사용자의 활동과 산출물을 추적 관리하며 커뮤니티로서 소통이 가능한 기반 시스템이 되어야 하며, 평가는 이러한 레파지토리를 기반으로 단계적 이고 즉각적인 피드백을 주는 미션을 통해 입문자의 흥미를 이끌고 성장을 시켜야 하며, 궁극적으로는 온라인 프로젝트 기반으로 문제해결능력과 협업에 대한 성숙도를 스스로 측정하고, 상호 독려할 수 있는 시스템으로 자리잡아야 한다. 이를 통해 온라인 SW교육이 대한민국 초중고학생들의 SW교육을 위한 효과적인 학습도구로 정착될 수 있다.

참 고 문 헌

국내 문헌

소프트웨어정책연구소 (2014),『SW중심사회 - 의미와 대응 방향

미래창조과학부 (2014), 소프트웨어 중심사회 실현 전략

삼성전자(2014,2015), 지속가능경영보고서

미래창조과학부 (2015) SW중심대학 추진 계획

최형신(2014) Computational Thinking 역량 계발을 위한 수업설계 및 평가 루부릭 개발, 한국정보교육학회논문지 제18권 제1호

정영식(2014), 소프트웨어 제작 분야의 성취 목표, 교수학습 방법 및 평가 방법에 관한 연구, 한국정보교육학회논문지 제18권 제1호

생활코딩, 처음만나는 스크래치
https://opentutorials.org/course/1415/6951

OLC주니어, 선생님이 알려주는 스크래치
http://olc.oss.kr/jr_new/list_04_teacher.jsp

소프트웨어야놀자, 소프트웨어야 놀자 소개
http://playsw.naver.com/info/introduction

Play Etry(http://play-entry.org/)

주니어SW (https://koreasw.org)

코다임(http://codigm.com)

해외 문헌

Businessinsder(2015), THE INTERNET OF EVERYTHING:

World Economic Forum (2014) John Chambers

https://agenda.weforum.org/2014/01/are-you-ready-for-the-internet-of-everything“, Code.org

PAT YONGRPRADIT(2013) WHAT IS CODE.UORG UP TO?, Code.org

CLASS CENTRAL(2015), By The Numbers:MOOCS in 2015

CODE.ORG (http://code.org)

SCRATCH (https://scratch.mit.edu/)

KODU GameLab(http://www.kodugamelab.com/)

Project Euler (https://projecteuler.net/

Dovelet (www.dovelet.com/)

GitHub (http://github.com)

부 록

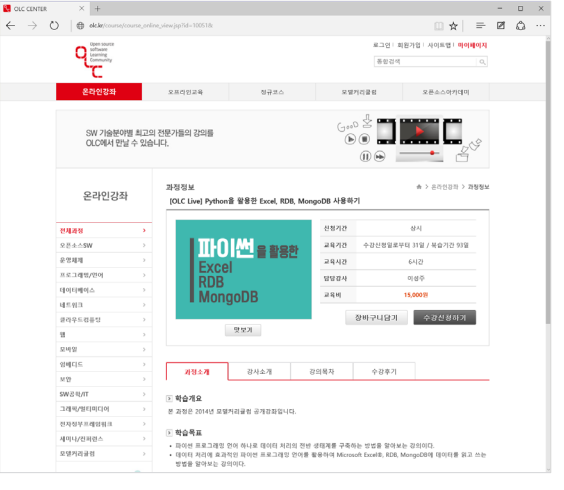
1. OLC 센터

대학생 또는 재직자 대상의 온라인 SW 교육 사이트중의 하나인 OLC(Open Source Software Learning Community Center)센터는 공개SW 전문인력 양성을 목적으로 구 지식경제부가 주최하고, 정보통신산업진흥원(NIPA)과 공개소프트웨어협회(KOSSA)가 공동주관하는 개방형 소프트웨어 교육센터이다[3].

OLC센터는 체계적인 양질의 커리큘럼으로 공개 SW 개발인력 부족 문제를 해소하고 계층별, 기능별 전문 인력 양성 기반을 구축하고 있다. 학생, 현직 개발자, IT관련 종사자 등 공개 SW에 관심 있는 누구나 참여가 가능하다.

대학생 또는 재직자 대상의 온라인 SW 교육 사이트중의 하나인 OLC(Open Source Software Learning Community Center)센터는 사용자가 직접 제작한 교육 콘텐츠로 유·무료의 강의실을 개설할 수 있는 온라인 오픈마켓 서비스를 제공하는 레파지토리를 운영한다.

[그림 4-1] OLC센터 온라인 강좌의 예

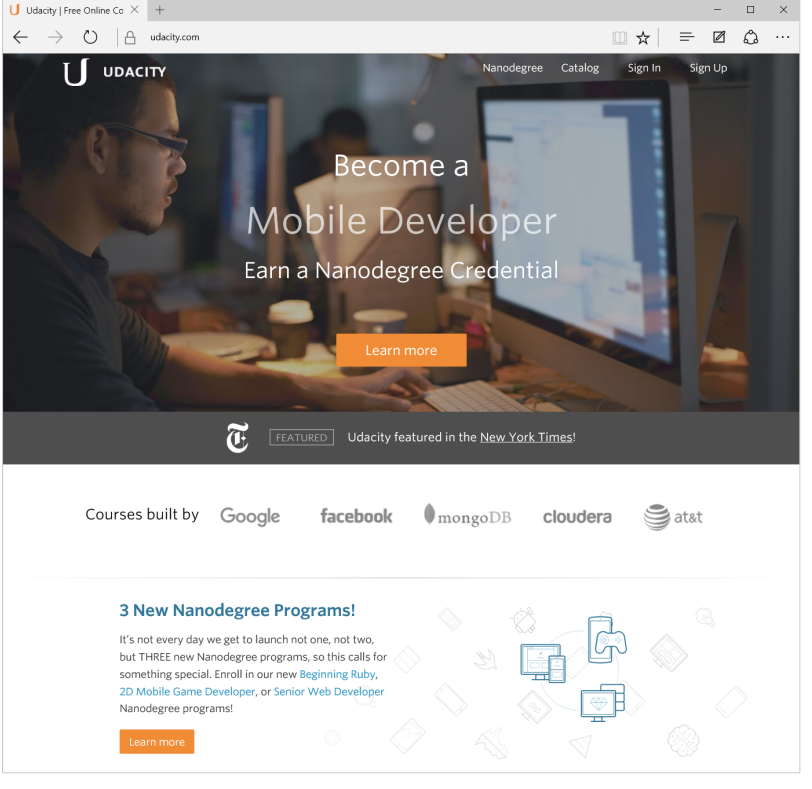


자료 : OLC스크린샷

1. Udacity

2011년 스탠포드대학에서 시작한 그림 3.13의 Udacity.com은 Coursera, 에드엑스와 함께 세계 3대 MOOC 사이트로 알려져 있다. Udacity 설립자 세바스찬 스룬(Sebastian Thrun) 교수가 처음 개설한 컴퓨터공학 강의에 160,000 여명의 학생이 등록한바 있다. Udacity에서는 초급에서 고급까지 다양한 강의를 제공하고 있다. 특히 특정 과목과 프로젝트를 이수하면 나노 학위(Nano degree)를 수여하는 서비스를 제공하고 있다.

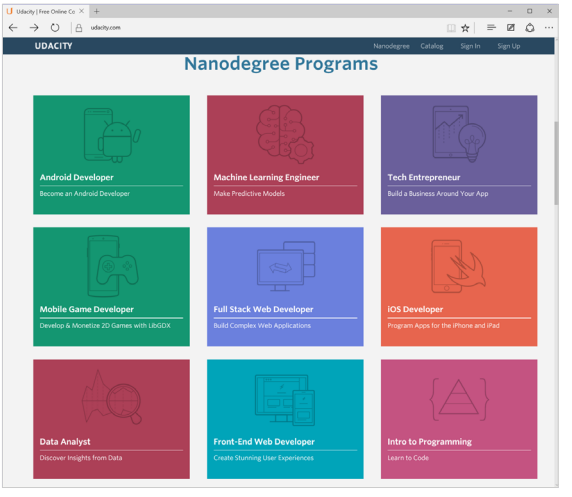
[그림 4-2] Udacity



자료 : Udacity 스크린샷

Udacity.com에서 제공하는 12개의 Nanodegree 프로그램은 그림 [4-3]와 같이 2015년 12월 현재 Android Developer, Full Stack Web Developer, Front-End Web Developer, iOS Developer 등 12개의 Nanodegree 프로그램을 제공하고 있다.

[그림 4-3] Udacity의 Nanodegree 프로그램



자료 : Udacity 스크린샷

2. Code Club[17]의 SW 교육

영국에서 자원 교사들이 모여 방과 후에 9세에서 11세 어린이들에게 코딩을 가르치는 모임에서 시작하여 어린이들이 생각과 배움을 만들고 공유하는 방법을 터득하게 하여 학교에서든 직업으로든 디지털 창작을 쉽게 할 수 있도록 하는 데 목적이 있다.

프로그램을 하는 방법 뿐 아니라 계산적 사고(computational thinking), 문제 해결(problem solving), 계획(planning), 디자인(design) 과 협업(collaboration)을 배우게 하는 것이 특징이며, 자원 교사들이 학교나 도서관 등의 기관에서 Code Club Project를 이용하여 1주 한, 두시간 정도에 하나의 프로젝트를 학습하는 형태를 가진다.

1, 2학기에는 스크래치를 가지고 프로그래밍의 기본을 공부하고 3학기에는 HTML과 CSS를 이용하여 웹개발을 학습하며 4학기에는 Python등의 개발 도구를 학습한다.

17) https://www.codeclub.org.uk/

[그림 4-3] 코드클럽 첫 페이지

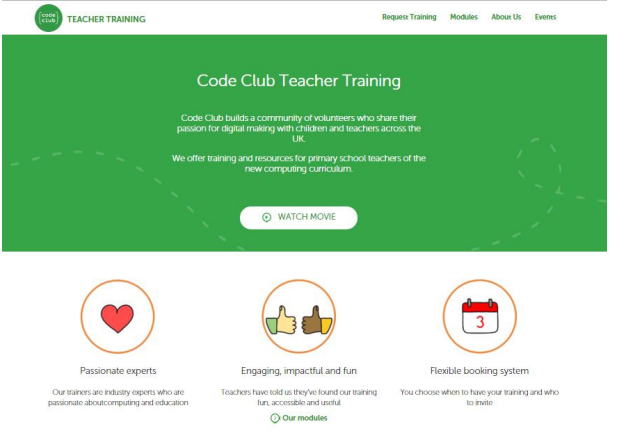
자료 : 코드클럽 스크린샷

[그림 4-4] 코드클럽 코스

자료 : 코드클럽 스크린샷

Code Club은 자원 교사들이 각 클럽을 성공적으로 운영하는데 필요한 정보와 노하우를 확실히 체득하도록 온라인 훈련 서비스를 제공하고 있다.

[그림 4-5] 코드클럽 교사 훈련 서비스



자료 : 코드클럽 스크린샷

영국 전역의 21,000개 초등학교에 Code Club이 설치되는 것을 목표로 함. 현재, 영국내에 4173개의 클럽이 등록되었고 5만여명이 지원하여 이용하며 2015년 11월 Raspberry Pi Foundation의 자회사가 되어 전세계로 Code Club 커뮤니티를 확장하는 기반을 마련하고 있다.

그림 4-6] 영국내 코드클럽 분포

자료 : 코드클럽 스크린샷

저 자 소 개

양 병 석

·숭실대 컴퓨터학부 학사
·현 소프트웨어정책연구소 연구원

이 원 주

·한양대 학사
·한양대 대학원 석사
·한양대 대학원 박사
·현 인하공업전문대학 교수

백 윤 철

·서울대 학사
·서울대 대학원 석사
·서울대 대학원 박사
·현 상명대학교 교수

길 현 영

·고려대 컴퓨터학과 졸업
·고려대 컴퓨터학과 석사
·University of Pennsylvania (미), Master in Computer and Info. Sci.
·Pennsylvania State University (미), Ph.D in Computer Sci. and Engineering
·현 소프트웨어정책연구소 선임연구원

김 윤 명

·전남대 문헌정보학과 졸업
·경희대 국제법무대학원 석사
·경희대 대학원 박사
·현 소프트웨어정책연구소 선임연구원

방송통신정책연구 15-진흥-055

효과적인 초중고 SW 온라인 교육체계연구

2016년 01월 31 일 인쇄
2016년 01월 31 일 발행

발행인 미래창조과학부 장관
발행처 미래창조과학부
경기도 과천시 관문로 47 정부과천청사
Homepage: www.msip.go.kr