2016 미래인재연구소 정기리포트 3호

- Computational Thinking과 교육 -

2016. 4. 1

경인교육대학교 미래인재연구소

들어가며

소프트웨어 중심사회 실현 정책에 따른 SW 교육 강화로 초중등교육에 정식교과로 안착될 날이 이제 얼마 남지 않았습니다. SW교과가 만들어지기 이전에 교과의 기반이 되는 철학과 사고력의 중심에 Computational Thinking이라는 사고력이 있었습니다. 자넷 윙에 의해 촉발된 이 새로운 사고력의 추상적 형태가 이제 그 실체를 발현하기 시작했습니다.

교육부와 미래창조과학부의 적극적인 지원, 기업의 적극적인 지원과 사교육시장의 활성화, 학부모들의 SW교육에 대한 필요성의 인식 개선 그리고 초중등 선생님들의 관심과 교육적용의 다양한 노력들이 보이기 시작한 겁니다.

국외 교육에서 컴퓨팅교육, IT교육, 컴퓨터과학교육, 코딩교육을 사례로 들지 않아도 이제는 우리 주변에서 그리 어렵지 않게 SW 교육의 전략과 새로운 내용의 적용 등에 대한 다양한 사례를 볼 수 있습니다.

그런데 한 가지 고민스러운 것이 바로 CT입니다. SW교육의 핵심 목표가 CT(CT)를 가진 창의융합인재 양성입니다. 하지만 현실의 SW교육 실천현장에서 보이는 모습은 CT구성 요소 중 자동화에 집중하는 경향이 뚜렷합니다. 스크래치나 엔트리를 활용하고 피지컬 교구를 사용해서 학습자들의 흥미를 높이고 코딩을 효과적으로 가르치기 위한 방법에 치우친 듯 보입니다. 우리가 경계해야 하는 것은 누구나 알고 있듯이 프로그래밍, 코딩, 소프트웨어 개발을 잘하는 인재를 양성하는 교육이 아니라는 겁니다.

이렇게 자동화를 위한 코딩 실습 기능에 집중하는 현상은 신생 교과로서 당연하게 나타나는 증상이기도 합니다. CT를 신장시키기 위해 코딩 실습의 역량이 반드시 필요한 것도 사실입니다. 하지만 SW교육의 목표가 사고력 중심, 문제해결력 중심의 창의인재양성을 위해서는 고도의 사고력을 이끄는 추상화 과정에 집중해야

합니다. 이런 추상화 중심 교육의 내용이 사실 학생들에게는 매우 어려운 것은 경험해 보지 않아도 쉽게 이해할 수 있습니다만 가르치는 우리들에게는 매우 중요한 사실입니다.

2014년부터 진행되어 온 SW교육 선도(연구)학교는 2016년에 900여개 학교가 운영되고 2017년에는 2000여개의 학교가 운영된다고 합니다. 첫 단추를 어떻게 끼우는가에 따라 2018년에 시작하는 SW교과의 정상적인 안착이 좌우됩니다.

이러한 점을 고려하여 미래인재연구소의 멤버들은 이번 정기리포트를 Computational Thinking을 주제로 다루고 국외 연구 자료의 소개를 통한 이론적인 고민과 철학 그리고 EPL, 피지컬 컴퓨팅 교육의 실제적인 전략이나 사례를 통해 교육 현장에서 고려해야 할 내용 등을 중심으로 정리하였습니다.

이번 정기리포트를 통해 SW교육이 기기를 다루는 교육, 프로그래밍 기법을 실습하는 교육, SW산업 역군을 길러내는 교육이 아니라 세계를 주도하고 자신의 삶을 보다 풍요롭게 개척해가는 미래인재 양성의 자료가 되길 바랍니다.

2016년 3월 31일
미래인재연구소장
한선관

목 차

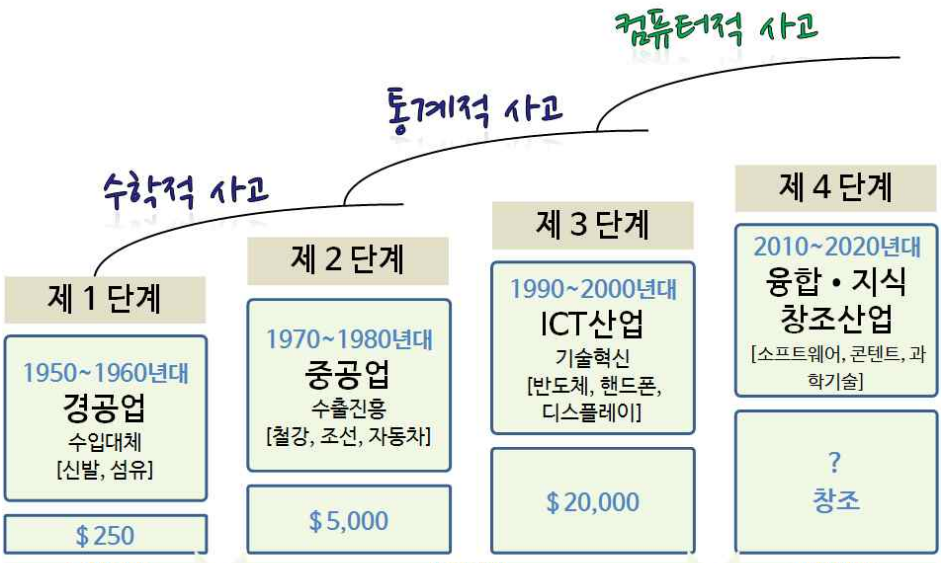
Computational Thinking

알파고와 이세돌의 바둑 대결은 단순히 구글의 인공지능을 시험해보는 것을 넘어 사회에 큰 파장을 일으켰다. 이세돌의 네 번째 대국에서의 승리로 사람들은 아직 인간이 컴퓨터보다 우월하다라는 안도감을 느끼기는 하였지만 여전히 두려움을 느끼고 있다는 것은 부정하지 못할 것이다. 인공지능의 발전으로 유토피아를 꿈꿀 것인지, 아니면 수많은 영화에서처럼 디스토피아가 펼쳐질 것인지는 아직 아무도 장담 못하지만 이것만은 모두가 공감하며 궁금해한다.

이제는 디지털 시대이다! 디지털 시대에서 요구하는 교육은?

농경사회, 산업사회, 정보사회를 거쳐 지금은 지식사회시대이다. 우리 아이들이 살아가야 하는 미래는 스마트 및 유비쿼터스 사회로 융합/유비쿼터스 컴퓨팅을 활용하며 살아야 한다.

현재의 세대는 산업사회에 태어나 산업사회에서 요구되는 교육을 받으며 자랐다. 수학과 과학이 그 대표적인 예이며, 글로벌 시대에 발맞추어 영어까지 공부하느라 여념 없었다. 많은 지식을 습득해야 했으므로 암기가 중요했고, 평가도 그에 맞추어서 이루어졌었다. 미래의 구성원으로서 살아가야 할 우리 아이들은 디지털 사회에서 디지털을 다룰 줄 알아야 하며 이를 바탕으로 생각해야 한다.



이미지 출처 : http://statkclee.github.io/window-of-statistics

세상의 데이터를 디지털화하여 우리에게 유용한 정보로 분석하고 창의적으로 문제를 해결할 수 있는 사고능력, 이러한 문제해결의 사고과정을 전문적으로 발휘하는 컴퓨터과학자의 관점에서 생각하는 것, 이것이 Computational Thinking(CT)를 이해하는 첫걸음이다.

자넷 윙은 CT를 ‘ 문제를 수립하고 해결책을 만들어 컴퓨팅 시스템을 효과적으로 수행되도록 표현하게 하는 사고 과정을 말한다’라고 하였다.

실제 2013 노벨화학상은 다중척도 모델링을 개발한 세 명의 과학자들에게 돌아갔다. 이전까지의 화학자들의 플라스틱 공과 막대를 가지고 화학분자 모델을 분석했으나 이들의 개발한 컴퓨터 모델로 인하여 컴퓨터로 화학작용을 예측하고 이해하게 된 것이다. 과거와 달리 컴퓨터 프로그램을 다루며 문제를 해결하고 있다.

2011년 마크 앤드리슨은 월스트리트 저널에 “Why Software is eating the world?'라는 글을 통해 모든 것이 디지털화 되어 가고 있다며 이를 소프트웨어 혁명이라고 하였다. 스마트폰, 자동차, 항공기, 영화, 금융, 교육, 학문 분야 등 소프트웨어가 모든 산업의 핵심을 이루고 있다.

이미지출처: http://undertheradar.co.kr/2013/07/22/

이처럼 디지털 사회의 문제는 디지털화 되어 있어 기존의 아날로그적인 문제해결 방법으로 해결하기 어려운 내용으로 매우 복잡하게 진화하였다. 예를 들면 빅데이터의 처리와 같은 문제이다. 인간의 뇌용량과 연산 속도로는 디지털화되어 있는 문제를 해결하기 어려우므로 이를 효율적으로 해결하기 위해서는 Computing Power와 이를 처리하는 특별한 사고력이 필요하다. Computing Power는 컴퓨터를 활용하는 수준에서 벗어나 문제해결의 사고과정에서부터 컴퓨터 과학자처럼 생각하는 메타 사고를 통해 이제 Computational Thinking의 이야기를 시작해보고자 한다.

Computational Thinking 보기

미래인재연구소장
한 선 관
han@ginue.ac.kr

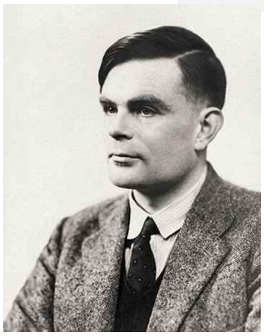
번스타인 부부의 명저 '생각의 탄생'에서 인간의 사고를 천재(다양한 분야에 종사하는)들의 사고를 분석하여 연구하였다. 그들은 천재들이 고등사고 능력을 발휘하는 방법을 종합해보면 총 13가지의 특징들을 가지고 있다고 보았다. 13가지의 생각도구들은 하위 단계부터 상위 단계로 분류되는데 크게 두 가지로 범주화된다.

(1) 관찰, 형상화, 추상화 → 패턴인식, 패턴형성 → 유추

(2) 몸으로 생각하기, 감정 이입, 차원적 사고, 모형 만들기, 놀이, 변형 → 통합

(1)단계에서 유추 능력이 가장 고급사고이자 CT의 핵심 부분과 가까운 사고능력이다. (2)단계는 인간 고유의 능력으로 기계(특히 컴퓨팅 머신)와는 차별화되는 특징이다.

앨런 튜링은 (1)단계 영역의 고급 사고(추상환, 패턴, 유추 등)를 계산적인 과정



즉, 컴퓨팅으로 가능할지 고민하였다. 일반적인 사실에서 관찰을 이끌고 형상화하고, 추상화하는 단계를 거쳐 인간의 지능을 가장 잘 표현하는 예측 능력을 유추에 의해 이끌어내는데 과연 계산으로 가능한지에 대한 고민이 세상을 바꾼 그의 첫 논문, "계산가능수와 결정문제에 대한 응용에 관하여" 이다.

이러한 튜링의 천재적 발상으로 산업과 삶의 전체가 바뀐 세상은 컴퓨터가 인간의 기본적인 활동들뿐만 아니라 유추하는 고급 사고력 영역까지 자동화하면서 인간의 직업을 컴퓨터에게 양보해야 할 처지가 되었다. 인간과 기계가 경쟁하게 될 미래에 대해서는 에릭 브린욜프슨의 '기계와의 경쟁'에서 찾아볼 수 있다.

이에 따라 우리는 생각의 탄생에서 제시하는 (2)영역의 사고력에 보다 집중해야 할 필요가 있다. 이 영역은 아직까지는 컴퓨터가 인간보다는 부족한 영역임은 부인할 수 없는 것이다. 물론 이러한 영역도 결국은 지능을 가진 기계(소프트웨어)에게 자리를 내주게 되겠지만 말이다. 시간이 문제일 뿐이다.

무거운 이야기로 서론을 이끌었는데 이제 이야기의 본론으로 넘어가자

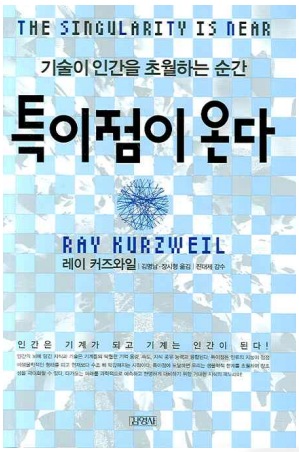
Computational Thinking? 어렵다. 용어 자체도 그렇고 한글로 번역하기도 어렵다. CT라고 번역해 쓰기로 했는데 아직까지도 Computation = 컴퓨팅 = Computing 이라는 공식이 절대 마음에 들지 않는다. 한글로 굳이 써봐도 그렇다.

컴퓨테이셔널 사고 = CT ....음.....

한글식 용어의 옳고 그름은 넘어간다. 용어의 무한루프에 빠져서 시간을 낭비하고 싶지는 않으니까 말이다.

인간의 사고력에 대한 고찰은 이해하기 어려운 부분이 있다. 창의력, 비판력, 논리력 등등 사고력이라는 것 자체가 추상적이기 때문이다. 대신 관점을 바꾸어 Computational Thinking이란 사고력의 용어 대신 그런 사고력을 가진 구체적인 사람, 즉, Computational Thinker(이런 사람에 대한 한글식 명명을 어떻게 할지에 대해서는 또다시 고민해봐야 한다)에 대해 고찰해 보면 어떨까?

현시대는 디지털 기술의 침공으로 기존 세상보다 두 배로 확장되었다. 아니 두 배 이상으로 확장되었다. 차라리 배수로 표현하기보다는 특이점(Singularity)에서 소개한 지수함수의 증가로 표현하는 것이 적절하겠다. 레이 커즈와일에 쓴 '특이점이 온다'라는 책을 보면 우리 사회의 변화, 발전 그리고 질적, 양적 증가에 대한 지수적 증가의 사례를 무수히 제시한다.



이러한 특이점 현상으로 인해 우리에게 주어진 문제와 해결 전략도 이제는 더 어렵고 더 많은 방법을 고민해야만 해결할 수 있게 되었다. 이러한 현실은 우리를 더 복잡한 상황에 던져주었고 우리에게 주어진 문제는 제대로 해결하기 위해 당연히 아날로그와 디지털의 통합 즉, 디지로그적 관점과 다른 영역과의 융합적인 관점으로 여러 분야를 넘나들며 해결해야 하는 상황이다. 이들을 연결해 줄 수 있는 융합적 사고력이 필요하다. 그래서 교육의 목표도 창의적인 문제해결력을 지닌 융합인재를 양성하는데 주력하고 있다. 이 부분에 대해서 교육정책자들과 연구자들, 교사들, 학부모들은 제대로 인식하고 있는지 궁금하다.

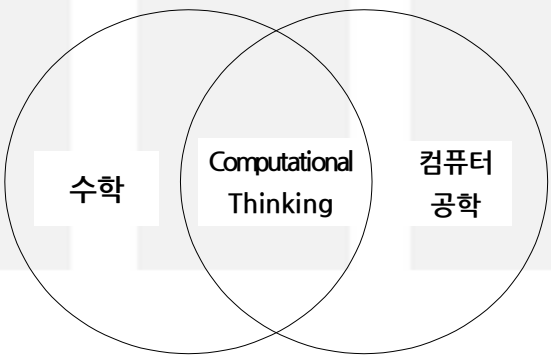
아날로그와 디지털이 뒤섞인 시대의 복잡한 문제는 아날로그 정보를 바탕으로 디지털 정보를 효과적으로 다루고 이를 바탕으로 문제를 해결해야 하며 이 과정에서 또 다른 창의성이 요구된다. 디지털 기술을 제대로 이해하거나 마음대로 주무를 수 있는 창의적 SW융합인재가 필요한 것도 바로 이 때문이다. 그렇다면 디지털 정보를 이용해 창의적으로 문제를 해결하는 사람은 어떤 사람이며 어떠한 사고가 요구되는 것일까?

디지털 정보로 문제를 해결하기 위해 우리는 컴퓨터라는 기계를 사용한다. 또한 컴퓨터라는 기계가 아직은 인간이 데이터를 가공해서 먹여주어야 하는 바보라서 자동으로 문제를 풀기는 어렵고 그 문제를 처리할 수 있도록 문제를 설계하고 처리할 수 있도록 인간이 나름대로의 절차적인 처리 과정을 만들어주어야 한다. 그 일을 하는 사람이 바로 **컴퓨터 과학자**이다. 컴퓨터과학자는 보는

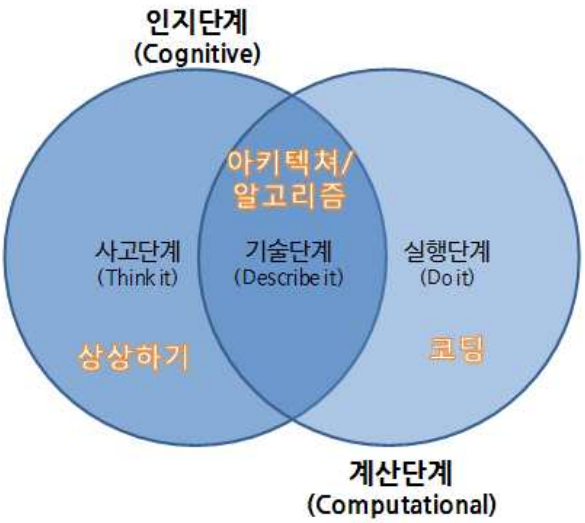
관점에 따라서, 요구하는 역량에 따라서 프로그래머, 시스템 관리자, 소프트웨어 설계자라고도 할 수 있다. 컴퓨터 과학자는 아날로그 세계와 디지털 세계의 중간 연결자로서 융합의 전문가인 동시에 수학을 매개로 자연의 언어(fact→data))를 기계의 언어(data→information)로 변환하여 처리해 주는 사람이다. 이러한 과정에서 컴퓨터 과학자처럼 생각하는 것이 바로 Computational Thinking이다. 컴퓨터 과학자는 수학의 언어를 이용하여 기계로 구현을 하기 때문에 계산적인 사고(그냥 직역하였다)를 하는 사람, 즉 Computational Thinker라고 할 수 있다. 앨런 튜링이 애초에 그렇게 아이디어를 내고 설계를 했다.

Computer Scientist = Computational Thinker
(컴퓨터 과학자) (계산적으로 사고하는 사람)

여기서 사용되는 수학의 특징은 인간(수학자)이 일반적으로 생각하는 방식의 수학이라기보다 튜링식 사고(기계를 전제로 한 계산적 사고)로 처리할 수 있도록 하는 것이다. 무엇인가 실제 적용해서 문제를 해결하고 산출해내는 다른 분야의 공학에서도 이러한 부분이 비슷하게 고민될 것이다.



예를 들면, 수학적으로 사고하여야 하는 것인데 이는 아날로그 언어를 디지털 언어, 즉 컴퓨터를 작동시킬 수 있는 기계어로 바꾸어 주어야 한다. 이것이 바로 디지털이며, 0과 1로 처리할 수 있는 최소단위 bit로 나타내는 것이다. 이 영역이 바로 컴퓨터 과학에서 말하는 이산 수학의 가장 기초가 되는 개념이다.



Hadley Wickham 개념 모형, 출처: Hadley Wickham "dplyr" talk, at useR 2014

Hadley Wickham(2014)이 제시한 개념 모형을 보면 CT의 모형과 절차와 일맥상통하는 부분이 있다. 인간의 사고 과정(think it)을 추상화의 관점에서 언어, 공식이나 상징으로 기술(describe it)하고 코딩(프로그래밍)을 이용하여 실행(do it)하는 단계를 인간의 인지(Cognitive) 단계를 계산(Computational)으로 처리하는 단계로 보고 있다. 이런 관점이 바로 인간의 문제해결을 Computational Thinker의 측면으로 바라보는 것이다.

글이 장황해지고 더 어려워진 듯하다. 튜링의 고민으로 다시 돌아가 쉽게 논의해보자. 컴퓨터를 생각해낸 앨런 튜링은 '계산 가능성 이론'을 기반으로 하였는데, 이는 일반적 알고리즘 보다는 인간의 뇌의 특성을 염두에 두고 고찰한 것이다. 계산 가능성 이론은 어떤 문제를 알고리즘으로 해결할 수 있는가를 다루는 이론이며 알고리즘은 재귀적인 방법을 통해 문제를 해결하는 이론이라고도 할 수 있다.

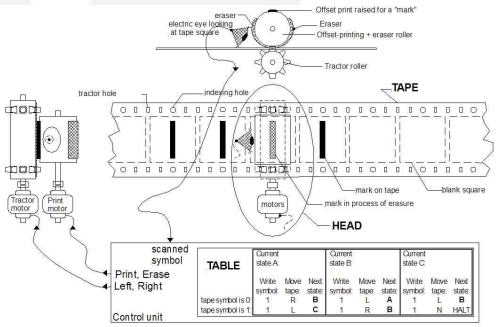
예를 들어, 5+4에 대한 결과값으로 바로 머릿속에 9라는 답이 떠오르지만 컴퓨터의 경우 5에 1의 값을 하나씩 늘리며(변수값에 추가하면서 반복되는 재귀적인 방법으로 계산하는 과정을 거치게 된다. 인간의 계산에 비해 컴퓨터는 매우 비효율적으로 보인다. 하지만 반대로 수가 커졌을 경우, 예를 들어 495683794+948372849의 계산 결과값은 인간은 듣고서 바로 답을 떠올리기

힘들게 된다. 왜냐하면 수의 길이에 대한 부담과 함께 연산 과정에서 생기는 중간 계산값(연산과정에서 나타나는 올림수, 부분계산의 결과값)에 대한 기억과 연산, 제어를 동시에 하기가 어렵기 때문이다. 뇌의 특성이 어떤 일이 주어지면 그 일의 부분을 분해하여 패턴을 찾고 문제를 동시다발적으로 처리하여 결과를 취합하는 병렬성에 강하다지만 주어진 특징이 다른 두 가지 일을 병렬로 동시에 처리하기는 어렵다. 삼각형과 원을 두 손으로 동시에 그리기 어려운 점을 생각해 보면 된다. 하지만 컴퓨터는 연산, 제어, 기억장치가 구분되어 정확하면서도 빠르게 결과값을 계산해낸다. 이러한 큰 수의 덧셈(덧셈 연산에 의한 올림값의 기억)이나 정렬(비교 연산에 의한 기억) 작업과 같은 문제들에서 생기는 인지적 과부하를 기계가 대신 처리해 줄 수 있도록 하는 필요성을 느껴 문제를 해결하고자 하는 것이 바로 '컴퓨터 과학자처럼 생각하기'이기도 하다.

즉, 컴퓨팅 파워를 이용하여 문제를 해결하는 것이다. 또한 컴퓨팅 파워로 문제를 해결하기 위해 컴퓨팅 기계가 처리하도록 효율적으로 알고리즘화 하는 것이 Computational Thinking이다.

*참고자료

튜링은 1936년 "계산 가능한 수와 결정문제의 응용에 관하여(On Computable Numbers, with an Application to the Entscheidungs problem)"라는 논문에서 인간의 논리적 사고를 기계에 비유한 구상을 밝혔었다. 'Computable'은 말 그대로 '계산 가능한' 이라는 뜻이며, 계산을 능히 잘 할 수 있는 사람이 'Computer'라 불리는 사람이었다. 이는 튜링 머신(Turing Machine)의 원형이 되었고, 튜링 머신은 계산을 수행하기 위한 기계에 불과했었다. 튜링 머신은 테이프와 머리, 상태 기록기, 유한한 표(행동표)로 구성되어 있으며, 작동 방식은 기계가 인식하거나 기록할 수 있는 기호, 0과 1이라는 테이프 기호를 사용하였다.



A fanciful mechanical Turing machine's TAPE and HEAD. The TABLE instructions might be on another "read only" tape, or perhaps on punch-cards. Usually a "finite state machine" is the model for the TABLE.

이처럼 ‘컴퓨터 과학자’처럼 생각하는 것이 'Computational Thinking'이며, 이를 우리나라에서는 ‘CT’라고 명명하고 있다. 하지만 위에서 말한 것처럼 컴퓨터 과학자가 문제를 해결하는 과정을 살펴보면 이는 수학을 근거로 하여 계산적으로 사고하고 있다는 것을 볼 수 있다. 이러한 의미에서 계산적으로 사고하는 컴퓨터 과학자 즉, ‘Computational Thinker’가 가지고 있는 사고력인 'Computational Thinking'은 ‘CT’라는 이름보다 ‘계산적 사고’라고 일컫는 것이 더욱 적절하다고 볼 수 있겠다. 더 구체적으로 명명한다면 ‘(디지털, 이산) 계산적 사고’일 것이다.

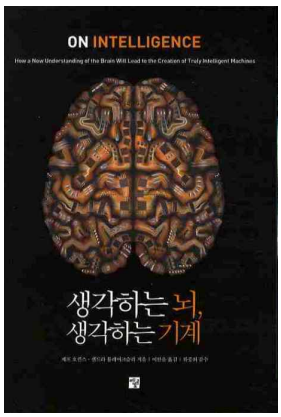
Computational Thinking의 절차 또는 구성요소를 살펴보기 위해 다시 생각의 탄생 중 (1) 영역의 내용을 살펴보자. 그리고 기존 연구(CSTA, Google)에서 제시한 Computational Thinking의 절차 또는 구성요소를 살펴보자

생각의 탄생: 관찰, 형상화, 추상화 → 패턴인식, 패턴형성 → 유추

CT: 자료 수집 및 분석, 분해, 패턴, 추상화, 알고리즘 그리고 자동화

인간의 사고과정은 감각기관을 통해 자연현상을 관찰하고 여기서 수집된 것을 모아 분석한다. 즉 인간의 뇌에서 이해할 수 있는 것으로 자료화 한다. 뇌는 자연스럽게 오감각을 최대한 사용하며 형상화하려 할 것이다. 특히 시각에 많이 하는 인간은 시각적인 형상화 작업을 주로 한다. 만일 바흐나 베토벤이라면 소리, 피카소, 고흐라면 그림을, 요리사라면 맛과 냄새를 형상화하여 데이터 처리를 할 것이다.

이렇게 형상화 한 것을 언어, 수, 음표, 색, 기호 등으로 추상화를 하게 된다. 추상화를 통해 나름의 상징화 과정을 거치면 자연스럽게 나타나는 것이 바로 패턴이다. 패턴의 발견과 스스로 만들어 보는 과정에서 다음에 일어날 시간적 사건, 공간적 변화를 예측하며 추론(유추)을 하게 된다. 이 유추 과정이 생명체의 생존과 직결되고 인간에게는 보다 고등적인 사고로 연결되어 문제를 해결하는 귀중한 도구로 활용된다. 유추의 과정은 과거를 돌아보며 자신의 문제를 해결하는 도구이며 미래를 예측하는 도구이기도 하다. 예측은 인간의 뇌가 가진 고유의 기능이며 추상화를 위한 고도의 생존 능력이다.



이러한 유추와 예측은 제프 호킨스와 샌드라 블레이크슬리가 저술한 '생각하는 뇌, 생각하는 기계'의 내용을 살펴보면 자세히 알 수 있다. 그 책에서 컴퓨팅 기계의 관점에서 인간의 뇌의 구조와 기능을 분석하며 CT를 다른 관점으로 이해하는 좋은 방법을 제시하고 있다.

이제 이러한 인간의 유추와 예측을 포함한 사고과정을 기계에 담아보자. 기계가 생각을 하도록 우리가 각 단계를 의미 있게 재구성해 주면 되는 것이다. 여기서 고민할 것은 인간의 관점에서 고민한 것을 기계의 특성에 맞게 어떻게 재구성하는가이다. 무엇을 염두에 두고, 어떤 방법을 사용하며, 기존의 예시 방법이 적절한지 고민하면서 이제 Computational Thinking의 절차 또는 세부 구성요소를 살펴보면서 글을 정리한다.

자료의 수집, 분석

이 부분은 많은 사람들이 ICT활용교육의 자료수집, 분석과 혼동한다. 자료를 순서대로 뽑아내어 엑셀로 입력하거나 그래프로 나타내는 정도를 의미하는 것은 아니다. 그 목적에 있어서 컴퓨팅 처리 또는 소프트웨어를 개발하기 위한 형태의 자료로 인식하고 추출하고 분석해야 한다. **실제 세계에서 자료를 수집한다는 것은 컴퓨터 과학자의 관점에서 컴퓨팅으로 처리할 수 있는 디지털의 형태로 뽑아낸다는 의미**이다.

컴퓨팅에서 자료는 bit, 이진수, 01, on, off, TF 로 디지털화된 것을 바탕으로 한다. 사람의 목소리를 인식하기 위해 수집하라는 것은 ICT 기기를 활용하여 여러 사람의 목소리를 녹음하라는 뜻이기도 하지만 기계가 사람의 목소리를 인식하기 위해서는 어떠한 형태의 자료로 수집해야 하며 어떻게 분석해야 기계가 계산 처리 가능한가를 염두에 두어야 한다. 그렇다면 이제 사람의 목소리를 수집할 때, 어떠한 방법으로 수집하고 잡음이 섞일 때(그런 데이터는 나중에 처리하기 어렵거나 쓸모 없기 때문에, 즉 잡음 데이터를 제거하는 계산 알고리즘이 너무 복잡하거나 제거하는데 비용이 많이 드는 것을 고려해야 하기 때문에) 어떠한 방법으로 해결

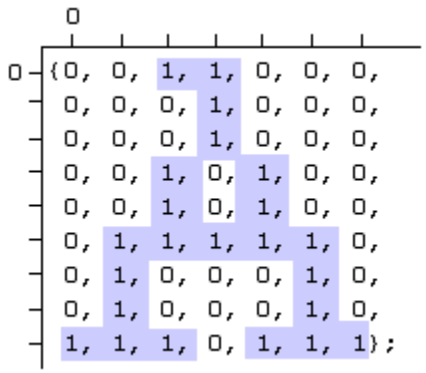
할지도 고민해야 한다. 수집된 자료의 분석 또한 이러한 점을 고민해야 한다.

자연에 존재하는 나무 자료를 수집하라든가 공기의 흐름, 사람들의 인식들을 수집하는 것은 컴퓨팅의 처리과정을 염두에 두고 수집하고 분석해야 한다는 의미다.

분해

분해는 컴퓨터가 처리할 수 있도록 자료의 분해, 절차의 분해, 문제의 분해로 구분된다. 분해도 역시 디지털 처리가 가능하도록 이진수화 하여 자료를 분해하거나, 절차도 나름대로의 알고리즘이 분리될 수 있는 것을 고려해야 한다. 문제 또한 각 모듈의 독립적 기능 즉, 객체로서 분해 가능한지 고려해야 한다.

A라는 문자 패턴을 인식하기 위한 자료 분해의 예를 보자. 눈이 없는 컴퓨터가 문자를 인식한다는 것은 실세계의 사실을 이진수로 자료화하여 0과 1로 분해하는 것을 가장 먼저 처리한다.



A라는 문자를 가로 7, 세로 9개의 격자로 분해하여 0과 1로 채워 이진 자료화한다. 분해를 통하여 이제 컴퓨터가 A를 인식하기 위한 기초 데이터가 마련된 것이다.

패턴

패턴은 나름대로의 추상화된 공식이나 개념을 완성하기 위해 꼭 필요한 과정이다. 프로그래밍 언어의 기본적인 제어문으로 반복문, 조건문, 호출문, 함수, 변수화, 동시처리 등을 고려했을 때 이러한 패턴은 컴퓨팅의 효율적인 알고리즘과 추상화된 공식을 추출하거나 처리하기 위해 매우 중요한 단계이다.

추상화

추상화는 Computational Thinking의 가장 중요한 구성요소이다. 사고력을 신장시키기 위한 목적으로 Computational Thinking교육이 중요시되는 것과 일맥상통한다.

사각형을 정의할 때 언어적 관점에서 복잡하게 설명할 수 있지만 수학적 정의는 고도의 추상화 용어를 통해 단순화하고 기호화 한다. 이 과정에서 복잡한 언어적 구조는 학생의 이해를 힘들게 한다. 수학적 단순화와 기호상징화 또한 학생의 수준에서 도전을 두렵게 한다. 물론 이 두 가지(언어, 수학)의 접근전략을 통해 고도의 사고력을 신장시키려는 의도가 있지만 말이다.

프로그래밍을 통해 사각형을 그리게 되면 언어적 표현과 수학적 상징을 프로그래밍 언어의 관점에서 조각 지식들을 절차화하고 상징화되는 부분을 인자값으로 정의하여 언어적 복잡성과 수학적 상징화의 어려움을 극복하게 된다.

사고의 절차화와 수학적 접근을 통해 추상화 과정을 거치게 되어 보다 적정한 수준에서 용어와 개념을 이해하고 문제를 해결하게 된다. 추상화된 프로그램 모듈의 인자값을 변화시키면서 자연스레 시뮬레이션 과정을 거치고 손으로 사각형, 오각형, 육각형을 그리는 것보다 숫자값을 다양하게 받은 컴퓨터가 대신 도형을 그려줌으로서 자동화 단계를 거치게 된다.

당연히 학습자들은 힘들게 도형을 그리는 대신 숫자를 바꿈에 의해 자신이 추상화한 프로그램 모듈로 자동적으로 다양한 시물레이션을 하며 개념과 문제를 해결하게 된다.

이러한 사각형 수업에서 CT는 추상화와 자동화의 효율적 사고력 신장의 목표를 명확히 해야 하며, 학습자들이 언어로 사각형을 표현하거나 수학적 기호와 상징으로 표현하는 과정을 프로그래밍 과정과 비교하는 단계가 있어야 한다. 또한 손으로 사각형, 오각형을 그리는 것과 컴퓨터로 자동적으로 그리는 과정을 비교하여 그 효율성을 학습자들이 인식하고 자신의 미래를 위해 어떤 것을 선택하여 사용할지를 결정하는 과정을 수업에 추가해야 한다.

알고리즘

알고리즘은 절차적인 사고에 대한 형상화 작업이다. 표현방법으로 순서도, 스토리로 나타내기, 슈도코드 등이 있다. 그런데 학생들에게 제시할 때 흔하게 실수하는 것이 바로 컴퓨팅 또는 프로그래밍을 고려하지 않고 예시를 든다는 점이다. 예를 들면 라면을 끓이는 순서, 이를 닦는 순서, 학교 가는 절차 등을 알고리즘으로 표현하라고 한다. 그래서 무엇을 하란 말인가?

수학에서 배웠고, 기술에서 배웠고, 국어에서 배웠고, 과학에서 배운 건데 왜 또 여기서 배워야 하는가?

자동화를 목적으로 컴퓨팅 처리를 하거나 소프트웨어를 개발할 수 있는 것을 알고리즘으로 예시를 들게 되면 알고리즘의 이해나, 그 효과, 쓸모에 대해 학생들은 제대로 배우게 될 것이다.

특히 알고리즘에서 고려해야 할 시간과의 싸움(처리 횟수), 공간과의 싸움(저장 공간, 메모리 사용)이 포함된 예시를 사용하면 그 효과가 좋을 것이다.

자동화

이 부분이 우리가 가장 많이 하는 부분이다. 스크래치든, 파이썬이든 구현해 보는 과정이다. 그냥 만들어 보는 것이 아니라 인간의 수고와 노력을 대신하는 자동화가 포함되어야 한다. 이를 통해 인간의 직업과 삶, 그리고 우리가 선택해야 할 미래에 대해 깊게 고민할 수 있는 개방적 실습이 된다.

추상화부터 알고리즘, 자동화를 위한 자료의 예시 중 흔한 실수가 간단한 것을 구현해 보는 것이다. 초보자를 위해 간단한 것을 하는 것은 좋으나 경험이 있고 나름의 컴퓨팅 개념과 실습이 적절히 이루어지면 계산 가능 한계의 수나 빅데이터 처리를 고려한 활동을 고민해야 한다.

간단한 덧셈, 구구단, 사각형을 그리는데 굳이 컴퓨터까지 동원해야 할까?

CT수업은 다양하게 접근할 수 있으나 '컴퓨터 과학자처럼 생각하기'의 관점에서 제시하는 수업은 다음과 같은 단계를 거치면서 CT의 자료수집, 분석, 분해, 패턴,

추상화, 알고리즘, 자동화의 의미를 제대로 이해할 수 있다.

단 계	내 용
① 언플러그드 활동	학생들이 이해하기 쉽게 작은 단위의 활동으로 구성
	-> 이 때 주어지는 문제는 굳이 컴퓨터로 처리할 필요가 없다.
② 계산 가능 한계의 수	언플 활동에서 다룬 데이터의 개수를 인간이 기억하거나 연산하기 어려운 정도의 예시로 '컴퓨팅 파워'의 필요성을 인식
	-> 이 과정에서 계산 기능 한계의 수에 다다르도록 데이터를 증가시켜 보면 컴퓨터의 역할(SW, 알고리즘, 프로그래밍)을 이해할 수 있다. 이 과정에서 자연스럽게 데이터의 분해, 패턴인식, 패턴 생성을 통한 추상화가 가능하고 문제 해결의 방법으로 알고리즘을 이해한다.
③ 빅 데이터 활동	거대한 수를 다룸으로서 컴퓨팅 파워의 강력함과 현실 세계 문제 해결을 경험
	자동화 이해

[참고문헌]

레이 커즈와일(2007), 김명남, 장시형 옮김, 특이점이 온다, 김영사, 파주

로버트 루트번스타인 , 미셸 루트번스타인(2007), 박종성 옮김, 생각의 탄생, 에코의 서재 출판사, 서울

에릭 브린욜프슨, 앤드루 매카피(2013), 정지훈, 류현정 옮김, 기계와의 경쟁, 틔움 출판사, 서울

제프 호킨스, 샌드라 블레이크슬리(2010), 이한음 옮김, 멘토르 출판사, 서울

Hadley Wickham(2014), Hadley Wickham 개념 모형, 출처: Hadley Wickham "dplyr" talk, at useR 2014

Jeannette Wing과 함께 바라본 CT

미래인재연구소 연구원
김 도 용
seofuiii@naver.com

1. CT(Computational thinking), 보편적인 사고와 기술을 지향하다.

CT는 앞으로 읽기, 쓰기, 셈하기처럼 사회를 살아가기 위해서 누구나 지녀야 할 핵심 역량이 될 것이다. CT는 컴퓨터 과학자처럼 세상의 문제를 바라본다. 컴퓨터 과학자들은 컴퓨터가 풀 수 있는 형태로 우리의 문제를 추상화하고, 알고리즘으로 엮어낸다.

- 기계의 소형화와 사물 인터넷의 발달
- 다양한 도구 및 자료의 오픈소스화
- 웹으로 연결 된 개인과 개인의 영향력 증대
- 컴퓨팅 기기 성능의 확장과 데이터의 빅뱅!!!

위의 네 가지 현상은 과학 기술의 발달과 인터넷 망의 개선으로 일어났다. 우리가 사용하는 컴퓨터의 성능, 휴대성은 나날이 발전하고 있고, 이제는 보일러, 정수기와 같은 가전제품에도 컴퓨터가 삽입되고 있다. 우리는 단순히 문제를 해결하는 것이 아닌 주어진 문제를 어떻게 해결하는 것이 더 편한지 고민하는 사치를 누릴 수 있게 되었다. 그리고 그런 활동의 바탕에는 컴퓨터를 이용하는 문제해결 능력 이른바 CT가 반드시 필요하다. 이것이 기존 시대의 읽기, 쓰기, 셈하기에서 벗어나 CT라는 새로운 역량이 탄생하게 된 이유이다.

CT는 하나의 학문 갈래가 아닌 핵심 역량이기에 남녀노소 누구나 쉽게 접해야 한다. 앞으로의 교육에서는 이것이 더욱 강조되어야 한다. 미학령기 아동들

에게는 그들에게 맞는 적절한 수준의 교수법을 개발해서 가르쳐야 할 것이며, 학령기 아동들에게는 기존 교과와 연계해서 컴퓨터 언어, 컴퓨터 미술 등 컴퓨터를 활용하는 간학문적 범주로 접근해야 할 것이다. 학교를 졸업한 후에도 시민 사회 교육의 일종으로 CT는 계속 교육되어야 한다.

2. CT, 무슨 사고(思考)를 그렇게 부르는가?

- 예제 - 학생들에게 경인교대 근처에서 10000원 내외로 만족도가 높은 식사를 찾게 하는 방법

CT를 활용하여 문제를 해결하는 것에는 기본 전제가 있다. 컴퓨터 공학 및 과학의 기본 개념을 활용해야 한다. 컴퓨터의 본질은 반도체의 전자 게이트를 이용한 계산기이다. 즉, 먼저 문제를 컴퓨터가 이해할 수 있도록 분해하고 추상화해야 한다. 또한, 프로그램의 본질을 파악해야 한다. 위 예제는 굳이 컴퓨터의 도움을 받지 않아도 해결할 수 있다. 주변 행인들에게 물어봐도 충분히 해결될 수 있는 문제이다. 그러나 프로그램이란 누가 언제 어떻게 돌려도 입력한 것에 대해 동일한 논리적 결과 값을 출력해낸다. 즉, 객관적이면서(*실수가 없고) 누구든지 동시에 적용 할 수 있게 해결하는 것이 CT를 활용한 문제 해결법이다.

1) CT의 시작, 분해와 추상화

CT를 제대로 활용해서 문제를 해결할 때는, 먼저 문제 속의 현상과 그 자체에 대한 정의를 해야 한다. 가령, 만족도가 높다는 질문은 현 시대에서는 컴퓨터가 알아들을 수 없다. 만족도에는 어떤 요소가 있는지, 그 요소를 수치화할 수 있는지를 생각해보는 것은 컴퓨터에게 우리의 문제를 전달하는 방법이다. 문제를 마주할 때, 필요한 속성만 골라내어 컴퓨터가 이해할 수 있게 수치화 하는 것을 추상화라고 부르며 이는 CT의 시작이다.

㉠ : 만족도가 높은 식사의 조건 찾기
㉡ : 경인교대 근처에서 10,000원 내외로 식사할 방법 찾기
㉢ : ㉡에서 찾아낸 방법들을 수치화하기
㉣ : ㉠을 바탕으로 평가표 만들어 결론내리기

2) 단계와 단계의 관계를 찾고, 시뮬레이션과 자동화.

위의 '식당 찾기' 알고리즘에서 각 단계 사이의 관계는 어떤지 살펴보자. 예를 들어 단계 ㉠과 ㉡은 서로 종속의 관계가 아니다. 독립적으로 시행되어도 전체 알고리즘에 오류를 주지 않는다. 연산장치가 2개가 있다면 2개를 독립적으로 구동할 수 있다. 단계 ㉢은 어떨까? 반드시 ㉡이 선행되고 나서야만 시작할 수 있다. 마지막으로 단계 ㉣은 ㉠,㉡,㉢과정을 통해 데이터베이스가 구축되어야만 진행 할 수 있다. 각 단계의 독립성과 종속성을 분석한 결과를 바탕으로 이를 재배열한다면 알고리즘을 아래와 같이 개선할 수 있을 것이다.

㉠-1 : 만족도가 높은 식사의 조건 검색
㉠-2 : 경인교대 근처에서 10000원 내외로 식사할 방법 검색
㉡ : ㉠-2를 수치화
㉢ : ㉡ 값을 ㉠-1에 비교하여 결론내리기

3) 컴퓨터 과학과 함께

이제 실제로 이 알고리즘을 컴퓨터에서 구동시켜보자. 어떻게 하면 더 효율적으로 알고리즘을 구동할 수 있을까? 예를 들어 단계 ㉢에서 결론을 내릴 때 어떻게 하면 전체 프로그램의 속도를 높일 수 있을 것인가? 데이터를 조금이라도 적게 다루기 위해서는 단계 ㉡에서 특정 점수 미만의 데이터를 자동으로 삭제해버리는 방법이 있다. 이는 데이터 캐싱을 줄여 물리적 비용을 절감할 수 있다. 논리적으로 이미 완성된 만들어진 알고리즘도 컴퓨터 과학적 요소를 적용하여 보면 개선할 수 있는 여지는 많이 남아 있는 것이다.

4. CT, 추상화(Abstraction)로 실체(Substance)를 얻다.

말하기가 언어와 목소리, 셈하기는 수와 규칙을 재료 및 도구로 삼는 것처럼 CT는 추상화를 사용한다. 현상을 바르게 추상화시키거나 문제를 단계별로 분해하여 구동하는 것, 혹은 단계와 단계의 관계를 정의하는 일은 모두 추상화의 일부이다. 또한, 이런 과정을 거쳐 이미 추상화 된 사고들을 더욱 심화하는 것도 역시 추상화 과정에 포함된다. 현재 컴퓨터의 진보된 연산능력과 기계 공학, 컴퓨터 과학의 발달은 우리가 추상화한 내용을 조작적으로 구현할 수 있게 한다. 각종 개발 도구를 이용해서 쉽게 그림/애니메이션을 만들기도 하고, 만들어진 자료는 다양한 방법으로 출력할 수 있다. 우리의 생각과 사고를 과거에는 상상하지도 못하는 방법으로 실체화하고 있는 것이다. 이처럼 CT의 핵심인 추상화는 일상생활의 자동화에 많은 관련이 있으며, 우리 사회에 일어나고 있는 기술적 진보는 CT의 필요성을 더욱 가중시키고 있다.

5. CT, 간학문적 요소를 가지다.

CT는 새로운 학문적 분류가 아닌 역량에 속하기 때문에 이미 모든 학문에 걸쳐서 활용되고 있다.

- 과학 분야에서는 DNA 지도 해석, 태양의 표면, 지구의 내부 모습 추리 등 우리가 직접 해보지 못하는 것들에 대해 시뮬레이션을 하며 예상과 추론 과정을 거친다.
- 공학 분야에서는 설계에서 시뮬레이션을 통해 시제품을 제작하지 않고도 예상되는 다양한 문제를 발견해내고 해결해 간다.
- 수학 분야에서는 인간이 연역적인 증명만으로는 풀어내기 어렵던 4색 지도 칠하기의 증명에 결정적인 도움을 주기도 한다.
- 사회 과학에서도 이는 활발하게 쓰인다. 경제학에서 그래프 추이를 계산하여 효율적인 투자를 할 수 있게 해주고, SNS 및 웹 서비스가 유저 친화적으로 변

하고 있는 것도 CT의 활약이다.

- 자동화 시스템이 더욱 정교해짐에 따라 로봇 수술이나, 자동 건강검진 등이 생겨났고, 게임, 예술, 스포츠 등의 분야에서도 CT가 활용될 수 있는 부분은 언제든지 존재할 것이다.

세상의 모든 일에는 수치화 할 수 있는 부분이 있다. 그리고 컴퓨터는 인간보다 월등한 연산능력을 가지고 있다. 위 두 가지 명제를 연결하는 것은 문명의 발달과 직접적인 관련이 있다. 인간의 바람을 추상화하여 컴퓨터의 힘을 빌려 기계화(자동화)해내는 것은 인류 기술의 진보를 이룩할 것이다.

6. CT, 메타포(metaphor)와 오퍼레이션(operation)으로 완성되다.

CT를 학교 현장에 도입할 때는 고려해야 할 요소가 상당히 많다. 그 중 가장 대표적인 것은 **'어떻게 가르칠 것이냐'**이다. 대부분의 현장에서 학생들은 프로그래밍에 흥미가 없거나 그것과 관련 된 직업을 원하지 않는다. 이런 학생에게 CT의 핵심인 **컴퓨터 공학적 요소 사용**은 어떻게 교육해야할까? 또, **컴퓨터 과학 용어**는 어떻게 가르쳐야 하는가? 그 답은 메타포와 오퍼레이션에 있다.

가령, 재귀를 이해하기 위해서는 하노이의 탑 비유를 사용한다. 하노이의 탑 문제는 가운데 기둥을 이용해서 왼쪽 기둥에 놓인 크기가 다른 원판을 오른쪽 기둥으로 옮기는 문제이다. 원판은 한 번에 한 개씩만 옮길 수 있으며, 작은 원판 위에 큰 원판이 놓일 수 없다는 조건이 따른다. 이 문제를 컴퓨터가 해결하는 일반적인 방법은 했던 시행을 반복하는 재귀 함수이다. 먼저, 원반이 1개일 때는 원하는 곳으로 하나를 옮긴다. 2개일 때는 1개를 임시 기둥으로 옮기고(캐싱), 원반이 1개였을 때의 시행을 2번 반복한다. 3개일 때는 2개일 때의 시행을 반복하고 맨 아래 것을 옮긴 후 다시 2개일 때의 시행을 반복한다. 즉, 일반적인 해법은 n개 일 때 n-1개의 원반을 옮기는 방법으로 옮긴 후, 제일 밑 원반을 옮기고 다시 n-1개의 시행을 반복하는 것이다. 학생들에게 이 문제를 풀게 하면서 3개일 때의 시행과 2개일 때의 시행이 어떤 관계가 있는지, 또 탈

출 조건은 어떻게 되는지를 물어보면서 자연히 알고리즘과 재귀의 개념을 학습할 수 있을 것이다.

일상생활의 모습에 빗대어서 설명해도 좋다. 메인 메모리를 냉장고로, CPU는 Register는 도마에, 그리고 요리하는 과정은 ALU+CU 등으로 나타낼 수 있을 것이다. 병렬 관계 및 교착 상태는 식사하는 철학자 문제로, 파이프라이닝을 공용 세탁기의 사용 문제로 바꾸어서 세탁을 빼는 것과 넣는 과정을 명령 파이프라인과 연산 파이프라인이 구분된 현재 컴퓨터 프로세스에 비유해서 생각해볼 수 있을 것이다. 이 외에도 비유를 활용할 수 있는 여지는 많다. 선형 구조(순차적으로 쌓여가는 자료의 모습)에서 스택(stack)은 밑에서 하나하나 쌓아올리는 접시로, 하향식 가지치기 구조 모델은 나무-가지-잎사귀 비유로, 제작과 분해는 퍼즐이나 레고, 설계도와 완성작으로 나타낼 수 있다.

7. CT, 컴퓨터와 함께 아득한 문제에 직면하다.

· P = NP 문제

P 문제는 주어진 시간 동안 알고리즘을 따라서 반복시행을 하면 쉽게 풀리는 문제를 말한다. 섞여 있는 값을 오름차순으로 배열하는 문제 등이 그에 속한다. 그에 반해 NP 문제는 해답을 내기 위해서는 모든 답을 검토해야 하는 문제를 말한다. 외판원의 최단 거리 구하기 문제나 임의의 수가 어떤 소수의 곱으로 이루어져 있는지와 같은 것들이다. P와 NP의 관계는 아직 밝혀지지 않았다. 이를 밝혀내는 것은 튜링 머신(계산기)에서 시작된 컴퓨터의 문제 해결 방식을 개선하는 귀중한 해답이 될 것이다.

· 컴퓨팅의 한계는 어디까지인가?

컴퓨터로 할 수 있는 것에 대해서도 마찬가지의 논란이 있다. 컴퓨터의 연산 능력의 한계는 존재할 것인가? 만약 존재하지 않는다면 컴퓨터의 능력은 어디까지 확장될 수 있을 것인가? 그렇게 된다면 우리는 무엇을 어디까지를 컴퓨터

라고 부를 것인가? 반도체 공정의 한계가 꾸준히 제기되고 기술적 발달로 그것을 극복해가고 있는 현 상황에서 이 역시도 계속 논의가 이루어지고 있다.

· 지성이란 무엇인가?

지성이 무엇인지도 역시 논란이다. 컴퓨터의 연산능력은 지성이라고 부를 수 있는가? 그렇지 않다면 튜링 테스트를 통과하여 인간과 구분되지 않는 컴퓨터의 능력을 우리는 무엇이라고 불러야 하는가? 휴리스틱을 사용하여 특정 값을 배제 인간의 직관을 모방한 컴퓨터의 지성이 인간의 지성과 다르다고 볼 수 있을 것인가?

· 정보란 무엇인가?

정보의 본질이 무엇이냐는 것도 하나의 중요한 담론이다. 실리콘 반도체 기반의 컴퓨터에서는 데이터를 0과 1의 배열로 나타냈다. 그러나 새로 발명된 양자컴퓨터가 양자중첩상태를 갖는 큐빗(qubit)에서는 0과 1의 동시 중첩상태를 사용한다. 데이터를 표기하던 근원이 흔들린 것이다. 이 논의는 컴퓨터 과학에서만 일어나는 것이 아니다. 예를 들면 생물학은 수많은 생물학적 발견 사실을 데이터베이스화시킨 집합체이다. 축적되어 온 관찰 값이 해당 분야를 대변하고, 나아가 새롭게 발견되는 관찰 값이 학문의 깊이를 심화시킨다. 그렇다면 생물학적 사실 그 자체는 정보에 속하는가? 그저 데이터에 속하는가? 이와 관련하여 Jeannette Wing은 데이터는 흙이고 정보는 금이라는 의미심장한 비유를 남겼다.

· 복잡도란? 더 단순할 수는 없을까?

복잡한 시스템을 단순하게 만들 수 있는 지도 중요한 논란거리다. 알고리즘이 복잡해질수록 연산 시간은 오래 걸리며 그에 따른 발열 및 전력 소모 역시 문제가 된다. 그렇다면 이런 알고리즘을 기존보다 더 간편하게 만들 수 있을까? 혹은 구상 단계에서 수많은 제안 및 요구를 이해하고 수용하면서도 최종적으로 만들어지는 시스템은 복잡도가 상당히 낮은 것을 설계할 수 있을 것인가? 이렇

듯 알고리즘과 시스템은 계속해서 간단해질 수 있을 것인가? 이는 많은 고민을 낳고 있다.

8. CT, 벗이 멀리서 찾아오니 즐겁지 아니한가?

CT는 여러 단계의 추상화를 통해 현상의 개념화, 자동화를 추구한다. 이는 현 시대를 살아가는 모든 인간이 필수적으로 알아야 하는 역량이 되고 있다. CT에는 인간이 컴퓨터를 유용하게 사용하기 위해서 거치는 모든 과정 속에 녹아 있다. 그러나 아직도 많은 사람이 CT에 대해서 생소하게 여기거나 혹은 컴퓨터 프로그래밍 요소로 오해하고 있는 것이 사실이다. 혹은, 컴퓨터 과학의 문제는 이미 다 해결되었고 이제 하드웨어의 발달만이 과제라고 주장하기도 한다. 현실은 그렇지 않다. 기술이 발달함에 따라 플랫폼은 오히려 더욱 커지고 이전에는 문제라고 여기지 않았던 것들, 혹은 이미 해결된 난제들이 새로운 과제가 돼서 우리 앞에 나타날 것이다. 1928년 증기선 윌리를 한 장 한 장 그려가며 전설의 시작을 알렸던 월트 디즈니는 12세 소년들이 너무나도 간편한 방법으로 캐릭터를 자신이 의도한 대로 자유롭게 조종하는 것을 상상하지 못했을 것이다. 이런 시대의 변화를 맞이하며 필자는 교육자로서 하루 빨리 CT를 대중들에게 소개하고 교육해야 함을 제안하고자 한다.

CT, CSTA가 말하다

미래인재연구소 연구원
류 미 영
rumy@gin.ac.kr

컴퓨팅 관련 교육에서 Computational Thinking에 대한 논의가 확산되고 미국의 컴퓨터과학을 가르치는 교사 모임(CSTA)을 중심으로 CT에 대한 구체적인 교육 자료가 제시되었다. 제시된 교육자료에는 구체적인 CT교육내용과 방법에 대한 사례가 안내되었다. 이에 CSTA에 대한 간략한 소개와 그들이 말하는 CT의 교육 내용 그리고 방법에 대해 개조식으로 살펴보고 이에 대한 개인적인 의견을 제시하였다.

1. CSTA

- 미국의 컴퓨터과학 교사 모임(Computer Science Teachers Association)
- 컴퓨터과학 교육에서 일어나는 이슈들에 대해 논의
- 컴퓨터과학 교육에 대한 다양한 자료를 공유하기 위해 만들어진 단체
- 관련 사이트 http://www.csta.acm.org/
- 2003년 K-12 학습자에게 체계적으로 컴퓨터과학을 교육할 수 있는 교육과정 기준안과 다양한 수업 형태 제안
- 2011년 개정된 K-12 컴퓨터 과학 기준안 발표(이후 CSTA 기준안)

2. 미국의 정보교육 현황

1) Jeanette Wing(2006)의 ‘computational thinking’

- 컴퓨터과학교육에 관한 것이 아니라 21세기를 살아가야 하는 모든 사람이

갖추어야 할 사고 능력에 관한 것.

- 학생들이 컴퓨터과학자처럼 생각하는 것이 아니라 일상의 생활에서 부딪히는 문제를 해결하기 위한 기본적인 능력으로 Computational Thinking (이하 CT)을 인지

2) ISTE(International Society for Technology in Education), CSTA(Computer Science Teachers Association), NSF(National Science Foundation)

- NSF는 연구를 위한 재정적인 지원을 하고 ISTE와 CSTA는 교사를 위한 자료집을 제작

3) 미국의 정보교육 현황

- CT는 필수교과가 아니며 학교에서 다루어지지도 않고 있는 현실
- 하지만 CT를 모든 교과와 통합하여 지도할 수 있는 방안의 필요성 인식
- 모든 교사들이 CT를 이해하고 자신의 교과에 활용함으로써 모든 학생들이 고등학교를 졸업할 때에는 CT능력을 갖추도록 해야 한다고 권고
- CT에서의 가장 중요한 개념은 문제해결능력
- CT는 새로운 것이 아니며 교사들은 이미 이러한 능력을 갖추고 있으므로 CT를 이해함으로써 교실에서 이루어지는 많은 학습활동에서 CT를 발견하고 발전시키는 것이 가능하다고 인식

3. CT의 조작적 정의 (Operational Definition of Computational Thinking for K-12 Education)

: CT 는 다음의 특성을 포함한 문제해결 과정으로 이해할 수 있다.

1) 문제해결을 돕는 컴퓨터나 다른 도구를 사용할 수 있도록 문제를 만들기 (Formulating problems in a way that enables us to use a computer and other tools to help solve them)

2) 자료를 논리적으로 배치하고 분석하기(Logically organizing and analyzing data)

3) 모형과 모의실험과 같은 추상화를 통하여 자료를 표현하기(Representing

data through abstractions such as models and simulations)

4) 절차적 사고를 통하여 해결 과정을 자동화하기(Automating solutions through algorithmic thinking(a series of ordered steps))

5) 가장 효과적이고 효율적으로 목표를 달성하기 위한 가능한 해결책을 확인, 분석, 실행하기(Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources)

6) 문제해결과정을 폭넓은 분야의 문제로 일반화하고 전이하기(Generalizing and transferring this problem-solving process to a wide variety of problems)

→ 컴퓨터 과학과 모든 학문 분야를 연관시킬 수 있는 컴퓨터로 해결할 수 있는 문제들의 해법을 분석하고 발전시키는 분명한 수단을 제공하는 문제해결방법론의 하나

→ K-12 컴퓨터 과학 표준 전체에 스며들어 있는 추상화, 자동화, 분석에 주안점을 둔 컴퓨터 과학의 핵심 원리

4. 컴퓨터과학 교육과정 국가 교육 표준 권고안

1) ACM과 CSTA가 K-12를 위한 컴퓨터과학 교육과정의 개발 배경

① 컴퓨터과학은 지적 측면에서 중요하다.

- 우리는 이미 디지털화되고 프로그래밍이 가능한 세상에 살고 있으며, 이러한 세상을 이해하기 위해서는 컴퓨터과학이 필요하다. 우리들이 일상에서 부딪히는 문제를 해결하기 위해서는 컴퓨터과학의 지식이 필수적이다.

② 직업선택의 폭이 매우 넓다.

- 최근에는 컴퓨터를 사용하지 않는 영역은 거의 없다. 영화 제작이나 질병의 유전자학에 대한 이해, AIDS백신 개발, 경영, 통계 등의 영역에서도 컴퓨터과학은 필수적이므로 컴퓨터과학은 다양한 분야에서 함께 일할 수 있는 지식을 제공한다.

③ 문제해결 방법을 가르친다.

- 컴퓨터과학은 학생들에게 문제해결 과정 자체에 관해 사고하는 방법을

가르친다. 문제를 규정하고 해법을 찾고 이를 위해 타인과 협업하는 능력을 배양한다.

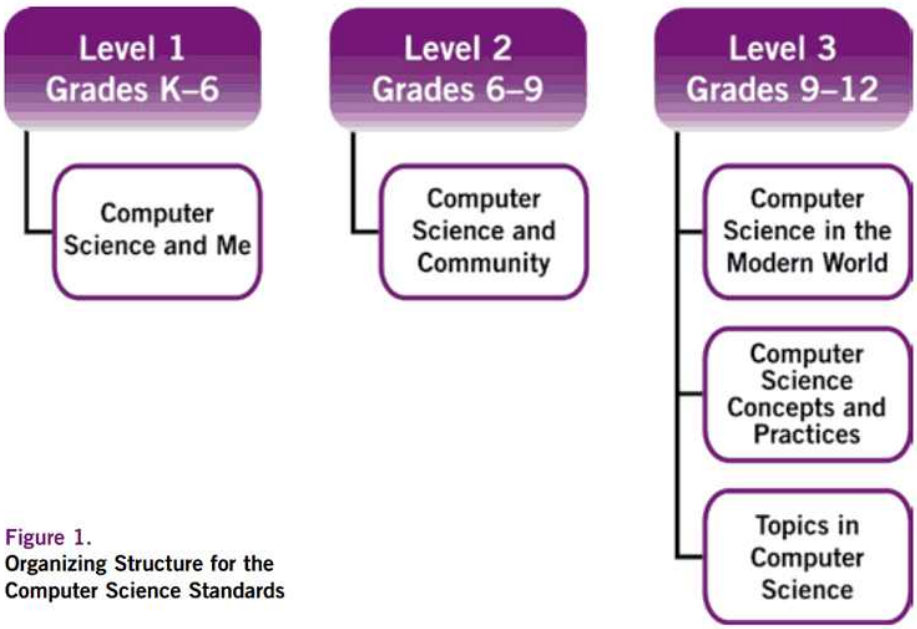
④ 다른 과학 분야를 뒷받침하고 이들 분야와 연관성을 갖는다.

- 오늘날 컴퓨터과학자가 다른 과학 분야에서 일하는 경우가 일반적이다. 모델링, 시뮬레이션, 가상화, 방대한 데이터 관리가 필요한 분야라면 컴퓨터과학은 필수적이다.

⑤ 컴퓨터과학은 모든 학생들의 관심을 끌 수 있는 학문이다.

- K-12에서 컴퓨터과학은 학생들의 관심사, 열정, 그리고 그들 주변 세상에 대한 참여 의식을 고양시키고 삶 속에서 목적과 의미를 찾을 수 있는 기회를 부여하는 것이어야 한다. 학생들은 자신의 스마트폰이나 로봇을 제어하고 물리나 생물학에서 시뮬레이션을 만들고 음악을 작곡하는 경험을 컴퓨터과학의 개념을 이용하여 수행할 수 있을 것이다.

2) CSTA K-12 컴퓨터 과학 교육과정 3단계



[그림1] 컴퓨터과학 교육과정의 구성

컴퓨터과학 교육과정의 단계별 내용을 정리하면 다음과 같다.

단계	내용
1단계 (K-6학년) Computer Science and Me (초등학교)	① CT에 관한 간단한 아이디어와 기본적인 정보기술을 통합하여 컴퓨터 과학의 기초 개념 소개 ② 이 교육과정으로부터 도출된 학습 경험은 컴퓨팅이 학생들의 삶의 중요한 부분임을 알 수 있도록 깨우치고 관심을 유도하고 도와주어야 함 ③ 학습 경험은 능동적인 학습, 창작, 탐구 활동에 초점을 맞추어 설계되어야 하고 사회과학, 언어, 수학, 및 과학 등의 다른 교과 내에 포함되도록 하여야 함
2단계 (6-9학년) Computer Science and Community (중학교)	① CT를 문제 해결의 도구로써 사용. ② 컴퓨팅의 편재를 인식하고 컴퓨터과학이 의사소통과 협력을 촉진하는 방법임을 인식하기 시작 ③ 관련된 이슈들을 언급하는 수단으로써 CT를 경험 ④ 교육과정으로부터 도출된 학습 경험은 학생과 연관되어야 하고 진취적이고 능력을 가진 문제해결자로서 스스로를 인지하도록 촉진 ⑤ 컴퓨팅이 학생들의 삶의 중요한 부분임을 알 수 있도록 깨우치고 관심을 유도하고 도와주어야 함 ⑥ 학습 경험은 능동적인 학습과 탐구 활동에 초점을 맞추어 설계되어야 하고 컴퓨터과학 과정을 명시적으로 가르치거나 사회과학, 언어, 수학, 및 과학 등의 다른 교과 내에 포함되도록 하여야 함
3단계 (9-12학년) Applying concepts and creating real-world solutions	① 학문으로써 컴퓨터과학을 3개의 과정으로 분리하고, 학생들은 고급 컴퓨터 과학의 개념을 습득하고 가상공간이나 실제의 제품을 개발하는 데 이 개념을 적용. 학습 경험은 실세계 문제의 탐구와 해결책을 구하기 위한 CT의 적용에 초점을 맞춤. 이러한 과제는 협력학습, 프로젝트 관리, 효과적인 의사소통을 포함. ⓐ 현대 사회의 정보과학(Computer Science in the Modern World) ⓑ 정보과학의 개념과 실습(Computer Science Concepts and Practices) ⓒ 정보과학의 주제(Topic in Computer Science)

3) CSTA K-12 컴퓨터 과학 교육과정 내용 영역

[그림2] 컴퓨터과학 교육과정의 요소

① 1단계 : '컴퓨터 과학과 나(Computer Science and Me)'는 저학년부터 6학년까지의 학생을 교육대상으로 선정하였으며, 학년에 따라 2개의 과정으로 교육내용을 나누어 제시

〈표1〉 K-12 컴퓨터 과학 표준 교육 1단계 교육내용

구분	Grade K-3	Grade 3-6
Computational Thinking	・나이에 맞는 문제를 해결하기 위한 기술적 자원(퍼즐, 논리적 사고) 사용 ・단계별로 생각, 아이디어, 이야기를 설명하기 위해 쓰기 도구, 디지털 카메라, 그리기 도구 사용 ・컴퓨터를 사용하지 않고 생일과 같은 정보를 순서대로 정렬하는 것을 이해하기 ・소프트웨어는 컴퓨터의 동작을 제어하기 위해 작성된다는 것 이해하기 ・0과 1이 정보를 표현할 수 있음 알기	・알고리즘 문제 해결에 필요한 기본단계를 이해하고 사용 ・컴퓨터 없이 간단한 알고리즘 이해 ・비트열이 문자와 숫자정보를 표현하기 위해 사용될 수 있는지 보여주기 ・문제를 해결하는데 시뮬레이션이 어떻게 사용될 수 있는지 설명하기 ・하나의 큰 문제에 접근할 때 고려해야 하는 하위 문제 목록 만들기 ・컴퓨터과학 및 기타 분야 사이의 연관성 이해

② 2단계 : '컴퓨터 과학과 커뮤니티(Computer Science and Community)'은 6학년부터 9학년까지를 교육대상으로 선정

〈표2〉 K-12 컴퓨터 과학 표준 교육 2단계 교육내용

구분	교육내용
Computational Thinking	・솔루션을 설계하는 문제 해결의 기본 단계의 절차적 사용 ・문제 해결에 관한 병렬화 과정 설명하기 ・컴퓨터로 처리할 수 있는 명령으로 알고리즘 정의하기 ・같은 문제를 해결하는 데 사용되는 여러 가지 알고리즘 평가하기 ・검색 및 정렬 알고리즘 만들기 ・명령 절차를 설명하고 분석하기 ・소리, 그림, 숫자, 텍스트 등을 다양한 방법으로 데이터 표현하기 ・문제의 구조 및 데이터를 시각적(그래프, 차트, 네트워크 다이어그램 등)으로 표현하기 ・학습과 연구를 위해서 콘텐츠별로 모델 및 시뮬레이션 상호작용 시키기 ・현실 세계를 정확하게 나타내는 컴퓨터 모델 분석하기 ・문제를 분해하기 위해 추상화 사용하기 ・컴퓨터의 계층 구조와 추상화의 개념 이해하기 ・숫자, 논리, 집합의 기능과 같은 수학요소와 컴퓨터과학 사이의 관련성 생각하기 ・CT에 대한 학문 간에 응용 프로그램의 예 제공하기

③ 3단계 : '개념 적용과 실제 솔루션 생성(Applying Concepts and Creating Real-World Solutions)'은 9학년부터 12학년까지의 학생을 교육대상, 3개의 과정으로 나누어 제시. 현대사회의 정보과학(3A, Computer Science in the Modern World), 정보과학의 원리(3B, Computer Science Principles), 정보과학의 주제(3C, Topic in Computer Science). 3B과목은 선택교과이며, 3A 과정을 선행해야 함

〈표3〉 K-12 컴퓨터 과학 표준 교육 3단계-3A/3B 교육내용

구분	3A(Computer Science in the Modern World)	3B(Computer Science Principles)
Computational Thinking	・복잡한 문제를 단순한 부분으로 나누기 위해 이미 정의된 함수, 파라미터, 클래스, 메소드를 사용하기 ・소프트웨어 문제를 해결하기 위해 사요한 소프트웨어 개발 절차 설명하기 ・알고리즘의 블록을 만드는 방법으로 절차, 선택, 반복, 재귀 설명하기 ・수집한 대규모 데이터를 분석하기 위한 기법 비교하기 ・2진수와 16진수의 표현 사이의 관계 설명하기 ・디지털 정보의 다양한 형식에 대한 표현과 장단점 분석 ・각종 데이터가 컴퓨터 시스템에 저장되는 방식 설명하기 ・자연현상을 이해하고 표현하는데 모	・문제를 다루기 쉬움, 어려움, 컴퓨터로 해결할 수 없음으로 분류하기 ・휴리스틱 알고리즘의 가치 설명하기 ・고전적 알고리즘을 비판적으로 검토하고 독창적인 알고리즘 구현하기 ・효율성, 정확성, 명확성의 측면에서 알고리즘 평가하기 ・데이터 분석, 활용하기 ・데이터 구조들(배열, 리스트) 사용법 비교하고 대조하기 ・다양한 2진수 형태(명령, 수, 문자, 소리, 이미지 등)의 해석에 대해 토론하기 ・과학적 가설의 검증을 돕기 위해 모델과 시뮬레이션 사용하기 ・모델링과 시뮬레이션을 통해 데이터 분석하고 패턴 찾기

	델링과 시뮬레이션 사용하기 · 문제의 복잡성을 관리하기 위한 추상화 값에 대해 토론하기 · 병렬 처리의 개념 이해하기 · 예술분야에서의 컴퓨터의 기능 이해	· 새로운 함수와 클래스를 정의하며 문제 세분화하기 · 프로세스를 나누어 쓰레드로 만들고 데이터를 병렬 스트링으로 나누어 시연하기

3C의 경우 상단의 교육과정과 다르게 다음과 같은 3가지 영역 중 1개를 선택할 수 있다.

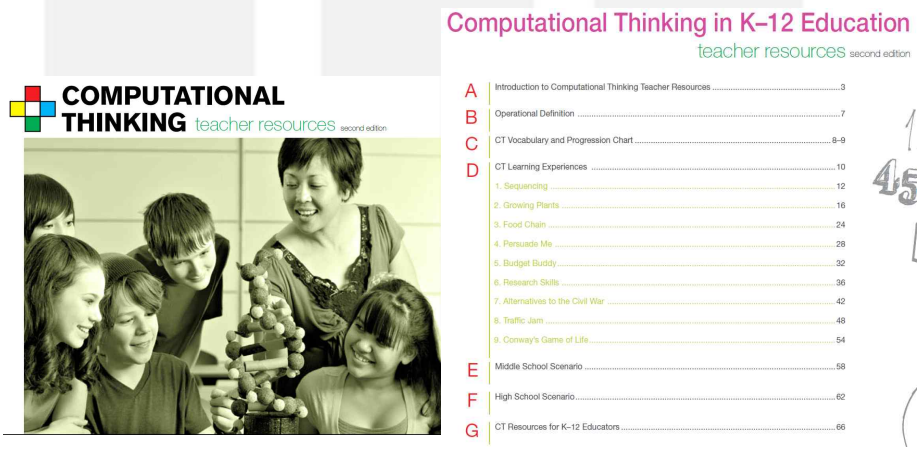
첫째, AP 컴퓨터 과학

둘째, 단일주제 심화 프로젝트과정

셋째, 전문인증관련 벤더 공급과정

5. 교육에의 적용사례

'Computational Thinking in K-12 Education(2011)'에 제시된 CT의 교육과정에의 적용 사례는 전체적으로 통합적 접근을 취하고 있다. CT는 모든 교과에 적용할 수 있다는 점을 다양한 교과와의 통합된 사례를 통하여 제시하고 있다.



[그림3] Computational Thinking in K-12 Education 자료집

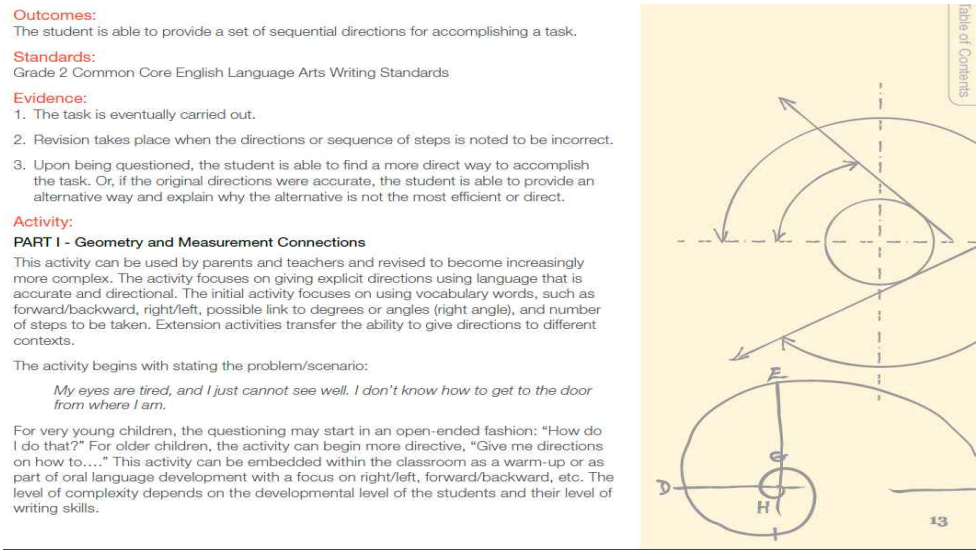
아래 표는 'Computational Thinking in K-12 Education(2011)'에서 제시하고 있는 CT의 세부 구성요소이다(현재 구글에서의 CT구성요소임).

	정의	K-2	3-5	6-8	9-12
자료 수집	해결해야 하는 문제와 관련된 알맞은 자료를 모으는 과정	경사길을 가장 빨리 내려오는 차를 찾기 위한 실험을 수행하고, 결승선을 통과하는 순서 기록	에시이 쓰기 전략을 파악하기 위해 글쓰기 예시 검토	문제 해결에 필요한 정보를 모으기 위한 설문문항 검토(예:친구에게 지난달 결석한 적이 있는지 등)	학생들이 '지구온난화가 삶의 질을 변화시켰는가?'에 대한 문항에 답을 하기 위해 설문문항 개발하고 양적, 질적자료 수집
자료 분석	자료를 이해하고 패턴을 찾아 결론을 도출	장난감 차의 무게차이를 고려하여 경주결과를 일반화하여 결론을 도출. 장난감 차의 무게 변화에 따라 결과가 바뀌는 것을 보고 결론을 검증	평가척도(루브릭)를개발하기위해 글쓰기 예시들을 좋은 것과 나쁜 것으로 분류.	디지털 계측기로부터 수집된 자료를차트로 만들고 분석차트에 나타난 동향, 패턴, 변화 및 이상점을 기술	"지구 온난화가 삶의 질을 변화시키지않았다"는 가설을 검증하기에 가장 적절한 통계 방법 사용
자료 표현	적절한 그래프, 차트, 글, 그림 등으로 자료 정리	무게가 변화할 때 장난감 차의 속도가 어떻게 변하는지를 보여주는 차트 작성	평가척도를 활용하여 각각의 예시글들을 평가하고 그 결과를 보여주는 차트 제작	자료를 다양한 형태의 차트들로 표현해 보고 가장 효과적인 시각화전략 선택.	"지구온난화가 삶의 질을 변화시켰는가"라는 질문에 관련된 자신의 입장에 따라 같은 자료를 다양한 방법으로 표현. 다양한 표현은 다양한 결론 도출 가능
문제 분해	문제를 해결 가능한수준의 작은 문제로 나누기	학교를 작은 여러 구역으로 나누고 각 구역에 대한 지도를 제작. 각 구역지도를 모아 학교 전체캠퍼스 지도를 작성	환경친화적 학교를 만드는 계획을 수립 종이와 캔의 분리 수거 전기절약, 음식물 쓰레기 재활용등으로 전략 세분화	월간 뉴스레터 발간을 위해 필요한 역할 책무, 시각표 및 자료 식별	"어떻게 하면 록스타가 될 수 있는가" 와 같은 큰 문제를 고려하고 작은 부분문제로 나눔. 학생이 제어할 수 있는 변인과 그렇지 않은 변인에 대해 토론
추상화	문제해결을 위해반드시 필요한 핵심 요소를 파악하고 복잡함을 단순화	세변을 가진 다양한 색상과 크기의 도형들을 삼각형으로 추상화	이야기를 듣고 주요소재를 반영하여 적절한 제목을 결정	역사의 한 시대를 공부한 후, 그 시대를 대표하는 상징, 테마 사건, 핵심인물, 가치파악(예: 국가나 단체 또는 집안을 상징하는 문장.	현대 사회의 본질적인 특성을 분석하여 정치적으로 가장 유사한 시대 선택.
알고리즘과 절차	문제를 해결하거나 어떤 목표를 달성하기 위해 수행되는 일련의 단계	학교에서 출발하여 지역의 주요 지형물을 찾아갈 수 있도록 길안내서 만들기	보드게임을 만들고 게임 설명서 작성 설명서대로 게임 수행 게임을 한 친구들로부터 피드백을 받아 설명서 수정	주어진 시간 내에 미로를 탈출하기 위한 길을 찾는 로봇 프로그래밍	대학 선택을 위한 의사결정과정에 대해 논의하고 이를 설명하는 알고리즘생성 알고리즘은 친구가 다니는 학교 재정지원여부, 입학성공여부 등과 같은불확실한 변인 제어가능

	정의	K-2	3-5	6-8	9-12
자동화	컴퓨팅 시스템이 수행할 수 있는 형태로 해결책 나타내기	교실에서 편지 대신 인터넷 기반의 도구를 활용하여 다른 주나 나라의 또래들과 그들의 문화를 배우면서 이야기하기	바코드 현금인출기 도서관 바코드 등 실생활에 사용되는 예시를 통해 자동화가 무엇인지 조사	공해 자료를 수집하는 센서를 프로그래밍하여 자료를 수집한 후, 컴퓨터 프로그램을 활용하여 이산화탄소 관측치를 내림차순으로 정렬	자동화로 인해 오늘날 거의 필요하지 않은 능력과 정보를 학습하는 것에 대한 장점 논의 이 능력에는 큰 숫자나 누기, 제곱근 구하기 철자확인 통계수식, 역사적인 날짜 기억
시뮬레이션	자동화의 결과이며, 문제를 해결하기 위하여 만든 모델을 실행시켜 결과 파악하기	길 안내서를 만든 후, 만든 것이 정확한지를 확인하기 위해 길안내서에 따라 실행	과정을 이해했다는 것을 보여 주기위해 애니메이션 제작	생산자 개체들의 일정 비율이 죽으면 생태계에 어떤 일이발생하는지에 대답할 수 있는 실험을 수행하기 위해 간단한 생태계 모델 사용. 사용자가 죽음 비율조정	“생일 문제”를 시뮬레이션하기 위한 스프레드시트 제작한 방에 생일이 같은 사람이 적어도 2명 있을 확률이 50%이상이려면 몇 명의 사람이 있어야할까).
병렬화	목표를 달성하기 위한 작업을 동시에 수행하도록 자원을 구성	기준을 정하여 한 반을 두 그룹으로 나누어 서로 다른 작업 수행 각각의 결과보다 전체의 결과가 좋아짐.	교사들은 팀 프로젝트 일정 역할 과제에 대해 계획하도록 하고 과제를 완료하기 위해 함께 일함(과업을 어떻게 나눌 것인지 어떤 과업들은 순차적으로 진행해야하고 어떤 과업들은 동시에 진행할 수 있는지를 결정.	학생팀은 비디오를만들기 위하여 대본 소품, 역할을 포함한 제작계획 수립 동시에수행할 수 있는 작업과 체크인 계획, 통합등 중요 단계들을 판별	워털루 전투를 이끈 각 군대의 활동순서 기술 활동은 실제 활동병사 모집과 지적활동군대 배치 결정을 모두 포함

이 자료집에서 제시한 CT의 여러 사례 중 두 가지 사례(CT Learning Experience)를 살펴보면 다음과 같다.

① 유치원-2학년(Sequencing) : 언어적인 지시를 통하여 문제를 해결하는 경험을 할 수 있도록 상황을 제시하였다. 앞이 보이지 않는 학생을 언어적인 방향지시(왼쪽, 오른쪽, 각도)를 통하여 문을 열수 있도록 한 상황에서 학생들이 목적을 달성하기 위한 지시 절차를 경험하도록 하였다. 이러한 경험은 로봇을 움직이도록 지시하는 과정과 매우 유사하므로 목적을 성취하기 위한 여러 방법을 탐색하고 가장 효과적이고 효율적인 방법을 평가하는 활동을 포함한다.



Outcomes:
The student is able to provide a set of sequential directions for accomplishing a task.

Standards:
Grade 2 Common Core English Language Arts Writing Standards

Evidence:
1. The task is eventually carried out.
2. Revision takes place when the directions or sequence of steps is noted to be incorrect.
3. Upon being questioned, the student is able to find a more direct way to accomplish the task. Or, if the original directions were accurate, the student is able to provide an alternative way and explain why the alternative is not the most efficient or direct.

Activity:
PART I - Geometry and Measurement Connections
This activity can be used by parents and teachers and revised to become increasingly more complex. The activity focuses on giving explicit directions using language that is accurate and directional. The initial activity focuses on using vocabulary words, such as forward/backward, right/left, possible link to degrees or angles (right angle), and number of steps to be taken. Extension activities transfer the ability to give directions to different contexts.

The activity begins with stating the problem/scenario:

My eyes are tired, and I just cannot see well. I don't know how to get to the door from where I am.

For very young children, the questioning may start in an open-ended fashion: "How do I do that?" For older children, the activity can begin more directive, "Give me directions on how to…." This activity can be embedded within the classroom as a warm-up or as part of oral language development with a focus on right/left, forward/backward, etc. The level of complexity depends on the developmental level of the students and their level of writing skills.

Table of Contents

13

② 초등학교 2학년(Growing Plants) : 식물을 기르는 내용과 관련된 동화를 읽고 요약하게 한다. 동화와 관련된 다양한 질문을 통하여 문제를 확인하게 하고 이와 유사한 경험을 도출한다. 그리고 문제를 해결한 경험을 그림으로 또는 글로 표현하게 한다. '식물 기르기'과정을 '컴퓨터과학' '과학' '사회'교과와 통합한다. [컴퓨터과학] 사람이 돌보지 않고 식물을 기를 수 있는 방법으로 [스프링클러 시스템]을 도입한다. 그리고 스프링클러가 정상적으로 동작하기 위한 조건들을 탐색하게 한다. 심화과정으로서 보다 복잡한 변인들을 제공하여 깊이 있는 사고를 유도한다. [과학]식물을 기르기 위하여 필요한 지식과 기능을 습득한다. 과학교과에서 CT를 도출한다. 예를 들면 식물을 기르기 위하여 필요한 자료의 목록을 만들게 한다. 정원을 꾸미기 위한 단계적 절차를 탐색하게 한다. 식물이 자라는 과정을 기록하고 차트와 그래프로 제시하게 한다. [사회과] 참여자들 간의 자료 공유 방법으로 온라인달력의 이용, 적절한 정보를 수집하는 절차 등을 학습하는 과제를 포함한다.

CSTA에서 말하는 CT를 정리하며..

CSTA는 기존의 교실에서 이루어지는 많은 학습활동에서 이미 CT를 신장시킬 수 있는 주제와 아이디어를 쉽게 발견할 수 있다고 생각한다. 그래서 CT를 컴퓨터 과학의 영역뿐만이 아니라 모든 교과에 접목하여 융합적인 교육이 가능하도록 구성하였다. 마치 STEAM 수업과도 같게 느껴지는 부분이다.

하지만 CSTA의 CT교육 사례 중 일부의 내용을 지나치듯 보면 수년전에 우리나라 교육과정에 포함되었던 ICT활용 교육의 적용 사례들이 오버랩되며 고개를 약간 기웃하게 만드는 부분이 있다. 알고리즘에 대한 교육 내용 중에서 '길안내서 만들기'(CSTA구성요소 예시자료 참고)를 학습할 때 사회교과에서도 충분히 다루는 건데 굳이 CT라 명명하며 SW교육의 주요 소재로 사용해야 하는지에 대한 의구심과 알고리즘의 일상 생활의 실제 사례로 국어 시간에 학습 소재로 많이 사용되는 '하루 시간표 짜기' 등이 그렇다. 엑셀을 활용하여 데이터를 수집하고 그래프로 표시하는 부분도 역시 ICT활용교육의 기억이 살포시 떠오른다.

하지만 CSTA는 컴퓨터 과학을 전공한 교사들의 모임이고 CT를 컴퓨터과학의 본질을 제대로 인지한 상태에서 교육과정을 구성하고 있다.

우리나라 SW교육에 적용하기 위해서는 이처럼 교사들이 컴퓨터과학, 프로그래밍 과학에 대한 전문적인 지식과 이해가 선행될 필요가 있다. 단순히 코딩을 적용하여 신장시키려고 하는 CT수업 모형이나 CT가 일반적인 문제해결능력을 위한 사고력 정도로 생각하고 SW교육에 적용시킨다면 가르치는 교사나 학생들이 근본적인 SW교육의 목표와 철학에 혼란을 갖게 될 것이다.

CT는 컴퓨팅을 통한 문제 해결과정을 추구한다. 본문의 예시로 제시한 '스프링클러 시스템 만들기'에서처럼 컴퓨팅을 활용하여 화분에 물을 주고자 하는 문제를 해결하기 위해 타교과와 연계하여 학습을 하고 있다. CT의 본질인 추상화와 자동화를 위해 자료를 분해하여 컴퓨팅 처리가 가능한 데이터로 수집하고 분석하며 효율적인 스프링클러를 만들기 위한 패턴과 추상화 그리고 자동화를 위한 프로그래밍의 관점에서 타교과와의 융합을 추구해야 한다.

CSTA자료를 분석하면서 한 가지 더 느낀 점은 CSTA에서 제시한 예시는 이상

적인 학습이지만 CT의 여러 가지 구성요소들을 필요한 것으로 쪼개어 제시하고 있어 ICT활용 수업 또는 타교과 연계 융합수업과 혼동을 주고 있다는 점이다. 앞서 이야기 한 것처럼 시수가 적은 SW교육에서는 컴퓨팅의 과정과 컴퓨팅으로 문제를 처리하는 컴퓨터 과학자의 관점에서 문제를 이해할 수 있는 구체적인 사례를 드는 것이 좋다. 디지털화 가능한 자료의 분해와 수집, 분석 그리고 프로그래밍 문법의 절차적 순서를 구성하기 위한 패턴과 알고리즘, 추상화에 따른 구체적인 자동화를 보여주는 작지만 임팩트있는 한국형 수업 주제를 준비해야겠다. CT는 디지털과 이진수를 계산적으로 처리하는 컴퓨터 과학을 기본으로 하고 있다는 점에서 CSTA의 자료를 깊게 고찰해보도록 하자!

[참고문헌]

김홍래, 이승진(2013). 외국의 정보(컴퓨터) 교육과정 현황분석, KERIS 이슈리포트

박효민(2014). 글로벌 소프트웨어 교육 현황 및 교육 도구 동향, Internet & SECURITY FOCUS

한국과학창의재단(2014). 초중등 단계 Computational Thinking 도입을 위한 기초 연구

Computational Thinking teacher resources. www.csta.acm.org

Google's Computational Thinking

미래인재연구소 연구원
전 수 진, Ph.D.
soojin.jun@inc.korea.ac.kr

1. 교육용 프로그래밍 언어에 대한 소프트웨어교육에서의 역할과 적합성

초등에서 교육용 프로그래밍 언어(EPL; Educational Programming Language)는 코딩이 단순해야 한다. 특히 초등학생들에게는 프로그래밍에 대한 흥미를 유지시키는 게 중요하기 때문이다. 아무리 훌륭한 개념과 기능이 있어도 코딩이 어렵거나 디버깅이 어려우면 안 된다. 천재가 아니어도 코딩을 할 수 있어야한다.

EPL이 꼭 비쥬얼 프로그램일 필요는 없으나 문법은 쉬워야한다. 조잡하고 복잡한 기호로 표현하거나 OOP(Object Oriented Programming)등의 차원 높은 개념이 들어 있을 필요도 없다. APPLE사의 초기 타이핑식 언어나 촌스러운 Visual Basic도 좋다.

중요한 것은 학생들이 코딩하기 쉽고 문법적 실수를 최소화 할 수 있는 언어여야 한다. 따라서 그동안 EPL 개발자들은 최대한 자연어와 비슷하면서도 문법오류에 의한 디버깅은 최소화 하도록 언어를 고안하고자하였다. 현재 우리가 많이 사용하는 스크래치나 엔트리나 거북명령 프로그램 등이 그런 맥락이었다. 단순히 화려하고 멀티미디어 표현이 좋아서 만이 아니다.

언어도구의 디자인이나 스크립팅 방식에 있어서 연령이나 경험 또는 학교급 별로도 선호하는 언어가 있겠지만 성별이나 취향에 따라 선호하는 언어가 다를 수도 있다. 그러나 대체적으로 초등학교에서는 엔트리나 스크래치가 유용할 수 있다. 엔트리는 디자인이 예쁘고 여학생들이 좋아하는 스타일의 스토리텔링방식의 장면전환 기능이 잘되어있다. 스크래치도 비슷한 수준의 코딩이 가능하며

방송하기 기능이 엔트리에 비해 강점일 수 있다.

프로그램 간에 이러한 소소한 차이는 있지만 학생들은 텍스트 코딩언어라도 즐겁게 자주 접하는 것을 더 선호할 것이다. 필자는 개인적으로 초등학생 때 텍스트방식의 APPLE 언어를 배웠었는데 어렵지 않았고 거부감도 없었다. 언어의 종류도 중요하지만 학생들 수준에 맞는 학습 내용과 교수법도 중요하다. 결국, EPL은 공장에서 돌아가는 복잡한 기계와 같은 범용언어와는 다르게 추상적인 '퍼즐 놀이감'의 느낌을 주면된다. 초등학교 시절 학생들이 느낀 흥미는 자신감을 갖게 하고 해당 학습을 지속가능하게 하기 때문이다.

소프트웨어교육은 사고력 신장을 위한 일반교과의 특성을 지닐까 직업적 소양을 키워주는 진로 교과의 특성을 지닐까? 이러한 의문은 응용학문이 교과로서 자리매김하게 된 첫 케이스이기에 생기는 혼란일 것이다. 건축이 미래 필수 핵심역량이라면 건축과목이 생겼을 것이다. 하지만 정보교과가 지난 10년간 지속적으로 논의가 되고 있다는 것은 이제는 단순히 직업교과로서의 응용학문을 뛰어 넘는다는 의미가 될 것이다.

반면 국어과에선 일반교과냐 작가나 소설가 등의 직업 교과냐 라는 논의가 없다. 그냥 모든 영역에서 필요하기도하고 특정 직업을 갖기 위해서도 필요한 교과이다. 정보교과도 그런 맥락에서 지난 10년간 요구되어왔고 교과가 되었다. 그러나 최근 SW교육이 몇몇 자원이 풍족하지 않은 나라들에서 특별히 대두 되었을 때는 국가의 관련 직업 인재 양성의 정책적인 의도도 있었을 것이다. 우리나라 또한 그러할 것이다. 개인적인 역량 계발과 기본 소양을 갖추는 일 외에도 국가경쟁력 측면의 필요도 분명히 있기 때문이다. 그러나 초등에서는 이 모든 가능성을 열어두는 것이 좋다.

'Facebook' 창업자인 '마크 주커버그'도 처음 컴퓨터를 배울 때에는 컴퓨터과학 지식을 모두 배우고 싶다거나 관련 전문가가 되고 싶어서 시작했던 것은 전혀 아니었다고 말했다. 그는 그냥 하고 싶어서 했으며 단지 자신의 여동생과 놀고 싶었던 것뿐이었다라고 말했다. 필자도 초등학생 때 단순 흥미로 프로그래밍을 했으나 직업을 염두에 두지는 않았다. 그리고 프로그래머도 되지 않았

다. 그러나 관련 영역에서 노력하고 있는 연구자이자 교사가 되었다. 그러나 그 당시 동시에 같은 컴퓨터학원을 다니던 친구들 대다수가 컴퓨터 관련 직업을 갖지는 않았을 것이다. 이렇듯 우리는 초등학생들에게 모든 가능성을 열어주어야 한다.

2. 소프트웨어교육과 CT에 대한 의견

Computational Thinking이 기존에 없던 사고력을 명명하는 것은 아닐 것이다. 그러나 기존의 연필과 종이나 어떤 구체물로 생각을 표현하는 것이 아니라 도구와 표현 방식이 달라지기 때문에 특별한 구분이 필요하다. 같은 도구 즉, 같은 컴퓨터를 사용한다 해도 프로그래밍을 한다는 것은 단순히 프리젠테이션이나 워드작성을 할 때와는 다른 사고과정과 사고수준을 요구한다. 즉 우리가 원하는 출력물이나 반응을 얻기 위해서는 '컴퓨터의 언어'를 거쳐야하기 때문이다.

도자기를 빚기 위해 점토를 만지는 것이 아니라 도자기를 빚는 로봇을 조정해야 한다. 그러기 위해서는 참 답답한 과정을 거친다. 사람 손으로 직접 빚는 것이 훨신 더 직접적이고 자연스럽고 직관적일 수 있는데 말이다. 소소한 행동들을 모두 계산해서 조정해야하고 지시의 순서가 잘못되면 안 되며, 무엇보다 로봇은 임기응변 능력이나 눈치도 없다. 그렇기에 이렇게 한 다리 건너서 내가 원하는 것을 한다는 것은 또 다른 사고능력을 요구하는 것이다. 이러한 인지능력을 '메타인지'라고도 말한다.

메타인지는 자신의 인지적 활동에 대한 지식과 조절을 의미하는 것으로 내가 무엇을 알고 모르는지에 대해 아는 것에서부터 자신이 모르는 부분을 보완하기 위한 계획과 그 계획의 실행과정을 평가하는 것에 이르는 전반을 의미한다[1]. 따라서 이 임기응변능력 없고 눈치 없는 로봇(컴퓨터)에게 지시하기 위해서는 특별한 메타인지가 필요하며 이러한 능력이 바로 CT이다.

쉽게 말해 이러한 CT는 그러한 로봇(컴퓨터)의 수준에 맞추어 일을 수행시킬 수 있는 능력이다. 그 로봇은 단순하지만 성실하다. 그래서 단순하고 반복적인

1) Flavell, J. H. (1979). Metacognitionandcognitivemonitoring. AmericanPsychologist, 34, 906-911.

일을 시킬수 있다. 우리가 한번만 고생하면 도자기 한 개 만들 시간에 수십, 수백배 또는 무한개의 것을 만들 수도 있다. 물론 그 결과물은 우리의 의도대로 매우 창의적일수도 있다. 컴퓨터가 단순하고도 성실하게만 수행해줬음에도 불구하고... 그것 참 놀랍다. 결국 이러한 단순과 성실이 바로 CT의 추상화(Abstract)와 자동화(Automation)이다.

어떻게 CT를 신장시킬 수 있는가? 신장시키기 위한 구체적인 방법과 도구들은 무엇인가?

초등에서는 간단하다. 학생들이 그 도자기 빚는 로봇을 잘 파악하여 조정해보는 과정을 경험해 보면 된다. 물론 도자기도 빚고 악기도 연주해보고 탐험도 떠나고....로봇의 다양한 삶을 만들어 보는 것이다. 따라서 컴퓨터의 속성을 잘 이해하도록 해야 하며, 간단하고 친숙한 주제를 가지고 프로그래밍 해보는 과정도 중요하다.

CT를 신장시키기 위해서는 기능 중심이 아닌 학생들의 사고력 중심의 수업이 되어야한다. 수학에서 계산만 잘한다고 수학적 사고력이 높은 것이 아니기 때문에 다양한 발문과 조작활동을 선행하는 것과 마찬가지로 CT 신장 교육에서도 다양한 교수전략이 필요하다. 그래서 많은 연구진들은 어린 학생들이나 초보자들을 위해 언플러그드와 같은 놀이를 통한 컴퓨터과학 교육방법이나 EPL을 이용한 게임이나 스토리텔링식 개발 등을 제안하고 있다.

또한 구글이나 CSTA에서는 식물기르기, 교통체증, 먹이 연쇄와 같이 학생들의 일상생활이나 타교과와 연결된 주제중심 프로젝트를 통해 CT를 기르기 위한 교육안을 제시하고 있기도 하다. 최근 우리나라 초등학교교육도 주제중심통합교과 융합교육이 화두가 된 것은 사실이지만 사실 진짜로 그렇게 하고 있는 곳은 참 드물다. 실제로 여러 가지 학교여건상 참 쉽지 않으리라 본다. 융합형 프로젝트 학습 뿐 아니라 단순히 프로그래밍 교육만 볼 때에도 쉽지 않은 것이 기존의 따라하기식 강의법을 벗어나는 것이다.

CT를 신장시키면 일상생활의 문제해결에 진짜 도움이 될까?

단순히 다른 교과 학습에 이것이 도움이 될 것인가를 생각하는 것에서 벗어날 필요가 있다. 우리의 사고는 그렇게 단순하지는 않다. 문제 해결 능력은 매우 다양한 상황에서 필요한 능력이다. 심지어 우리는 우리 생활 전반에서 우리가 맞닥뜨리는 관계 또는 마음의 복잡한 문제 또한 우리는 면밀히 분석해야 할 때가 있다. 그럴 때, 가만히 우리의 마음과 상황을 들여다보며 그 문제들을 분해하고 이해하고자 하는 과정을 습관적으로 할 수 있어야 한다.
물론 구체적인 학습 상황에서도 긍정적인 영향은 분명히 있다. 개인적으로는 프로그래밍을 했던 경험이 나의 논리적 사고에 지대한 영향을 미쳤다. 이후 이러한 논리적으로 생각하는 습관은 학습적인 면에서는 논술이나 논문과 같은 논리적인 글을 쓰는데도 도움이 되었다고 확실히 느낀다.

Google이나 CSTA에서 정의하는 CT와 이번 2015년 정보교과에서 추구하는 CT와 어떻게 다를까?

미국 네브라스카 대학에서는 미국 과학재단의 지원을 받아 CT와 창조적 사고를 통합한 IC2Think 프로젝트에 의해 'CT Lesson Plan'을 개발하였다. 이를 Google for Education에 공개했는데, 2015년 7월에 Google CT Team 으로 통합하고 업데이트 중이다. 또한, Google에서는 CT는 다양한 특징과 성향을 포함한 문제해결 과정이며, CT는 컴퓨터 어플리케이션 개발에 필수적이지만 수학, 과학, 인문학을 포함하는 모든 분야에 걸쳐 문제 해결을 지원하기 위해 사용될 수 있다. 또한, 교육과정을 통해 CT를 배운 학생들은 학교생활, 교과목뿐만 아니라 교실 밖에서도 이를 사용할 수 있다고 설명하고 있다. 그리고 CT의 개념을 "컴퓨팅에서 문제해결과 관련된 '정신적 과정 (추상화, 알고리즘설계, 분해, 패턴인식 등)'과 '실제 산출물(자동화, 자료제시, 패턴의 일반화 등)'이라고 정의하였다.

CT에 대한 생각은 다음 표와 같이 Google이나 CSTA도 Wing[2]의 생각을 기반으로 하였다.

2) J.M. Wing, ""Computational Thinking,"" CACM Viewpoint, March 2006, pp. 33-.35.

Wing/CSTA/Google CT 비교[3)]

Wing(2008)	CSTA&ISTE(2011)	Google for education (2015)
추상화	**자료수집**	
	자료분석	**자료분석**
		패턴인식
	자료제시	**자료표현**
	문제분해	**분해**
	추상화	**추상화**
		패턴일반화
자동화	알고리즘 및 절차	**알고리즘 디자인**
	자동화	
	병렬화	
	시뮬레이션	

그리고, 이러한 CT 요소들의 각각의 세부 개념은 다음과 같다.

1) 추상화: 중심생각을 정의하기 위해 관련정보를 추출하고 식별하는 것
2) 알고리즘설계: 유사한 문제를 해결하거나 작업을 수행하기 위한 지침의 순서 시리즈를 만드는 것
3) 자동화: 컴퓨터 또는 기계가 반복 작업을 하는 것
4) 자료수집: 정보를 모으는 것
5) 자료분석: 패턴을 찾거나 통찰을 개발함으로써 자료를 이해하는 것
6) 자료표현: 적절한 그래프, 차트, 단어, 이미지로 조직하고 묘사하는 것
7) 분해: 자료, 관리 가능한 부분, 좀 더 작은 문제, 과정들을 쪼갠다.
8) 병렬화: 좀 더 효율적으로 목표를 달성하기 위해서 큰일에서의 작은 일들을 동시에 처리하는 것
9) 패턴 일반화: 예측결과를 테스트하기 위해 관찰된 패턴의 이론, 모델, 법칙, 원칙을 만드는 것
10) 패턴 인식: 데이터의 규칙성, 동향, 패턴을 관찰하는 것
11) 시뮬레이션: 실제세계의 처리과정을 모방할 모델을 개발하는 것.

그들은 우리보다도 더 세세한 교수학습 콘텐츠를 발빠르게 개발했다. 그들의 교수학습과정안을 보면 대체로 프로젝트 학습 형태로 일상생활의 문제들과 관련지어 대체로 CT를 잘 녹여 내었다. 그리고 프로그래밍 구현이 중심이라기보

3) 출처: 김석전, 전용주, 김태영, Googl CT 수업지도안 분석, 한국컴퓨터교육학회 동계 학술대회, 2016, pp.67-71.

다는 CT과정을 중점으로 활동들이 이루어진 것을 볼 수 있다. 알고리즘이나 프로그래밍을 하기 위한 상황이해나 문제해결 등의 이전 과정이 구체적이며 프로그래밍 구현 부분은 일부에 지나지 않는 것을 보면 코딩보다는 학생들의 사고과정에 더 중점을 둔 것은 사실이다.

반면 우리나라는 SW교육이라는 말이 강조 되면서 해외 사례와 같은 Wing의 연구를 기초로 한 CT를 염두에 두긴 하지만 실제로는 프로그래밍 구현에 더 많은 연구가 치중되어 있는 것은 사실이다. 물론 CT는 프로그래밍 구현 과정에서도 잘 길러 질수도 있다. 그러나 국내에서 개발하고 있는 교육과정이나 교과서에서는 아직 CT를 추상적인 개념 이상으로 충분히 녹여내지 못하고 있어 보인다. 아직은 CT에 대한 개념에 대한 이해가 정책자 뿐 아니라 교사나 각종 연구기관 모두가 상이한 것도 사실이다. 앞으로 더 많은 논의와 협의를 통해 우리나라 실정에 적합한 CT의 개념과 적용방향을 합의해 가야 할 것으로 보인다.

또한 학교급별로도 CT를 녹여내는 방법이 다를 수 있을 것이다. 초등학교에서는 광범위한 범위에서 CT를 다룰 수 있겠다. 재미난 게임이나 시뮬레이션을 설계하고 구현해 보는 간단한 활동을 통해서도 기초적인 CT는 발현될 수 있다. 그리고 이렇게 프로그래밍 구현도 즐겁고 재미있지만 그 기능에 치중하기 보다는 미래 사회에서 컴퓨터 기술이 구현될 법한 여러 가지 상황들을 가정하여 상상하여 보고 그러한 상황에서 사용자의 요구를 분석하고 새로운 기능이나 기술을 설계해보고 간단한 도구나 프로그래밍 언어로 구현해 보는 활동들을 프로젝트 방식으로 진행할 수 있겠다.

물론 이러한 과정을 학생들이 충분히 학습하기 위해서는 교육과정 내에 확보되어야 할 시수가 충분히 확보되어야 하는 것도 사실이다. 현재 초등 교육과정에 배정된 실과 17차시 정도로는 쉽지 않은 과정이긴 하다. 그러니 현실적으로는 현실 가능한 방법으로 학생들에게 CT를 경험하게 하는 것이 중요할 것이다.

K-12 교육에서의 CT

미래인재연구소 연구원
서 정 원
silove1203@gmail.com

미국의 K-12교육에서 말하는 CT에 대하여 Purdue대학교 Aman Yadav 교수의 2011년 발표자료를 참고하여 안내하고자 한다.

1. 교육에서의 Computational Thinking

현대 사회는 정보가 유력한 자원이 되어 그 가치를 중심으로 사회나 경제가 운영되고 발전되어 간다. 이전 시대에서는 단순히 정보를 획득하고 이해하는 것으로도 사회 전체에 지대한 영향을 미쳤던 반면, 현대 사회는 그러한 정보를 단순히 획득하는 것에서 나아가 그 정보를 활용하여 새로운 가치를 창출해냄으로써 사회 변화를 주도할 수 있게 되었다. 더욱이 일상의 문제들이 더욱 복잡해지고 다양해지면서 이전의 방식으로는 효율적이게 해결할 수 없어졌다. 이러한 시대 상황으로 볼 때, 미래 사회를 살아가게 될 미래의 인류에게 필요한 역량은 정보를 창의적으로 가공하여 새로운 가치를 창출해내는, 그리고 맞닥뜨리게 될 문제들을 다각적인 측면에서 효율적으로 해결해내는 능력이 될 것이다.

위에서 언급한 필요성에 의해 Computational Thinking이 현대 사회의 중요한 핵심 역량으로 꼽히고 있다. 이에 따라 미래 사회의 인재를 양성해야 하는 교육 분야에서도 CT에 대한 관심이 높아지고 있다. 특히나 미국의 경우 지금까지의 성장 동력이 자국민이 아닌 타 국민으로 이루어진 고급 인력을 바탕으로 기술 발전에 있었기 때문에, 미국 내에서의 미래 인재 양성에 주력하고 있다. 이를 입증하듯이 미국의 교육 학제 K-12 교육에서도 CT를 중시하고 있는 추세

이다. 미국을 비롯한 세계 각국의 교육 현장에서 CT에 대한 관심을 지극히 당연하고 필요한 것이다.

2. K-12에서의 Computational Thinking를 신장시키는 방식

앞서 이야기했듯이, 미국에서 CT 증진을 위한 교육은 적극적으로 진행되고 있다. 미국의 학제 K-12에서 CT를 중요시 하는 이유를 다시 정리해보자면 다음과 같다. 첫째, CT는 국가 성장의 원동력이 될 수 있다. 컴퓨팅과 관련된 직업은 미국 경제에서 가장 빠르게 성장하는 직업 중 하나이다. 컴퓨팅 분야의 진보가 경제의 모든 부문에서 중요하게 자리 잡아 가고 있음에도 불구하고, 미국의 청소년들은 컴퓨팅에 대한 기회가 별로 없기 때문에 CT 증진을 위한 교육이 필요한 것이다, (Source: http://www.nsf.gov/pubs/2010/nsf10619/nsf10619.htm) 둘째, CT를 통해 다양한 연계 효과를 얻을 수 있다. CT를 함으로써 학생들은 창의력과 문제해결력을 기를 수 있고 기존에 알고 있던 문제 해결 기술을 향상시킬 수 있다. 또한 소프트웨어를 사용하는 사용자 차원을 넘어 프로세스를 디자인하고 지식을 생성하는 창조자의 역할을 하게 된다. (Source: Pat Phillips, NECC 2007, Atlanta)

K-12에서는 CT를 증진시키기 위해 교수학습에서 CT를 적용한다. 교수 학습 상황에서 일반적인 원리를 추상화하고 다른 상황에 적용하는 방법을 배워본다. 또는 그러한 원리들을 이해함으로써 문제를 푸는 방법을 배운다. 즉, 추상화의 자동화로서의 CT를 신장시키기 위해서 학생들을 모델링과 시뮬레이션에 노출시키거나 스스로의 모델과 시뮬레이션을 구축하도록 격려하는 것이다. 이러한 교수학습에서의 적용 방법을 구체적으로 예를 들자면 다음과 같다. 첫째, 일상적인 지도에서 CT의 개념을 통합시킨다. CT 예제 및 자료를 모든 학년에 넣어 CT와 관련된 지식을 쌓을 수 있도록 하는 것이다. 둘째, 일상적인 작업에서 CT와 관련된 용어를 사용한다. 예컨대 “OO을 하기 위해 알고리즘을 만들어 보자"라고 말한다. 셋째, 비판적으로 검토하고 정보를 사용하도록 격려한다. 넷째, 추상화를 하게 한다. 그들의 학습을 다른 상황에 전이시킬 수 있는 기회를 제공하는 방법을 이용한다. 다섯째, 수업 중 CT 형성을 위한 활동을 한다.

3. K-12에서의 Computational Thinking 적용

초등 교육에서는 컴퓨터의 수행을 이해시키기 위한 기초를 형성하기 위하여, 양치질과 같은 일상적인 일을 15개의 개별적인 단계로 분해해본다, 영어 교육에서는 랩과 시 사이의 유사성을 확인하고 유사한 형태와 특징을 바탕으로 랩퍼를 시인에 매치시키는 활동을 해본다. 역사 교육에서는 시간에 따라 이민자의 비율이 변화하는 원인을 조사하기 위해 역사적인 사건과 통계 데이터를 연구한다. 그 외에도 여러 부분에서 CT를 적용시킬 수 있는데, 예컨대 복잡한 데이터들을 분석하기 위해 스프레드시트, 그래프를 사용하게 한다. 데이터 사이의 패턴(예를 들어, Facebook의 배치, Google의 공공 데이터, 아마존과 넷픽스의 추천)을 분석하고 지식을 생성해내기 위해 사용하게 하는 것이다. 또한 가설을 테스트하고 공식화할 수 있게 해주며 결과를 검토할 수 있게 해주는 시뮬레이션을 사용할 수도 있다.

반면에 고등학교에서는 CT 함양을 위한 교육이 새로운 도전에 직면한 상황이다. 현재 K-12 단계 중 고등학교 단계에서의 Computer Science 교육은 과학보다는 사업과 관련된 과목에 들어가 있다. 컴퓨터 과학을 스프레드시트 및 데이터베이스의 파생물 정도로 생각하고, CS AP 교과 과정은 프로그래밍에 집중되어 있다. 따라서 수업은 폭 넓은 기술 및 사회 이슈보다는 프로그래밍의 세부 사항에 초점을 맞추는 경향이 있다. 이에 따라 CT와 CS의 원리를 포함하는 CS 교육을 확대할 필요가 있다. 또한 이전 학년에서의 CT를 더 향상시킬 수 있도록 새로운 교육을 도입해야할 것이다.

4. K-12에서의 Computational Thinking에 대한 나의 의견

K-12에서의 CT 교육을 살펴보며 가장 인상적이었던 점은 교육 콘텐츠의 질이나 제공에 관한 부분이다. 몇몇의 적용 예시들을 보며, 복잡하지 않으면서도 효과적인 교육 내용을 담고 있다는 생각이 들었다. 예컨대 일상적인 작업인 양치질을 분해해보는 활동 예시나 여러 웹사이트의 데이터 사이의 패턴을 분석하여 지식을 생성해내는 활동 예시를 살펴보며 단순하지만 CT의 요소를 경험해 볼 수 있는 좋은 예제라는 생각이 들었다. 그리고 이러한 좋은 콘텐츠의 개발

및 제공이 가능한 것은 미국 내의 다양한 기관의 협력과 콘텐츠 공유가 가능했기 때문이라고 생각한다. (K-12에서의 CT를 소개한 Purdue 대학의 강의 자료 전문을 살펴보면, CT 교수학습 자료를 제공하는 다양한 기관을 소개하고 있다.) 우리나라의 교육 역시 제한된 기관에서 교육 자료를 개발 및 제공하기보다는 더 많은 기관과의 협력을 통해 질 높은 교육 콘텐츠 및 자료를 개발하여 제공해야할 것이다. 또한 교육 자료나 교수학습방법에 대한 국가 교육과정 상의 안내가 너무 제한적일 경우 실제 현장에서의 보급 및 적용이 어려워질 것이라고 생각한다. CT와 같이 복잡 다양한 상황에서 길러지는 역량일수록 더 유동적이고 유연하게 적용할 수 있도록 개방적인 안내가 이루어져야 한다.

이와 관련하여 교사의 역량이 매우 중요하다고 생각한다. 정해진 매뉴얼 안에서 천편일률적인 CT 교육을 하기 보다는 학생들의 사고를 확장하고 진정한 CT를 경험할 수 있도록 적절한 교육 내용을 제공하고, 적절한 방식으로 진행할 수 있는 역량이 필요하다. 또한 K-12의 예에서 보았듯이, 초등교육에서의 CT 역시 매우 중요하다. 이는 중등교육에 종사하는 교사보다 상대적으로 CT에 대한 개념이 빈약한 초등교사 역시 이제는 CT에 대해 더욱 연구하고 현장에서 적용하려는 노력을 해야 한다는 것을 뜻한다. 우리나라의 경우는 특히 초등교사에 대한 CT 및 컴퓨터과학에 대한 연수를 통해 인식의 변화와 역량의 신장이 필요하다. 한 번에 모든 교원들의 인식과 역량을 변화시킬 수는 없을 것이다. 소프트웨어교육이나 컴퓨터과학교육에 관한 전문 교원을 활발하게 양성하여 그들을 중심으로 CT 교육이 제대로 보급될 수 있도록 적극적인 지원이 필요하다고 생각한다,

Purdue 대학의 강의 자료를 살펴보며 고민하게 된 점은 CT교육에 관한 평가 부분이다. 처음 서론 부분에서 언급했듯이 CT는 현대 사회의 새로운 역량으로서 일종의 문제해결역량 혹은 가치창출역량이라고 할 수 있기 때문에, 지식과 정보의 획득이 이루어졌는지를 확인해보던 기존의 평가 방식과는 차별화되어야 한다. 교육 장면에서 학생들이 맞닥뜨린 문제에 대해 CT를 바탕으로 해결하였는지를 확인하는 데에는 다각적인 평가 방식이 필요할 것이다. 물론 K-12의 교육 내용 중 단순히 CT 관련 용어를 써보는 활동 등은 기존에 지식을 확

인해보던 평가 방식을 사용해도 무방할 것이다. 그러나 그러한 기초적인 단계를 거쳐 궁극적으로 얻고자 하는 CT력을 바탕으로 한 문제해결능력을 확인해 보기 위해서는 CT의 각 요소를 잘 적용했는지 확인해 볼 수 있는 평가가 필요할 것이다. 데이터 수집, 데이터 분석, 데이터 표현, 문제 분해, 추상화, 알고리듬, 자동화, 시뮬레이션, 병렬화 등에 얼마나 능숙한지, 이를 얼마나 잘 활용하여 문제를 해결해내는지 평가해야 한다. 현재 CT 교육 평가 자료가 활발하게 개발되고 있는데, 개발자와 평가를 직접 사용하게 될 교사들 모두 CT의 본질에 대해 생각하고 이와 관련하여 평가에 대해서도 고민해야 할 것이다.

[참고문헌]

Aman Yadav, Ninger Zhou, Chris Mayfield, Susanne Hambrusch and John T. Korb(2011), Introducing Computational Thinking in Education Courses, Proceedings of the 42nd ACM technical symposium on Computer science education, pp.465-470

영국의 컴퓨팅 교육과정

-

미래인재연구소 연구원
조 현 룡
hrcho@ice.go.kr

교육부 소프트웨어 교육 운영 지침(2015.2.)에서는 초·중등학교에서 소프트웨어 교육에서 추구하는 인재상으로 'CT를 가진 창의·융합 인재'를 양성하는 것을 목표로 하고 있다. 이를 위해 기본적으로 갖추어야 할 역량이 바로 'CT'이다. 컴퓨팅 교육과정의 선구자인 영국의 컴퓨팅 교육과정에서는 CT에 대하여 컴퓨터가 문제를 해결하는 것을 돕도록 우리가 접근하는 일련의 사고과정이라고 정의하고 있다. 우리나라 교육부 지침에서도 이와 같은 맥락으로 CT에 대하여 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력이라고 명시하였다.

CT는 컴퓨팅 교육과정을 설계하기 위한 핵심이므로 CT가 추구하는 것이 무엇인지를 이해해야 한다. 이에 대하여 영국과 우리나라의 교육과정에서 제시하는 CT의 구성 요소를 비교하면 다음과 같다.

	영국 교육과정	교육부 운영지침
논리적 추론	예측하고 분석하기	자료를 분석하고 논리적으로 조직하기
알고리즘	단계를 만들고 규칙 정하기	알고리즘적 사고를 통하여 해결방법을 자동화하기
추상화	불필요한 세부적인 것들을 제거하기	모델링이나 시뮬레이션 등의 추상화를 통해 자료를 표현하기
패턴들과 일반화	발견하고 유사한 것들을 이용하기	문제를 컴퓨터로 해결할 수 있는 형태로 구조화하기
평가	판단하기	효율적인 해결방법을 수행하고 검증하기
		문제 해결 과정을 다른 문제에 적용하고 일반화하기

흔히 CT에 대하여 '컴퓨터에 대해 생각하거나 컴퓨터처럼 생각하는 것'라고 생각하는 경우가 많은데 이는 대표적인 오개념이다. 영국의 교육과정에서는 CT는 '컴퓨터로 작업을 하기 전에 이루어진 사고과정'이며 '컴퓨터가 우리에게 문제를 해결할 수 있도록 돕는 방법을 생각하는 것을 찾는 과정'이라고 설명하고 있다. CT로 문제를 해결하기 위해서는 두 가지 단계를 거치는데 첫째, 문제를 해결하기 위한 단계를 생각하고 둘째, 컴퓨터로 하여금 문제를 해결하게끔 하기 위한 기술을 사용한다.

CT는 컴퓨터 과학자나 소프트웨어 개발자뿐만 아니라 다른 영역에도 매우 유용하게 사용된다. 학교에서는 찾아보면 복잡한 문제를 해결하기 위하여 작은 문제들로 나누고, 규칙들과 단계들을 정하는 과정에서 활용할 수 있다. 교육과정에서 CT를 사용할 수 있는 예는 다음과 같다.

- 이야기 쓰기활동: 주제를 작성하고 인물과 배경을 설정한다.
- 미술, 음악, 디자인, 기술: 학생들이 무엇을 만들 것인지 정하고, 작업의 단계를 설정하며, 복잡한 문제들을 몇 개의 작은 문제들로 나누어본다.
- 수학: 문제를 풀기 전에, 문제의 중요한 정보를 찾아본다.

1. 논리적 추론

논리적 추론은 어떠한 사건이 일어났을 때 그 일이 왜 일어났는지 설명하는 것이다. 컴퓨터는 주어진 명령에 의해서만 작동하기 때문에 입력에 대한 결과를 예측할 수 있다. 컴퓨팅에서는 CPU, OS, 프로그래밍 언어 등에서 논리적 추론을 사용한다.

교육과정에서 논리적 추론을 사용할 수 있는 예는 다음과 같다.

- 언어 교과 : 학생들은 이야기 다음에 무슨 이야기가 나올 것인지, 이야기에서 인물의 행동에 대해 설명하는 활동
- 과학 교과 : 실험결과를 어떻게 도출하였는지 설명하는 것
- 역사 교과 : 역사적 원인과 결과에 대한 토론 활동

컴퓨팅 교육과정에서 논리적 추론을 활용하기 적합한 단계는 다음과 같다.

- 1단계 : 학생들이 프로그램의 작업을 예측하기 위하여 논리적 추론을 사용한다.
- 2단계 : 간단한 알고리즘들이 어떻게 작동하는지 설명하기 위하여, 알고리즘들과 프로그램들에 있는 오류들을 찾고 수정하기 위하여 논리적 추론을 사용한다.

2. 알고리즘

알고리즘은 문제 상황에 직면했을 때 이 문제를 해결하는 가장 좋은 방법을 찾는 것이다. 예를 들어 길 찾기의 경우에는 시간이 가장 짧고, 빠른 길을 찾는 것이 가장 좋은 방법이다. 현실에서 알고리즘은 검색엔진, 페이스북 등에서 사용된다.

교육과정에서 알고리즘을 사용할 수 있는 예는 다음과 같다.

- 실과 교과 : 요리 레시피
- 언어 교과 : 구조적 글쓰기
- 과학 교과 : 실험 방법 알아보기
- 수학 교과 : 암산, 컴퓨터 기반의 교육용 게임

컴퓨팅 교육과정에서 알고리즘을 활용하기 적합한 단계는 다음과 같다.

- 1단계 : 학생들이 알고리즘이 무엇인지 이해하고 디지털 기기의 프로그램에서 알고리즘이 어떻게 사용되는지 알아본다. 예를 들어, 사각형 그리기의 경우 로고와 스크래치에서 그리는 법이 다른 것처럼 똑같은 문제를 해결하는데도 다른 알고리즘이 사용된다.
- 2단계 : 학생들이 나름의 목적을 가지고 프로그램을 디자인한다. 알고리즘으로 설계하고, 알고리즘을 설명하고 오류를 찾고 수정하기 위해 논리적 추론을 사용한다.

3. 분해

분해는 문제를 작은 부분들로 나눔으로써 문제를 해결하는 과정이다. 분해는 복잡한 문제들을 해결하고 큰 프로젝트들을 관리하는데 도움을 준다. 현실에서 분해는 소프트웨어 개발에서 복잡한 과정들을 작은 과정들로 나누는 것, 스마트폰, 랩톱컴퓨터 제작과 같은 하드웨어 제작에서 각각의 부품들을 전문 업체에서 따로 제작한 뒤 조립하는 것 등에서 사용된다. 학교에서도 모둠별 과제, 역할분담 등에서 분해를 사용하고 있다.

교육과정에서 분해를 사용할 수 있는 예는 다음과 같다.

- 과학, 지리 교과 : 식물의 각기 다른 부분이나 영국 안에서 각기 다른 국가를 표시한다.
- 언어 교과 : 이야기의 각기 다른 부분들을 만들어본다.
- 수학 교과 : 문제를 해결하기 위하여 문제를 쪼개어본다.

4. 추상

추상은 중요한 것이 무엇인지 식별하고 중요하지 않은 것을 무시하여 복잡한 것을 단순화 하는 과정이다. 학교의 시간표는 추상의 대표적인 예라고 할 수 있다. 시간표는 일주일의 과정과 언제 어떤 과목을 누구에게 배울 것인지를 추상화하여 함축하고 있다.

교육과정에서 추상을 사용할 수 있는 예는 다음과 같다.

- 음악 교과 : 팝송을 피아노 반주용으로 편곡한 것은 음악 조각들에 대한 추상화로 생각할 수 있다.

컴퓨팅 교육과정에서 추상이 사용되는 예는 다음과 같다.

- 학생들이 소프트웨어를 사용하거나 웹을 탐색할 때 컴퓨터 내부 또는 인터넷에서 무슨 일이 일어나는지 호기심을 갖는 것에 대해 격려하는 것
- 학생들이 그들에 대해 알고 있는 주제에 프리젠테이션이나 동영상을 함께 넣을 때, 세부적인 내용으로 들어가는 동안 핵심 정보에 초점을 맞추어야

하며 표현할 수 있는 방법에 대해 생각하는 것 : 이 과정 모두가 추상화에 포함된다.

5. 패턴과 일반화

패턴과 일반화는 문제 해결을 쉽게 하기 위해 문제를 바꾸어 만드는 것이다. 사각형의 넓이 구하기로 수학 공식을 만들어 내는 것은 패턴과 일반화를 한 것이다.

교육과정에서 패턴과 일반화를 사용할 수 있는 예는 다음과 같다.

- 어린 나이에는 동요와 이야기의 반복 문구에 익숙해지고, 시간이 지나면 전통적인 이야기나 다른 장르의 반복되는 이야기를 알 수 있다.
- 음악 교과: 학생들은 많은 음악 장르에서 반복되는 멜로디나 베이스 라인을 인식하는 방법을 배운다.
- 수학 교과: 학생들은 일반적으로 패턴을 파악하고 추론하는 연구를 한다.
- 영어 교과: 학생들은 일반적인 맞춤법 규칙과 예외인 것들을 알 수 있다.

소프트웨어는 어떻게 만들까?

1. 조작활동(만지작 거리기)

- 교사가 프로그램의 작동에 대해서 설명하는 것보다 학생들이 직접 스크래치와 같은 프로그램을 이용하여 스스로 코딩해보는 것이 더 의미가 있다.

2. 제작

- 창의적으로 제작한다.
- 프로그램의 발전을 위하여 자신과 타인의 비판을 반영한다.
- 예술적 창조성이 강조되는 작품들을 찾아본다. 예를 들어 디지털 음악, 이미지, 애니메이션 작업, 가상환경 3D 프린터기술 등이 있다.

3. 디버깅

오류를 찾고 해결하기 위해, 자신의 알고리즘과 코드를 통해 얻은 생각에 대한 책임을 학생들에게 부여하는 것은 프로그래머와 같이 작업하고 생각하는 법을 배우는 중요한 학습의 일부이다.

- 수학 교과: 풀이과정을 확인해본다.
- 영어 교과: 이야기를 수정해본다.

4. 인내심

- 컴퓨터 프로그래밍은 때로는 매우 어렵고 만들고자 하는 강한 의지가 필요하다.
- 학생들이 프로그래밍 작업을 하면서 어려움을 겪을 때 사용할 수 있는 전략을 찾아보게 한다. 이를 위해 문제점이 무엇인지 정확히 찾아보고, 해결책을 빙이나 구글 등을 통해서 찾아보게 한다.

5. 협동

- 소프트웨어는 프로그래머 팀에 의해 개발되고, 공유 프로젝트를 함께 작업하므로 컴퓨팅 수업에서 협동할 수 있는 기회를 제공하는 것은 매우 중요하다.
- 모둠별 수업에서 학생들이 각자의 역할에 대한 이해를 하는 것은 매우 중요하다.

영국의 중등 교재 '중등 소프트웨어(SW)교육 컴퓨팅 사고력 키우기1'를 보면 '컴퓨터과학, 정보기술, 디지털 활용 능력'을 지도하기 위해 창의적인 학습법을 이용하는 교재라고 그 목적을 명확히 하고 있다. 정보소양능력과 정보활용능력을 동시에 기르고자 하며 교재의 구성도 'IT-생각하기(사고토의활동), IT-계획하기(계획), IT-컴퓨팅하기(컴퓨팅 활동)'으로 되어 있다.

학습은 실제 컴퓨터가 없어도 되게, 다른 교과목과 연계 학습이 이루어지도 한다. 또한 기존에 알고 있는 것을 새로운 지식과 연결할 수 있을 때 효과적인 학습을 이룰 수 있다는 구성주의 철학을 바탕으로 하고 있음도 알 수 있다.

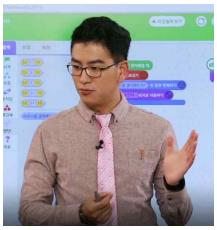
[참고문헌]

Computational Thinking, www.barefootcas.org.uk

Introduction to computational thinking, http://www.bbc.co.uk/education/guides/zp92mp3/revision

제임스 아벨라 외(2016). 박현유, 김성식, 최현종 옮김, 중등 소프트웨어(SW) 교육 컴퓨팅 사고력 키우기1, 한국과학창의재단

Creative Computing에서 추구하는 CT



미래인재연구소 연구원
홍 수 빈
subinvan@gmail.com

Computational Thinking(이하 CT)를 향상시키는 것은 소프트웨어 교육의 중요한 목표 중에 하나이다. CT에 대한 깊은 고민을 갖기 전부터 스크래치의 매력에 빠져 소프트웨어 교육을 해 온 교사의 한 사람으로서 항상 이런 의문을 가지고 있었다.

"스크래치를 배우면 자연스럽게 CT가 신장되는가?"

물론 스크래치와 같은 프로그래밍 기능만을 익힌다고 해서 CT가 길러지는 것은 아닐 것이다. 교육의 목표를 달성하기 위해서는 그에 맞는 지식, 기능, 태도를 함께 익힐 필요가 있다.
미국의 스크래치 팀에서는 Creative Computing이라는 관점에서 수년간 온라인 사이트를 운영하고 다양한 창의컴퓨팅 워크숍을 통해 축적된 연구 자료를 바탕으로 CT를 세 가지 측면에서 정의하고 있다.

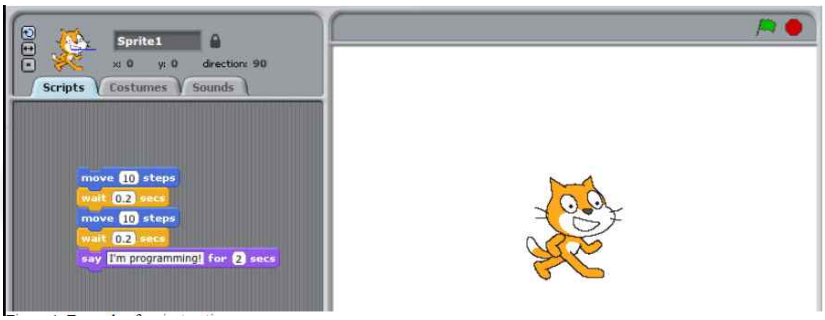
- **CT 개념(Computational Thinking Concepts)**
- **CT 실습(Computational Thinking Practices)**
- **CT 관점(Computational Thinking Perspectives)**

이는 각각 소프트웨어 교육에서 학습해야 할 지식, 기능, 태도를 의미한다. 이 글에서는 이러한 관점에서 CT를 정의하고 활동을 소개하고자 한다.

1. CT 개념(Computational Thinking Concepts)

컴퓨팅 개념은 소프트웨어 교육에서 학습해야 할 컴퓨팅 지식을 의미한다. 스크래치 팀에서는 학생들이 프로그램을 만들 때 유용하게 많이 사용하는 개념들 7가지로 정리하였다. 여기에는 컴퓨터 과학(Computer Science)에서 사용되는 개념들이 포함되어 있으며 스크래치에 국한 되는 것이 아니라 컴퓨터 프로그래밍 활동에 기초가 되는 개념이라고 할 수 있다.

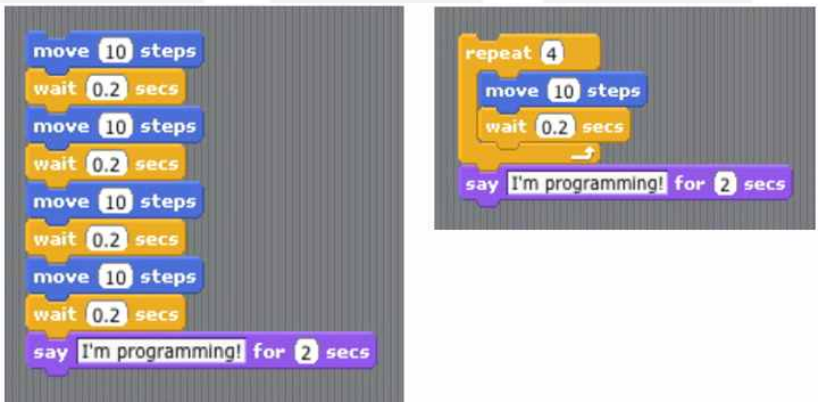
(1) 시퀀스(Sequence) : 과업이 순차적으로 실행되는 과정을 파악하는 것이다. 이를 위해서는 행동이나 과정을 명시적으로 설명할 수 있어야 한다.



https://scratch.mit.edu/projects/97342218/

위와 같은 스크립트를 실행시켰을 때 스프라이트에는 어떤 변화가 나타날 것인지 짐작해 보고 설명하는 활동이 대표적인 시퀀스 파악 활동이 될 수 있다.

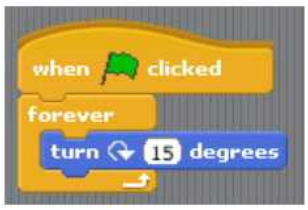
(2) 반복(Loops) : 같은 시퀀스를 여러 번 실행하게 만들어 주는 것이다. 코드의 효율성과 가독성을 높일 수 있다.

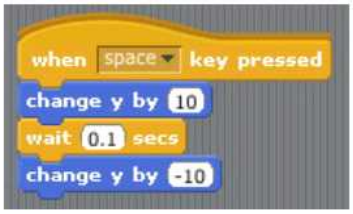


https://scratch.mit.edu/projects/97343283

스크립트에서 똑같이 반복되는 부분을 찾아 반복하기 블록을 사용하여 간단하게 나타낼 수 있다. 이러한 과정에서 추상화 기법이 사용된다.

(3) 이벤트(Events) : 어떤 일이 일어나게 하는 원인이 되는 것이다. 방송하기 기능 등은 상호작용하는 프로젝트를 만들기 위해 필요한 핵심적인 요소이다.

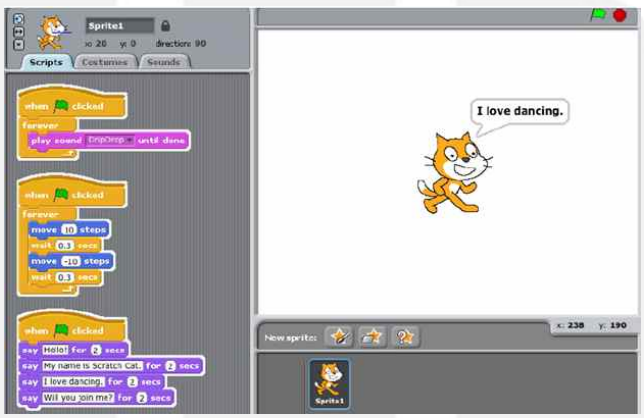




https://scratch.mit.edu/projects/97344120/

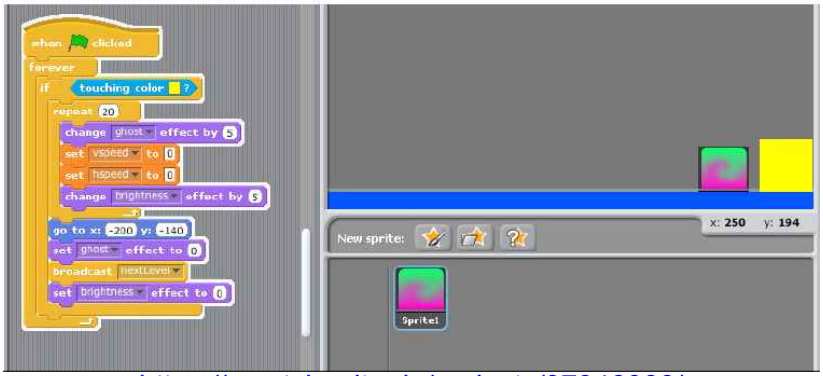
프로그래밍에서 언제 명령을 수행할 것인가는 매우 중요한 부분이다. 이는 한 명령을 수행한 이후에 다른 명령을 수행할 시점을 정하거나 사용자와 상호작용을 하는 부분에서 프로젝트를 만들 때 반드시 필요한 요소이다.

(4) 병렬처리(Parallelism) : 여러 시퀀스가 동시에 실행되게 한다. 병렬처리는 둘 이상의 객체를 동시에 실행시킬 때도 사용가능하며, 하나의 객체 내부에서 다른 동작을 동시에 실행하는 것을 의미하기도 한다.



https://scratch.mit.edu/projects/97345782/

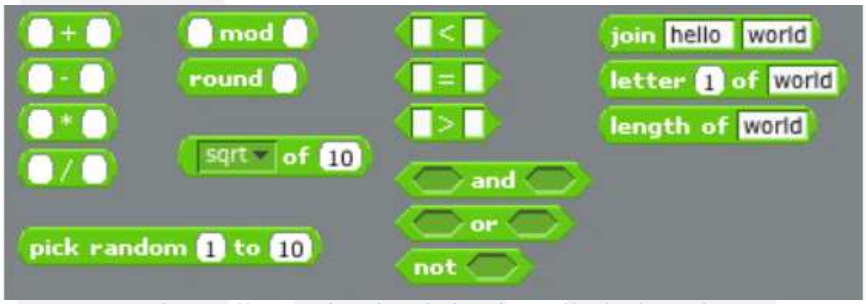
(5) 조건(Conditioanl) : 특정한 상태인지 아닌지를 판단하여 명령을 수행할 것인지 말 것인지를 결정한다. 상호작용하는 프로젝트를 만들 때 핵심적인 기능 중에 하나이다.



https://scratch.mit.edu/projects/97346066/

조건에 따라 참과 거짓이 나뉘며 그에 따라 수행하는 명령이 달라진다. 고차원의 프로젝트를 만들기 위한 핵심이다.

(6) 연산(Operators) : 수와 문자열 처리에 관련된 명령이다. 수학적인 사칙연산, 함수 등을 포함하고 있으며, 논리연산(and, or, not)을 구현할 수 있다.



https://scratch.mit.edu/projects/97346775/

숫자나 문자값을 가지는 블록이 있는가 하면 참과 거짓을 판단하는 블록이 있다.

(7) 데이터(Data) : 변수(variables)와 리스트(Lists)를 포함하는 개념이다. 값을 저장하고 수정 검색하기 위해 필요하다.

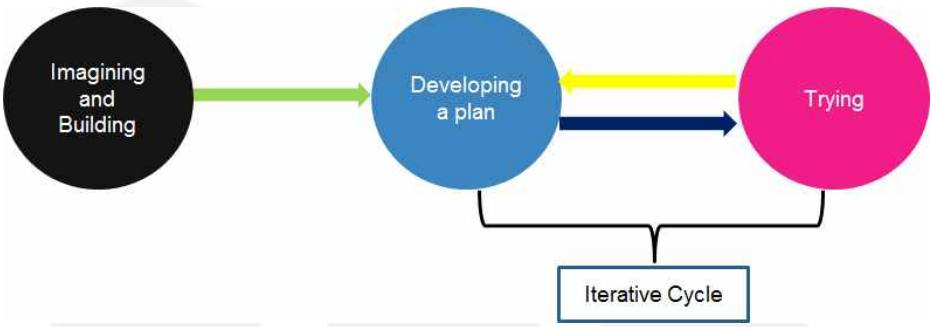
https://scratch.mit.edu/projects/97347524

위에 게임에서 스크래치 사용자는 게임에 점수를 추가하고 싶다는 필요와 욕망에 의해 자연스럽게 변수를 접하고 사용하게 된다.

2. CT 실습(Computational Thinking Practices)

컴퓨팅 실습은 소프트웨어 교육을 통해 익혀야할 기능을 의미한다. 스크래치를 사용해 프로젝트를 만드는 사람들은 제작과정에서 자연스럽게 아래와 같은 기능을 사용하게 된다. 여기에 학생들의 인터뷰영상을 첨부하여 정리하였다.

(1) 실습하기와 반복하기(Being incremental and iterative) : 프로젝트를 설계(design)하는 것은 한 번에 이루어지지 않는다. 아이디어를 구현하고 발전시키고 실행시키고 더 발전시키는 과정을 거치게 된다.



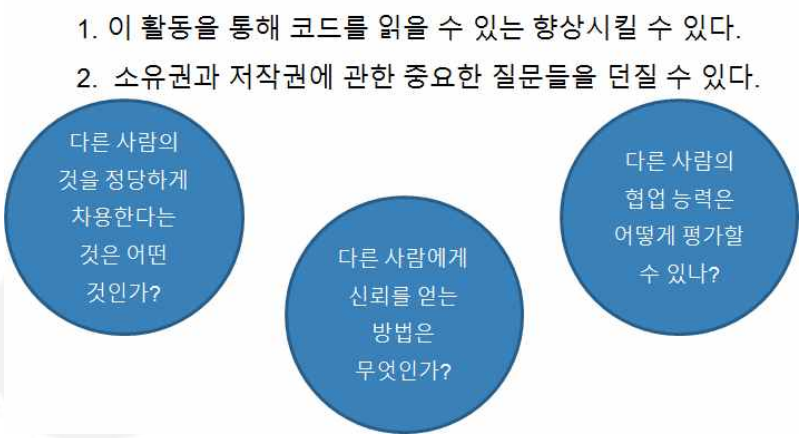
초등학생: https://vimeo.com/106046496
중등학생: https://vimeo.com/106046497
고등학생: https://vimeo.com/106046863

(2) 테스팅하기와 디버깅하기(Testing and debugging) : 원하는 프로젝트를 확실하게 만들기 위해 문제상황을 찾고 수정하는 것이다. 프로젝트 설계자가 문제를 다루고 예측하는 전략을 향상 시켜줄 수 있는 중요한 활동이다.

1. 문제를 인식하라. (identify (the source of) the problem)
2. 자신의 스크립트를 훑어보라.(read through your scripts)
3. 스크립트를 시험해 보라.(experiment with scripts)
4. 스크립트를 다시 작성하라(try writing scripts again)
5. 제대로 작동하는 스크립트의 예를 찾아라.(find example scripts that work)
6. 문제에 대해 말고 있는 다른 사람에게 물어보라. (tell/ask someone else about the problem)
7. 휴식을 취하라.(take a break)

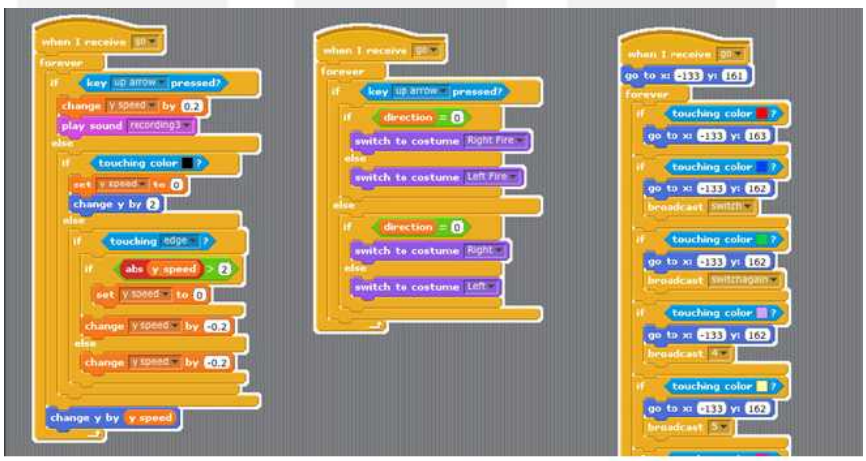
초등학생: https://vimeo.com/106046864
중등학생: https://vimeo.com/106046865
고등학생: https://vimeo.com/106046866

(3) 재사용하기와 재구성하기(Reusing and remixing) : 내가 만든 작품이나 다른 사람이 만든 작품을 이용하여 작품을 만들 수 있다. 이를 통해 설계자 스스로 만들 수 있는 것 보다 높은 수준의 작품을 만들 수 있다.



초등학생: https://vimeo.com/106433596
중등학생: https://vimeo.com/106433595
고등학생: https://vimeo.com/106433597

(4) 추상화하기와 모듈화하기(Abstracting and modularizing) : 각 코드에서 핵심적인 부분을 찾고 이를 이용하여 각 알고리즘 별로 모듈화 하여서 간단하게 나타 낼 수 있다. 이를 통해 전체와 부분의 관계를 파악하고 탐색한다.



스프라이트의 움직임을 제어 | 스프라이트의 모양을 제어 | 장애물과 관련된 이벤트를 제어

초등학생: https://vimeo.com/106437612
중등학생: https://vimeo.com/106437614
고등학생: https://vimeo.com/106438541

3. CT 관점(Computational Thinking Perspectives)

컴퓨팅 관점은 소프트웨어 교육에서 길러주어야 할 태도라고 할 수 있다. 스크래치 사용자들을 인터뷰해 본 결과 스크래치로 프로그래밍을 경험하면서 스스로에 대한 이해, 다른 사람과의 관계, 기술적인 세계에 대한 이해가 높아지는 것을 발견할 수 있었다.

(1) 표현하기(Expressing) : 그림 그리기, 글쓰기와 같이 자신을 표현하는 매체(medium)로서 사용할 수 있다는 것을 인식하게 하는 것이다.

'나는 무엇인가 만들 수 있다.'

인터뷰 내용을 살펴보면 학생들은 창의적으로 새로운 것을 만들어 내어 나를 표현할 수 있다는 점에서 스크래치가 '페이스북'보다 재미있다고 말하기도 한다. 이러한 점을 볼 때 소프트웨어 교육은 컴퓨터를 이용해서 자신을 자유롭게 표현할 수 있는 태도를 길러주는 것이다.

(2) 연결하기(Connecting) : 다른 사람과 함께하고, 다른 사람을 위한 창작의 가치를 깨닫게 하는 것이다.

'나는 다른 사람들과의 교류를 통해 다양한 일들을 할 수 있다.'

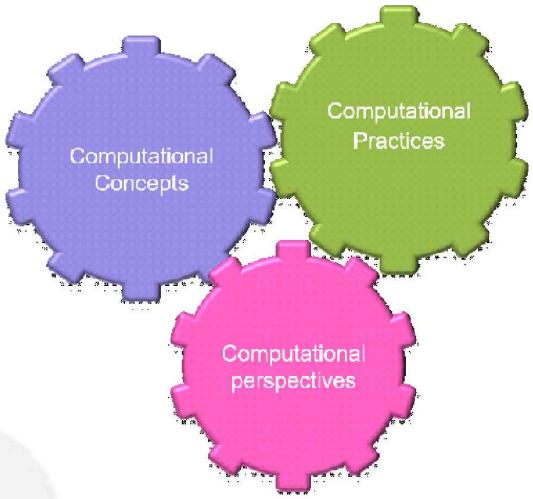
온오프라인을 통해 다른 사용자들과 관계를 맺으며 의사소통을 하고, 다른 사람의 작품을 리믹스하기도 하면서 발전하는 자신을 발견하게 된다. 또한 공공의 이익을 위한 프로젝트를 제작하고 공유하는 것의 가치를 인식하며 미래 사회를 살아가는 데 필요한 사회성을 기르게 된다.

(3) 질문하기(Questioning) : 급변하는 시대에 새로운 기술에 대한 적응력과 자신감을 길러주는 역할을 한다.

'CT에 따라 나는 세상을 이해하기 위한 질문을 할 수 있다.'

급변하는 세상을 이해하기 위한 적절한 질문을 찾고 이에 대해 탐구할 수 있는 자신감과 태도가 길러 질 수 있다.

소프트웨어 교육을 통해 CT를 길러주기 위해서는 단순히 프로그래밍 기능만을 가르쳐서는 안 될 것이다. 교사가 올바른 관점과 철학을 가지고 컴퓨팅 지식과 기능, 태도를 병행해서 지도해야 한다.



[참고사이트]

http://scratched.gse.harvard.edu/ct/

[참고문헌]

경인교대미래인재연구소(2015), 스크래치와 함께 하는 창의컴퓨팅 가이드북, 생능출판사, 파주

K. Brennan, M. Resnick(2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada. 2012(1), 1-25

수업 안에 CT 녹여내기

미래인재연구소 연구원
서 희 정
s.thefirst100@gmail.com

"CT를 가진 창의융합인재 양성"

CT는 SW교육의 핵심 목표이자 미래 세계의 주역으로서 자신과 인류 모두의 삶을 풍요롭게 개척해가는 미래인재 양성 교육의 중심에 자리 잡고 있다. 이를 위해서는 사고력과 문제해결력 중심의 창의인재양성을 위한 코딩 실습의 역량 뿐 만 아니라 고도의 사고력을 이끄는 추상화 과정 등에 집중된 교육과정과 실제 수업이 필요하다.

"그렇다면, 실제 수업을 어떻게 해야 하는가?"

CT에 대한 논의가 활발한 가운데, 미국의 스크래치 팀의 세 가지 측면에서 정의하고 있는 CT의 개념(concepts), 실습(practices), 관점(perspectives)의 내용들을 중심으로 하여 많은 수업 연구와 실제 투입의 과정을 꽤 오래 반복해 왔다. 이를 바탕으로 한 실제 수업에서 학생들의 CT 향상을 위한 교사의 역할부터 실제 교수학습과정안의 예시까지 살펴보자.

1. 교사의 역할

Q1. "창의 컴퓨팅을 시작할 때 학생들을 도와주는 최선의 방법은 무엇인가?"

A1. 자신만의 방법으로 철저히 혼자 탐색하도록 내버려둬야 한다. 교사는 그런 활동과 가르침 사이에서 균형을 잡아줘야 한다.

Q2. “교사로서 무엇을 알아야 하는가?”

A1. 교육자가 모든 것을 알 필요는 없지만, 인지적 안내자로서 학생들에게 질문을 해야 한다. 또한 어려운 문제를 학생들이 스스로 해결할 수 있도록 도움을 줄 수 있는 조력자로서의 역할을 해야 한다.

-시모어 페퍼트(Seymour Papert)

많은 교사들이 고민을 하다가 선택을 하거나 혹은 큰 고민 없이 이전에 컴퓨터 수업을 해오던 방식을 그대로 고수하여 학생들은 교실 앞의 커다란 화면 속 교사의 활동 모습을 보며 ‘따라하기’에 급급한 현실과 쉽게 마주할 수 있다. 이 수업 속에서 학생들이 과연 CT를 키울 수 있는 ‘배움’이 일어날 수 있을까? 그렇게 하기 위해서는 학생들의 사고력을 촉진 할 수 있는 ‘질문’과 체계적이고 계획적인 ‘가르침’, 충분한 시간 속의 ‘탐색’, 그리고 든든한 ‘지원’이 함께 균형을 이루며 존재해야 할 것이다.

2. 무엇을 가르칠 것인가

1) 교수내용의 분석

가. 창의컴퓨팅 문화

구분	내용
지적	CT의 개념 및 실습과 관련된 지적인 부분
물리적	책상, 의자, 컴퓨터 배치를 통한 타인과의 상호작용을 고취
정의적	할 수 있다는 자신감, 새로운 것에 대한 두려움 극복하는 도전자세

세 가지 측면 중, 특히 물리적인 부분에 대한 지원이 시급하다. 상호작용을 통해 서로 평가하고 격려하며 상호 배움이 이뤄지고, 이것이 지적·정의적 측면의 문화를 조성하는 기반을 이룰 수 있기 때문이다.

나. 세부내용

구분	내용
개념	시퀀스, 반복, 이벤트, 병렬처리, 조건, 연산자, 데이터
실습	실험과 반복, 테스팅과 디버깅, 재사용과 재구성, 추상화와 모듈화
관점	표현하기, 연결하기, 질문하기

2) 학습과제의 선정

시작하기, 탐험하기, 애니메이션, 스토리, 게임, 고급기능, 해커톤(팀프로젝트)

3) 학습목표의 설정

가. 예시 목표

단순한 명령의 시퀀스를 사용하여 복잡한 동작을 표현하는 것을 배울 수 있다.

나. 학습 문제

스프라이트를 어떻게 움직이게 할 수 있을까 생각해보자.

기존 교과의 성취 기준과 학습 목표, 학습 문제와 같은 맥락으로 학습 목표나 성취 기준이 체계적으로 확립 된 후, 학생들의 발달 단계와 흥미 유발에 적합한 학습 문제로의 제시가 필요하다.

3. 어떻게 가르칠 것인가

1) 교수학습 과정안 작성

① 5가지 교수학습 모형 중 선택

② 기존 교과의 교수학습 모형 선택도 가능(0단원~1단원)

③ 학생들의 사고력 향상에 주안점을 둔 교사의 발문

④ 협업문화, 재구성과 재사용 문화 ,격려 문화 조성을 바탕으로 활동

⑤ 자기화 할 수 있는 디자인 저널의 적극적 활용

2) 수업자 의도 작성

- 수업 지도안 곳곳에 수업자 의도로 자세하고 구체적으로 생각 공유하기

3. 무엇으로 가르칠 것인가?

1) 교수학습 자료

① 스크래치 스튜디오(포트폴리오)

② 디자인 저널

③ 평가 그룹과의 협력 자료(음성, 비디오, 텍스트파일)

④ 스크래치 카드

⑤ 프레젠테이션 자료(PPT 등)

교수학습 자료 중 특히 중요한 것은 스크래치 스튜디오와 디자인 저널의 계획적이고 꾸준한 활용이다. 재구성과 재사용의 문화와 능력 향상을 위해 스크래치 스튜디오는 개인별, 주제별, 학급별 등 다양하게 활용되어야 한다. 또한 디자인 저널의 형태는 줄 없는 공책부터 온라인 노트 프로그램까지 자유롭게 정해서 활용할 수 있는데, 학생들의 발달단계와 학교의 실제 수업 현장의 물리적 환경 수준에 적합한 방법을 선택해서 활용해야 한다.

4. 어떻게 배움이 일어나게 할 것인가?

1) 창의 컴퓨팅

① 창의적인 잠재력, 상상을 통원한 창의성 발휘 지원하기

② 삶 속에서 즐길 수 있는 역동적이고 상호작용적인 컴퓨터 미디어를 만드는데 필요한 지식, 실습, 기초 리터러시 강조하기

③ 컴퓨터적인 관점에서 삶의 모든 측면을 살펴보며 그 동안 배운 학문과 함께 사용하도록 격려하기

학생들의 창의성을 마음껏 발휘할 수 있는 학습 분위기 조성이 우선되어야 하고 그 위에 다양한 활동들과 충분한 시간이 지원된다면 잠재력을 이끌어 낼 수 있는 배움이 일어날 수 있다.

2) 해커톤

① 계획, 제작, 공유가 기본 뼈대인 팀 프로젝트로 핵심적인 단원

② 프로젝트 준비하기→ 계획하기→ 본격적으로 제작하기→ 프로젝트 검토하기→ 점검하기→ 다시 제작하기→ 인터뷰 그룹→ 프로젝트 점검하기 마지막으로 제작하기→ 발표 준비하기→ 발표하기→ 축하와 격려 주고 받기.

공식적인 발표에서 칭찬과 격려가 무척 인색한 곳 중에 하나로 실제 우리의 교육 현장이 더 이상 포함되어서는 안 된다. 실제 MIT 학회 때, 우리나라에서 초등학교 3~4학년이 만들었던 수준의 프로그램을 고등학생이 굉장히 자랑스럽게 발표하고 학회에 참석한 교수님, 박사님, 연구원 등 모든 이 역시 굉장히 놀라워하고 대단하게 여기며 공유가 이뤄진다. 학생들이 정말 즐길 수 있는 환경인 것이다.

학생들이 이 문화를 탈피하여 서로의 성장과 발전을 위한 조언과 격려를 아끼지 않고, 인터뷰 그룹(상호 평가 모둠)과의 상호 작용을 바탕으로 검토와 수정을 반복한 후 최종 작품을 발표했을 때 그 성취감과 행복감을 만끽할 수 있고, 서로 축하와 격려가 오가 축제를 즐기는 듯 발표 공유회 문화가 자리 잡을 수 있기 위한 수업과 노력이 필요하다.

3) CT(CT)

- 개념, 실습, 관점의 집합으로서의 CT에 대한 정의

5. 교수학습 과정안 예시

1) Unit1. 탐험하기(나에 대하여)

2) 나에 대해서 소개하는 프로그램 만들기

단 원	Unit1. 탐험하기(나에 대하여)	수업장소	컴퓨터실
본시주제	나에 대해서 소개하는 프로그램 만들기	교 과 서	42~43
수업모형	U-M-C 모형(놀이-수정-재구성)	차 시	6/6 (40분)
학습목표	학생들이 개인적으로 관심 있는 아이템을 이용해 나에 대해서 소개하는 프로그램을 만들 수 있다.		
학습문제	나에 대해서 소개하는 프로그램 만들어 봅시다.		

학습단계	학습과정	교수 - 학습활동	시간	자료(▶) 및 유의점(☞)
도입	동기유발	▣ **동기유발 : 선생님의 자기소개**   **T 선생님이 보여 주는 프로젝트를 보며 알게 된 사실을 이야기 하여 봅시다.** - 선생님에 대해서 새롭게 알게 된 사실은 무엇인가요? **T 나를 설명할 수 있는 물건에는 어떤 것이 있나요?** - 나를 특징을 설명하거나 나를 대표할 수 있는 물건에는 어떤 것이 있나요?	5‘	☞교재에 제시된 예제 작품이나 선생님이 제작한 프로젝트를 사용한다. https://scratch.mit.edu/projects/94541542 https://scratch.mit.edu/projects/67643836/ ☞몇몇 학생의 발표를 통해 프로젝트에 대한 이해도를 높인다.
	학습문제 탐색	▣ **학습문제 제시하기** **학습문제** 나에 대해서 소개하는 프로그램 만들어 봅시다.		
Using (놀이)	활동 1	▣ **활동1 : 예시프로젝트 자유탐색** **T 선생님이 공유한 스튜디오에 접속하여 예시 작품을 자유롭게 탐색하여 봅시다.** - 나를 설명하기 위하여 어떠한 스프라이트가 사용되었나요? - 각 스프라이트에는 어떠한 효과가 사용되었나요? - 그러한 효과를 구형하기 위해 어떤 블록들이 사용되었나요?	5‘	☞프로젝트 스튜디오
	활동 2	▣ **활동2 : 기능 익히기** **T 나를 대표할 수 있는 스프라이트를 고르세요.**	5‘	☞스프라이트를 고를 때 시간이

학습 단계	학습 과정	교수 - 학습활동	시간	자료(▶) 및 유의점(☞)
Modify (수정)		- 나와 비슷한 캐릭터를 골라서 스프라이트에 추가하세요. - 스프라이트를 화면 중앙에 위치시켜 주세요. **T 다음과 같은 스프라이트를 클릭했을 때 반응 할 수 있도록 만들어 봅시다.** - 스프라이트와 사용자가 상호작용할 수 있습니다. 		오래 걸리지 않도록 주의한다. ☞예시작품을 리믹스하여 사용하도록 안내한다. https://scratch.mit.edu/projects/94540170/ ☞워크북 43쪽
reCreate	활동 3	**▣ 활동3 : 프로젝트 제작하기** **T 나에 대해서 자랑하고 싶은 내용을 디자인저널에 적어보세요.** - 내가 잘 하는 것에는 어떤 것이 있나요? - 내가 좋아하는 것은 어떠한 것들 인가요? - 내가 좋아하는 음식은 무엇인가요? **T 나를 표현할 수 있는 스프라이트를 골라서 화면에 배치해 주세요.** - 스프라이트 저장소에 있는 이미지를 사용하세요. - 각 스프라이트를 통해서 말하고자 하는 내용은 무엇인가요? **T 스프라이트에 어떠한 효과를 넣을 것인가요?** - 어떠한 효과를 넣고 싶은지 생각하여 보세요. 이때 스크래치 카드를 활용하세요. - 우리가 사용할 블록에는 어떤 것이 있는 지 살펴보세요. **T 각 효과는 어떤 방법으로 실행시킬 것인가요?** - 스프라이트를 실행시킬 때 어떠한 이벤트 블록을 사용할 것인가요?  **T '나에 대하여' 프로젝트를 만들어보세요.**	15 '	☞시간을 고려하여 스크래치에 내장된 이미지를 사용하도록 안내한다. ☞스크래치 카드는 사용할 블록을 고려하여 선택적으로 제시한다.

학습 단계	학습 과정	교수 - 학습활동	시간	자료(▶) 및 유의점(☞)
공유하기	학습정리	**▣ 짝 공유 활동** **T 내가 만든 프로젝트를 짝에게 소개하여 보세요.** - 친구의 프로젝트에서 칭찬할 만 한 점은 어떤 ₩것이 있나요? - 친구에게 배우고 싶은 것은 어떤 것이 있나요? **▣ 학습정리** **T 오늘 활동을 하면서 생각하거나 느낀 점을 디자인 저널에 기록하고 발표해봅시다.** - 직접 블록을 선택해보면서 재미있는 기능이 많다는 것을 알았습니다./ 순서에 잘 맞게 블록을 맞추어야 원하는 대로 스프라이트가 움직인다는 것을 알았습니다./ 내가 바꾸고 싶은 배경으로 무대를 꾸밀 수 있어서 즐거웠습니다.	10‘	
	다음 차시 예고	**T 오늘 활동 중에 어떤 점이 놀라웠나요?** - 배경을 새롭게 바꿀 수 있던 점이 놀라웠습니다./ 스프라이트를 다양하게 만들 수 있던 점이 놀라웠습니다./ 예제를 따라하는 데에 그치지 않고 스스로 바꿀 수 있다는 것이 놀라웠습니다. **T 다음 시간에는 10개의 제한된 블록으로 스크래치 프로그램을 만들어봅시다.**		

♥수업자 의도♥

이 수업에서는 스크래치를 처음 접하는 학생들이 스크래치의 기본적인 인터페이스를 익히고 흥미를 느끼게 하는데 목적이 있다. 그 과정에서 교사가 일방적으로 지도하고 알려주는 것이 아니라 학생들이 자유롭게 탐색해 보고 의견을 나누면서 스스로 발견하게 하는 방법을 사용한다. 하지만 블록을 드래그-앤-드롭해서 서로 연결한다는 핵심적인 인터페이스는 놓치지 않도록 지도해 주어야 한다. 스크래치 카드를 이용해서 간단한 프로젝트를 만들어 보는 것 또한 인터페이스를 익히게 하는 데 목적이 있으며 학생들이 카드를 스스로 선택하게 함으로써 자발적이고 자율적인 활동이라는 인상을 줌으로써 학생들의 참여도와 흥미를 높일 수 있다.

6. CT 실습의 평가 도구

: 스크래치와 함께 하는 창의컴퓨팅 가이드북 참고

실험하기와 반복하기	낮음	보통	높음
어떻게 프로젝트를 개발했는지 순서대로 이야기해 보세요.	프로젝트 개발에 대한 기본적인 설명을 하거나 자세한 설명이 없다.	창작 순서에 따라 프로젝트 개발의 일반적인 예시를 제시한다.	창작 순서에 따라 프로젝트 개발 방법과 프로젝트의 다양한 요소에 대해 자세하게 설명한다.
프로젝트를 진행하면서 얼마나 다양한 시도를 해 보았나요?	시도한 것에 대한 예시를 제시하지 못한다.	프로젝트에서 시도한 것들의 일반적인 예시를 제시한다.	프로젝트에서 시도한 다양한 것들의 구체적인 예시를 제시한다.
어떤 수정 사항들이 있었나요? 그리고 왜 수정하게 되었나요?	수정을 하지 않았거나 예시를 들지 않고 단지 수정했다고만 이야기한다.	프로젝트를 만들면서 1개의 구체적인 수정 사항만 이야기한다.	프로젝트에 추가한 구체적인 것들과 이유를 설명한다.
프로젝트를 만들면서 노력한 점이나 새로운 것들을 시도할 때 사용한 다양한 방법을 이야기해 보세요.	새로운 것을 시도한 예를 제시하지 못한다.	1개의 새로운 것을 시도한 예를 제시한다.	구체적으로 시도한 새로운 것들을 제시한다.

테스팅하기와 디버깅하기	낮음	보통	높음
프로젝트를 실행한 결과가 여러분이 예상했던 것과 무엇이 다른지 설명해 보세요.	예상했던 실행 결과와 달랐던 점을 이야기하지 못한다.	프로젝트가 잘못된 것을 설명하나 원하는 실행 결과가 무엇인지 말하지 못한다.	프로젝트를 실행했을 때 원했던 결과와 발생한 결과를 구체적으로 설명한다.
문제의 원인을 살펴보기 위해 스크립트를 어떻게 분석했는지 이야기해 보세요.	문제를 설명하지 못한다.	스크립트를 보고 해석할 수 있으나 코드에서 문제를 찾는 구체적인 사례를 들지 못한다.	코드에서 문제를 찾는 구체적인 예를 들고 스크립트를 해석할 수 있다.
어떻게 작동되는지 시험하고 수정했는지 이야기해 보세요.	문제와 해결방법이 무엇인지 모른다.	작동되는지 시험하고 수정한 일반적인 예만 들 수 있다.	작동되는지 시험하고 수정한 구체적인 예를 들 수 있다.
문제를 해결하기 위해 고려한 다른 방법에 대해 이야기해 보세요.	문제 해결 방법을 하나도 설명하지 못한다.	문제 해결 방법의 일반적인 예만 들 수 있다.	문제 해결 방법의 구체적인 예를 들 수 있다.

재구성하기와 재사용하기	낮음	보통	높음
다른 프로젝트를 실행해 보고 스크립트를 살펴보면서 아이디어를 얻은 것을 이야기해 보세요.	다른 작품에서 찾은 아이디어를 이야기하지 못한다.	아이디어를 얻은 작품에 대해 개괄적인 설명을 한다.	아이디어를 얻은 작품에 대해 구체적인 설명을 한다.
다른 프로젝트의 일부분을 선택하고 작품에 활용한 방법을 이야기해 보세요.	다른 프로젝트의 스크립트, 아이디어, 자원을 활용한 방법을 설명하지 못한다.	다른 작품으로부터 활용한 스크립트, 아이디어, 자원을 알고 있다.	다른 작품으로부터 활용한 스크립트, 아이디어, 자원에 대해 구체적인 설명을 한다.
기존의 프로젝트를 개선하기 위해 어떻게 수정했나요?	개선한 프로젝트에 대해 이야기하지 못한다.	개선 작품의 수정 내용에 대해 개괄적인 설명을 한다.	구체적으로 개선 작품의 수정한 것을 이야기한다.
작업의 기초가 되거나 아이디어를 얻은 작품의 출처를 어떻게 밝혔나요?	다른 사람에게 출처를 밝히지 못한다.	아이디어를 얻은 사람의 이름을 이야기한다.	스크래치 사이트나 프로젝트에 아이디어를 얻은 사람들을 기록한다.

추상화하기와 모듈화하기	낮음	보통	높음
프로젝트에서 필요한 스프라이트를 정하고 어디에 사용할지 어떻게 결정했나요?	어떻게 스프라이트를 결정했는지 말하지 못한다.	특정한 스프라이트를 선택한 것에 대해 개괄적으로 설명한다.	프로젝트의 목표에 근거해서 어떻게 스프라이트를 선택했는지 구체적으로 설명한다.
프로젝트에 필요한 스크립트를 정하고 만드는 것을 어떻게 결정했나요?	스크립트를 만든 방법을 설명하지 못한다.	특정한 스크립트를 만들기로 결정한 것에 대해 개괄적으로 설명한다.	프로젝트의 목표에 근거해서 결정한 스크립트에 대해 구체적으로 설명한다.
다른 사람들이 이해하기 쉽도록 하기 위해 어떻게 스크립트를 만들었나요?	어떻게 스크립트를 만들었는지 설명하지 못한다.	스크립트를 만든 방법을 개괄적으로 설명한다.	스크립트를 만든 방법과 왜 그렇게 했는지를 구체적으로 설명한다.

CT, 스크래치에서 찾다

미래인재연구소 연구원
최 상 현
chachoi83@nate.com

최근 초등학교 SW교육에서 EPL(Educational Programming Language)로 스크래치(scratch)를 많이 사용하고 있다. 그 이유는 쉽게 배울 수 있고 직관적으로 조작할 수 때문이다. 학생들은 스크래치를 통해서 애니메이션, 스토리, 게임 등 여러 가지 프로젝트를 쉽게 만들 수가 있다. 그리고 이를 통해 SW에 즐거움을 느끼고 궁극적으로는 CT를 배양 시킬 수가 있다.

1. CT의 필요성

그렇다면 CT는 무엇일까?

Wing(2006)은 CT에 대해 '컴퓨터과학의 기초개념을 토대로 문제해결, 시스템 설계, 인간행동의 이해를 내포한다'고 하였으며 아이들의 분석능력을 키우기 위해 CT를 읽고, 쓰고, 셈하기에 추가해야 한다'고 하였다.[4] 아이들 스스로 자신이 머릿속에 가지고 있던 아이디어들을 직접 소프트웨어를 통해 구현해 보고 시행착오를 겪으며 논리적 사고를 자연스레 기르게 된다. 또한 정답만을 배우는 교육에서 벗어나 다양한 방법으로 문제를 해결해 나가려는 열린 생각을 갖게 되어 문제해결력, 창의성 증진에도 많은 도움이 된다.

이러한 CT를 기르기 위해 가장 중요한 것은 SW를 직접 만들어 보는 경험이다. 이러한 경험을 통하여 아이들은 자신의 생각을 표현하며 다양한 시도를 하게 되고, 그 결과 우리가 상상하지 못한 다양한 결과물을 만들어 낼 수 도 있다. 결과적으로 기존의 누군가 이미 만들어 놓은 것을 소비하는 소비자에서, 아이들에게

4) Jeanette Wing(2006)은 컴퓨터과학의 기초개념을 토대로 문제해결, 시스템 설계, 인간행동의 이해를 내포함

자신이 생각하는 대로 자유롭게 만들어보면서 생산자로서의 경험을 가능하게 하는 것이다.

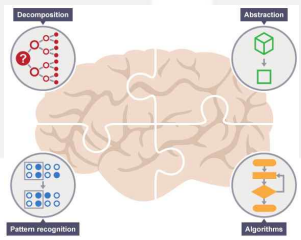
2. CT의 구성요소

일반적인 CT의 구성요소와 그 정의를 보면 다음과 같다.[5)]

구성요소		정의
자료 수집		· 문제해결에 필요한 자료를 모으기
자료 분석		· 자료의 이해, 패턴 찾기, 결론을 도출하기
구조화		· 문제를 그래프, 차트, 그림 등으로 시각화하기
추상화	분해	· 문제를 관리 가능한 수준의 작은 문제로 나누기
	모델링	· 문제 해결을 위한 핵심요소를 추출하고, 모델 만들기
	알고리즘	· 문제를 해결하기 위한 일련의 단계를 알고리즘으로 표현하기 (절차적 표현)
자동화	코딩	· 프로그래밍 언어를 이용해, 문제해결과정을 자동화하기
	시뮬레이션	· 프로그램(소프트웨어)을 실행하기
일반화		· 문제해결과정을 다른 문제에 적용하기

3. CT의 구성요소 재정의

CT와 프로그래밍을 서로 구분하여 다른 영역으로 제시한 영국의 key stage 3 Computer science와 Google for educators에 재정의된 CT는 다음과 같다.

구분	KS3 용어[6)]		Google 용어[7)]
분해(D)	Decomposition		Decomposition
패턴 인식(P)	Pattern recognition		Pattern recognition
추상화(A)	Abstraction		Abstraction
알고리즘(A)	Algorithms		Algorithm design
프로그래밍(P)			

4. CT와 스크래치

5) CT의 구성요소(교육부.KERIS, 2015)
6) 영국 key stage 3 Computer science. CT와 Programming을 구분하여 다른 영역으로 제시함.
7) Google for educators. https://goo.gl/iuslIi

CT는 컴퓨터를 통해 구현될 수 있는 방법으로 문제들을 해결하기 위한 접근이다. 학습자들은 단지 도구의 사용자가 아닌 도구의 개발자가 되기 위해 실제 또는 가상의 인공물(artifacts)들을 생각하거나 자료를 분석 처리해야 한다. 이를 위해 추상화, 재귀, 반복과 같은 개념들을 사용한다.(Barr 2011)

현실과 달리 미래의 문제는 융합적인 관점에서의 접근이 필요하므로 이를 해결하기 위한 융합적 사고가 필요하다. 프로그래밍은 코딩 과정에서 인간의 절차적 사고를 바탕으로 직렬적, 병렬적 사고력, 예측 능력 등의 다양한 사고력을 사용한다. 스크래치는 단순히 그대로 따라 하기 식의 프로그래밍 언어 교육(EPL)이 아니라 학생들이 프로젝트를 계획하고 만들고 수정하고 공유하는 과정을 통해서 혁신적인 해결책을 찾도록 하여 창의적인 사고를 고취시킬 수 있는 프로그램이다. 이에 따라 스크래치 프로그램에서 CT의 구성요소를 찾을 수가 있다.

예를 들면 스크래치에서 정사각형을 만들고 정다각형을 예측하여 설계하는 과정을 통해 CT의 구성요소인 패턴인식과 추상화가 가능하다. 그리고 스크래치의 반복문을 활용하여 효율적인 설계를 함으로써 자동화가 가능하다. 또한 스크래치 프로그램 설계 및 디버깅을 통해 전체적인 CT 신장이 가능하다.

그렇기 때문에 스크래치는 미래 사회에 필요한 능력(=CT)을 키우기 위한 교육 프로그램으로 적합하다고 볼 수가 있다.

5. 스크래치에서 활용 가능한 교수학습 모형

스크래치에서 활용 가능한 교수학습 모형은 다양하다. 대표적으로 사용될 수 있는 모형을 소개한다[8].

8) SW교육 교수학습 모형 개발 연구(한국교육개발원.KERIS, 2015)

1) DMM 모형

단계명	주요 방법	세부 교수학습 내용
Demonstration (시연)	설명 시범보이기 예시	가르치려고 하는 핵심전략과 기능을 교사가 설명하거나 시연을 통해 학생들에게 소개한다.
Modeling (모방)	따라하기 질문,답변	교사의 시연내용을 학생들이 그대로 따라 실습한다. 실습의 과정에서 질문을 통해 학습자들이 교사들의 시연을 모방(modeling)한다.
Making (제작)	만들기 반복활동	시연과 모방의 단계에서 배운 내용을 토대로 학생이 직접 만들어보는 활동을 한다. 반복적으로 진행하되 단계적, 전체적인 활동을 학습자들이 전개한다.

2) UMC(재구성중심모형)

단계명	주요 학습방법	세부 교수학습 내용
Using (놀이)	조작, 체험, 놀이, 활용, 탐색	학습 내용이 담긴 프로젝트를 시연해 보거나 조작해 보면서 프로젝트를 이해하는 단계이다 즉, 먼저 결과물을 가지고 놀아보며 친숙해지도록 한다. 혹은 직접 교수법을 이용하여 교사의 시법을 따라 간단한 프로젝트를 제작해가며 작동시켜 보도록 한다.
Modify (수정)	추가설계, 수정, 확장, 보완	간단히 제공된 프로젝트에 아이디어를 추가하거나 내용을 확장하여 설계한다. (새로운 스프라이트 추가 및 수정, 변수 추가, 스테이지 확장 등)
reCreate (재구성)	재구성, 구현, 개발, 산출	학습한 기능이나 내용을 활용하여 자신만의 확장된 프로그램을 설계하여 제작해 본다.

3) DDD 모형(개발중심모형)

교육방법	개발중심모형		
탐구학습법	탐구	설계	개발

4) NDIS 모형(디자인중심모형)

교육방법	디자인중심모형			
프로젝트학습법	요구분석	디자인	구현	공유

6. 스크래치를 활용한 CT수업 지도안 예시

: 본 지도안은 한국교육학술정보원의 보고서(2015). SW교육 교수학습 모형 개발 연구에서 발췌함

장애물 피하기 게임 만들기

적용학년	3학년~6학년
과목 및 차시	창의적 체험활동, 실과 (3차시)
수업모형	U-M-C
수업전략	디버깅
도구 및 자료	컴퓨터, 스크래치나 엔트리
중점 CT요소	■분해 ■패턴인식 ■추상화 ■알고리즘 ■자동화

1. 개요

장애물 피하기 게임에 U-M-C 모형을 적용하기 위해서는
먼저 학습자가 장애물 피하기 게임을 실제 해 보는 과정에서 게임에 들어있는 각종 게임 요소(독수리, 장애물 등)를 파악하여 그들의 관계를 나열하고 게임 요소(독수리나 장애물의 움직임)가 가지고 있는 패턴을 파악하도록 하는 활동이 선행되어야 합니다. 다음 단계로, 교사는 의도적으로 프로그램의 일부분을 변경하여 학생들에게 제공하고 학생들은 교사와 함께 변경된 부분이 장애물 피하기 게임에 어떤 영향을 미쳤는지를 탐색해 보도록 합니다. 예를 들어, 독수리의 움직임 축을 X에서 Y로 변경하여 제공해 주는 활동을 통해 프로그램 요소들 간의 관계를 보다 명확히 파악할 수 있으며, 이동 기준축의 변경으로 인해 발생할 수 있는 여러 부가적인 예상하고 해결해 볼 수 있습니다. 마지막 단계로 학생이 중심이 되어 장애물 피하기 게임을 변경해 보는 활동을 하게 됩니다. 변경(해결)하고자 하는 여러 요소들을 학생이 스스로 결정하고 문제를 해결하는 과정을 통해 학생은 CT의 여러 요소들을 통합적으로 경험할 수 있습니다.

2. 학습 목표

○ 다양한 요소를 수정하여 나만의 장애물 피하기 게임을 만들 수 있다.

3. 수업 단계 및 활동

Using (놀이)
- 완성된 장애물 피하기 게임을 학생이 직접 놀아보게 한다 - 이 단계에서 학생들이 게임의 구성요소를 찾고, 게임의 구성요소들이 갖고 있는 특징과 관계, 사건 속에서 패턴을 찾아보도록 한다.

분해 활동	- 장애물 피하기 게임에 필요한 요소를 분해도로 표현하기 (독수리, 장애물)
패턴인식 활동	- 독수리의 움직이는 규칙을 찾아보기 (위, 아래로 움직인다.) - 장애물의 움직임 생각하기 (장애물은 오른쪽에서 왼쪽으로 움직인다.)

Modification (수정) - 교사중심
- 교사가 의도적으로 오류를 제시하여 학생들이 수정하거나 불완전한 소스를 제공하여 학생들이 채워간다.(디버깅) - 프로그램 내에서 핵심이 되는 모듈을 학생들이 찾아낼 수 있도록 한다.

추상화 활동	- 독수리가 움직이는 방향축을 Y축 기준에서 X축 기준으로 변경하여 제공하고 학생들이 오류를 해결하는 과정에서 핵심적인 개념 발견하기 - 움직이는 기준 축이 바뀜으로서 새롭게 발생할 수 있는 문제 예상하기
알고리즘 활동	- X축 블록을 Y축 블록으로 변경하기 - 독수리가 화면의 경계를 넘어가지 않도록 독수리가 이동할 수 있는 Y축 좌표값 제한하기

reCreating(재구성) - 학생중심
- 학생 스스로 변경할 수 있는 요소가 무엇인지 파악한다. - 파악한 요소를 변경함으로서 발생하는 부수적인 문제점들을 해결한다.

분해, 추상화, 알고리즘, 프로그래밍 활동	- 변경하고자 하는 부분 나열하기 (요소, 환경, 규칙 등) 예) 독수리의 속도, 독수리의 크기, 장애물의 크기, 장애물의 종류 등 - 해당 부분에 관련된 핵심 코드 찾기 예) 독수리와 장애물의 속도를 결정하는 변수 찾기 예) 독수리의 크기를 변경하는 코드 찾기 - 변경 의도대로 코드 수정 및 추가하기 예) 독수리 속도 변수 값 변경하기 예) 코드나 스프라이트 변경을 통해 독수리의 크기 변경하기 - 테스트 및 시뮬레이션을 통해 디버깅하기

4. 평가 계획

연번	평가 기준	방법
1	장애물 피하기 게임에 필요한 요소를 말할 수 있는가?	발문
2	제시된 오류를 수정할 수 있는가?	수행평가
3	바꾸고 싶은 요소를 찾아 변경해야 할 코드를 발견할 수 있는가?	관찰
4	테스트와 디버깅을 거쳐 최종작품을 만들 수 있는가?	수행평가

[참고문헌]

교육부, KERIS (2015). 컴퓨팅 사고력의 구성요소

한국교육개발원, KERIS (2015). SW교육 교수학습 모형 개발 연구

Wing, J. M. (2006). *Computational Thinking. Communications of the ACM, 49(3),* 33-35

Bar, D. Harrison, J. & Conery, L. (2011). Computational thinking: A Digital Age. ISTE

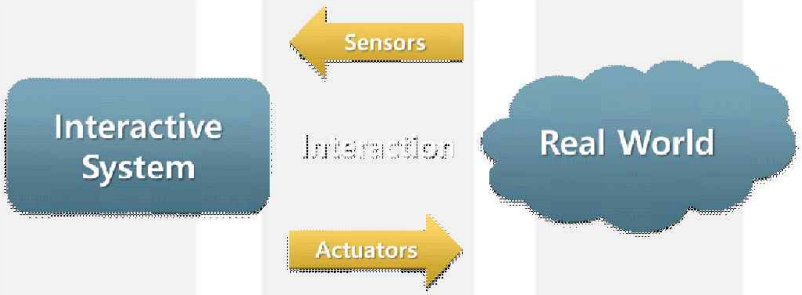
한국교육학술정보원(2015), SW교육 교수학습 모형 개발 연구 보고서

피지컬 컴퓨팅에서의 CT

미래인재연구소 연구원
안 준 별
thekhan83@gmail.com

1. 피지컬 컴퓨팅이란?

최근 피지컬 컴퓨팅(Pysical computing)과 CT에 대한 관심이 커지고 있다. 교육에 대한 이야기를 하기 전에 피지컬 컴퓨팅의 개념이 만들어지게 된 배경에 대해 살펴 볼 필요가 있다.



[피지컬 컴퓨팅의 개념도]

피지컬 컴퓨팅은 보통 센서와 엑츄에이터를 이용해서 아날로그의 현실세계와 디지털 세계가 서로 상호작용(interaction) 하는 시스템을 말한다. 다소 공학적인 개념이기 때문에 우리는 흔히 피지컬 컴퓨팅이 공학과 관련이 깊다고 생각하지만 사실 그 시작은 컴퓨터 공학자들이 아닌 인터랙티브 아티스트들이었다.

인터렉티브 아트는 기존의 작가 중심의 작품이 아닌 감상자가 스스로 작품을 만들어 재창조 해나가는 예술 부분이다. 그러기 위해서는 관객이 바뀔때마다

일관된 경험을 제공해 주어야 할 필요가 있었다. 인터렉티브 예술가들은 이를 위해 점차 다양한 디지털 기술들을 이용하기 시작했다. 하지만 작은 LED 나 간단한 센서들을 프로그래밍을 통해 제어한다는 것은 매우 어렵고 복잡한 일이었다. 이런 작업을 하기위해서는 임베디드 시스템(embedded system)이라는 컴퓨터 공학의 전문분야를 알아야 했기 때문이다. 이에 인터랙티브 예술가들은 본인의 생각을 자유롭게 실현하기 위해 임베디드 시스템을 보다 쉽고 비전공자들도 쉽게 다룰 수 있게 변모시켜 나갔다. 전자-전기적인 문턱과 프로그래밍에 대한 기술적 제약을 낮추기 시작했고, 그 결과 이탈리아의 마시모 반지 교수에 의해 아두이노가 개발되었다. 매우 값이 싸고 진입 문턱이 낮은 아두이노는 컴퓨터 비전문가들도 쉽게 피지컬 컴퓨팅을 다룰 수 있게 해주었다. 무엇보다 놀라운 일은 비전공자들도 약간의 노력만 들이면 프로그래밍을 통해 여러 센서나 엑츄에이터들을 조작할 수 있다는 점이다.

메이커 페어. 피지컬컴퓨팅의 발전은 메이커들에게 새로운 세상을 선사했다.

이런 피지컬 컴퓨팅은 예술을 넘어 메이커들에게 빠르게 확산이 되기 시작했다. 아두이노를 필두로한 피지컬 컴퓨팅은 만들기를 좋아하는 메이커들에게는 상상하는 무엇이라도 손쉽게 만들어 낼 수 있게 해주는 놀라운 도구였다. 아이디어와 약간의 전자적 지식, 간단한 코드 몇줄이면 무엇이든 만들게 해줬다. 거리에 따라 바람의 세기를 조절해주는 선풍기부터 집안의 온도를 실시간으로 모니터링하여 최적의 난방을 해주는 조절 시스템까지 이른바 '스마트'라고 불리우는 다양한 도구들을 저렴한 가격에 쉽게 만들어 나가게 해주었다. 때때로는 이런 실생활이나 실용적인 부분에 더해 드론이나 로봇까지 구현하여 재미난 프로젝트들을 진행해나가기도 한다. 피지컬 컴퓨팅의 대중적 접근은 D.I.Y(Do It Yourself)로 대표되는 메이커 문화를 통해 더욱 확산되고 있는 실정이다.

최근에는 단순히 예술이나 취미를 넘어 경제적 접근으로까지 이어졌다. 피지컬 컴퓨팅을 통해 누구나 아이디어만 있으면 쉽게 시제품을 만들 수 있게 되었

고, 이러한 시도는 물건을 만드는데 있어서 사회적, 시간적 제약을 크게 줄여주어 적은 인력만으로도 창조적 생산활동이 가능하게 되었다.

2. 피지컬 컴퓨팅과 CT

앞서서 살펴본 것처럼 피지컬 컴퓨팅은 전문 분야에서 일반으로 점차 확산되어 나가고 있다. 그렇다면 이런 피지컬 컴퓨팅은 CT와 어떤 관련성이 있을까?

CT에 대한 정의는 여러가지가 있을 수 있지만 기본적으로 자동화와 추상화를 바탕으로 컴퓨터과학의 기본 개념, 원리에 따른 문제 해결, 시스템 설계, 인간 행동의 이해를 포함하는 사고 능력이라 할 수 있다. 이는 다양한 컴퓨팅 장치를 이용한 자동화 수행을 통해 강화될 수 있다. 다양한 문제상황을 컴퓨팅 파워를 통해 해결해 나가는 경험을 통해 CT가 길러질 수 있다. 때문에 CT 에는 문제해결에 적합한 추상적 개념을 선택하고 구성하기 위한 추상화 능력과 추상화 과정을 통해 만들어진 문제해결방법을 자동화하기 위해 적합한 컴퓨팅 도구를 선택하고 사용할 수 있는 능력이 포함된다. 이러한 CT능력 향상을 위해서는 실제 세계의 모습을 컴퓨터가 이해하거나 반응할 수 있도록 물리적인 표현의 폭을 넓혀 다양한 문제를 해결해 가는 경험이 매우 중요하다. 피지컬 컴퓨팅은 그 자체가 현실세계와 디지털 세계와의 상호작용을 위한 고도의 융합 활동이기 때문에 CT와 깊은 연관이 있다고 볼 수 있다.

본 필자는 피지컬 컴퓨팅을 통한 CT 교육 접근 방법을 크게 세 가지로 나누어 제시하려 한다.

먼저, CT교육을 위한 접근을 들 수 있다. 이런 접근방식은 학습자의 사고력을 향상시키기 위해 피지컬 컴퓨팅의 상호작용이 바탕이 되는 로봇이나 키트를 활용한다. 대표적인 예로 아동의 사고력 교육을 위해 로고 프로그램을 만든 시모어 페퍼트의 로고터틀(logo turtle)을 들 수 있을것이다. 시모어 페퍼트

(Seymour papers)는 그의 책 '마인드스톰'에서 다음과 같은 이야기를 한다.

> *"어린이가 프랑스어를 배우려면 프랑스에 살면 자연스럽게 되는 것처럼 어린이가 수학을 배우려면 수학나라에 살 수 있도록 해주는 것이 수학을 자연스럽게 배울 수 있도록 도와주는 것이다. 그리고 컴퓨터가 바로 수학을 언어로 상호작용하면서 대화를 나눌 수 있는 매체이다."*[9]

초창기의 터틀머신. 페퍼트는 사고력 교육을 위해 로고 프로그래밍을 아동들에게 가르쳤다.

페퍼트는 로고 프로그램을 적용하기 위해 모니터에서 벗어나 로고 터틀(logo turtle)이라는 거북모양의 로봇을 이용하였다. 로고터틀은 아동들의 명령을 받아 여러 가지 도형들을 그려나갈 수 있는 피지컬 컴퓨팅 도구이다. 아동들은 명령을 내리는 과정에서 자연스럽게 로고터틀과 상호작용을 하게 되는데 이때 길러지는 사고력은 순차, 반복, 분기, 조건, 자동화 등 프로그래밍과 관련된 CT 사고력과 밀접한 관계가 있다.

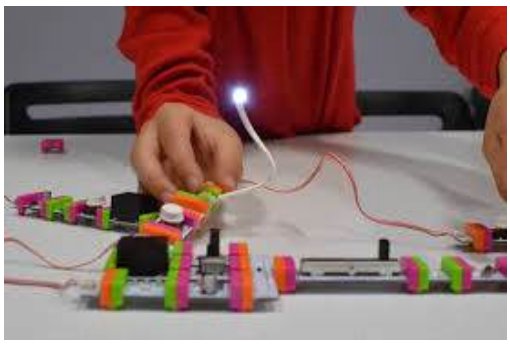
최근에는 로고터틀 이외에도 사고력 교육 측면의 피지컬 컴퓨팅 교구들이 개발되고 있다. 대표적인 예로 비봇과 리틀비츠가 있다. 비봇은 유-초저 수준에 적합한 학습 도구로 전-후 직진과 좌-우 90도 회전, 실행키 등을 가지고 있는 손바닥만한 귀여운 벌모양의 로봇이다. 이동할 위치로 미리 방향키를 누른 후, 실행키를 누르면 기억된 명령에 따라 이동한다. 순차적 사고를 익힐 수 있으며 바닥판을 이용하면 다양한 활동이 가능하다.

비봇-어린이로고. 터틀머신의 축소판이라 할 수 있다. 유치원-초저에서 사용되고 있다.

리틀비츠는 크게 센서, 엑츄에이터, 전원, 연결

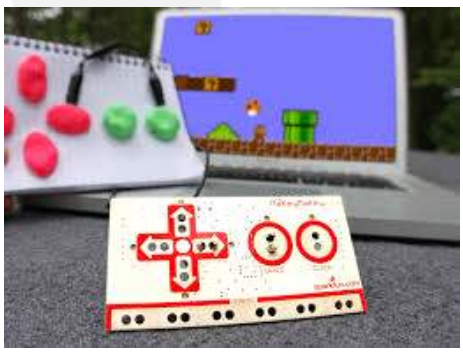
9) 최승준. 2015.12. https://www.facebook.com/seungjoon.choi/posts/10207336029094513

부로 이루어져 있으며, 작은 블록들에 자석이 붙어 있어 손쉽게 연결이 가능하다. 전원-센서-엑츄에이터 순으로 연결하면 바로 작동하며, 복잡한 프로그래밍이 전혀 필요없어 피지컬 컴퓨팅 요소 학습과 기본적인 CT 사고력을 기를 수 있다. 또 간단한 메이커(maker)활동이나 시제품 제작이 가능하여 누구나 쉽고 재미있는 창의적인 프로젝트도 가능하다. 이러한 사고력 교육에 초점이 된 피지컬 컴퓨팅은 도구는 공통적으로 별도의 코딩이 필요 없고, 누구나 쉽고 간단하게 피지컬 컴퓨팅을 경험할 수 있도록 직관적이고 쉽다. 보통 순차, 반복 같은 간단한 CT를 학습할 수 있으며, 피지컬 컴퓨팅의 구성에 대해 학습하기 용이하다고 할 수 있다. 단점으로는 쉽고 간단하기 때문에 복잡한 명령을 내리는데 한계가 있다. 어린 학생들이나 SW 교육을 위한 도입단계에 주로 적합하다.



리틀비츠. 정말 쉽고 간단하다. 피지컬 컴퓨팅을 위한 기술적 문턱을 바닥까지 낮춰 누구나 쉽게 피지컬컴퓨팅을 경험할 수 있다.

다음으로, 예술과 디자인프로세스를 기반한 다양한 창의적 활동을 통한 접근이다. 이 접근 방식의 핵심은 피지컬 컴퓨팅이 가지고 있는 강력한 동기유발을 활용하는 것이다. 피지컬 컴퓨팅을 통해 우리는 눈에 보이지 않는 머릿속 생각을 간단한 프로그래밍을 통해 손으로 잡을 수 있는(tangible) 구체물을 직접 조작할 수 있다. 의도에 따라 즉각적으로 반응하는 구체물을 다룬다는 점은 경험하는 사람에게 매우 매력적이다. 프로그래밍을 통해 간단히 LED 를 깜빡이게 한다던가 모터를 돌려보는것 만으로도 학습자들은 신기해하며 몰입한다. 내 의도에 따라 사물에 명령을 내릴 수 있는 행위만으로도 정말 즐거워한다. 여기에 더해 피지컬 컴퓨팅 도구를 이용하여 간단한 예술 프로젝트를 진행하면 어떨까? 대표적인 피지컬 컴퓨팅 도구인 메이키-메이키(makey-makey)의 경우 인터페



Makey Makey 인터페이스의 확장을 통해 쉽고 재미난 활동이 가능하다.

이스 장치를 확장시켜 줄 뿐이지만 다양하고 재미난 활동 등이 쉽고 간단하게 가능하다. 블록형 프로그래밍언어로 간단한 악기를 만들기 활동을 한다고 했을 때 키보드를 활용하여 연주를 하는 것과 메이키-메이키를 이용하여 과일로 악기를 만드는 것 중 어느 것이 더욱 몰입도 있는 활동이 될 수 있을까?

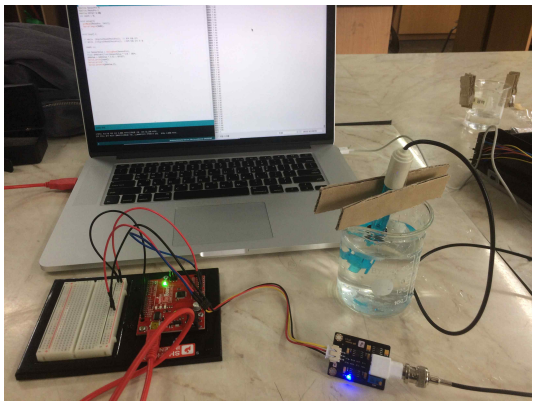
사실 디지털로 아날로그를 제어하는 과정은 매우 복잡하고 어렵다. 하지만 피지컬 컴퓨팅의 매력 앞에서는 학습자들은 다소의 어려움을 쉽게 넘어선다. 코딩은 프로그래밍에 그칠 수 있는 반면 피지컬 컴퓨팅은 센서와 엑츄에이터를 통해 실제 세계(real world)에 간여할 수 있기 때문에 좀 더 실제적이다.

학습자들에게는 간단한 기계에 자신이 영향력을 끼치는 행위 자체가 놀라운 경험이고 행복이다. 피지컬 컴퓨팅의 이 같은 장점은 학습자가 자신의 주변의 문제 해결을 위해 관념적으로 접근하는 것이 아닌 물리력을 동반한 실제적인 해결을 할 수 있는 가능성을 열어준다. 코드의 어려움, 전자-전기적 지식의 어려움은 그들에게 장애물이 될 수 없고, 목표를 위해 끊임없이 몰입하게 한다. 작게는 코딩을 통해 엑츄에이터가 어떻게 조작이 되는지 확인하고 수정해보며 센서와 엑츄에이터의 기본적인 사용법을 익혀 나간다.

여기에 더해 작은 프로젝트를 진행하여 실제 세계에 물리력을 실현 시킬 수 있는 시제품이나 모형을 설계(design)하고, 만들어 적용해 보고, 개선한다. 일련의 순환적 과정을 통해 학습자들에게 남다른 성취감과 실제적 경험을 줄 수 있으며, 이런 과정 속에서 재사용(reuse), 재구성(remix), 테스트와 디버깅 등의 CT(practice)를 기를 수 있다. 다양한 피지컬 컴퓨팅 도구들을 통해 자신의 수준에 맞는 기술적 제약을 극복해 나갈 수 있게 도우며, 누구나 즐겁게 자신의 아이디어를 실현시키고 뽐낼 수 있게 도와 줄 수 있다.

마지막으로 컴퓨팅 파워를 이용한 다양한 데이터 수집과 분석을 위한 접근을 들 수 있다. 이 부분은 앞의 두 가지 접근 방식보다 보다 CT에 본질적으로 가깝다고 할 수 있다. 우리는 복잡한 문제를 해결하기 위해 현실 세계를 분석해 나가야 할 필요가 있다. 외부 세계의 일을 컴퓨터를 통해 문제를 해결해 나가기 위해서는 아날로그 세상을 디지털로 변환하여 살펴볼 필요가 있다. 이러한

과정은 피지컬 컴퓨팅을 통해 이뤄지고 여기서 수집된 디지털 데이터들을 정보적인 방법으로 분석-해석함으로써 인간의 인식보다 정밀하고 객관적으로 세상을 바라 볼 수 있다.



실시간 ph계측장치. 간단한 구성이지만 구현하기 위해서는 생각보다 많은 시행착오가 있었다.

간단한 예를 하나 들어보자. 최근에 필자는 학생과 과학 보고서를 준비하고 있다. 학생은 용액의 ph 변화를 분석하기 위해 스포이드와 리트머스 시험지를 이용하였다. 스포이드 몇방울을 넣고 시험지를 색변화를 관찰하는 방법으로 말이다. 하지만 문제가 생겼다. 리트머스 시험지 색의 변화는 사람의 감각기관을 이용하기 때문에 다소 주관적이며 수치화하기 어려웠다. 그리고 스포이드 방울을 일일이 몇 방울씩 떨어트리는 방법은 매우 비효율적이었다. 이에 필자는 학생과 ph센서와 포토인터럽터를 이용하여 실시간 ph 변화 계측 장치를 만들어 보았다. 뷰렛을 이용해 자동으로 용액을 떨어트리고 포토인터럽터로 실시간으로 방울 수를 카운트하게 했다. 이와 동시에 방울을 카운트 할 때 마다 ph센서로 ph를 계측하게 했다. 결과는 매우 놀라웠다. 약 2000방울 정도 떨어질 때마다 실시간으로 계측이 되었고, 덕분에 그래프도 만들어 보다 확실한 분석이 가능했다. CT 문제해결 과정이 이런 것이 아닐까?

이러한 접근은 CT 문제해결 뿐만 아니라 추상화와 자동화에 대해 보다 깊은 이해와 통찰을 가져올 수 있다. 센서와 엑츄에이터의 작동을 하기위한 핵심 코드를 살펴보면 실제 세상의 것을 어떻게 추상화하여 나타내는지 쉽게 알 수 있다. 예를 들어 DC 모터와 서보(servo) 모터의 경우 작동방법에 따라 서로 다른 방법으로 제어된다. DC 모터의 경우 전류의 세기를 제어하지만 서보 모터의 경우는 각도를 제어한다. 모든 피지컬 컴퓨팅에서는 센서와 엑츄에이터는 측정하고자 하는 값과 제어하는 값을 모두 전기적 신호로 제어하게 된다. 센서와 엑츄에이터에 따라 제공되는 각각의 데이터 시트를 분석하여 수식으로 처리하여 코드에 반영한다. 추상화에 대해 이런저런 해석이 많지만 이러한 추상화가

좀 더 Computational 한 추상화가 아닐까 싶다. 이러한 과정은 다소 어려울 수도 있지만 보다 본질적인 CT 전반을 경험할 수 있을 것이다.

3. 앞으로의 과제

이상으로 피지컬 컴퓨팅과 CT에 대해 생각해 보았다. 피지컬 컴퓨팅은 CT 를 위한 학습도구로서 굉장히 매력적이다. 하지만 자칫 학습도구 자체에 함몰되어 SW 교육 본질을 잃을 수 있다. 아직까지 피지컬 컴퓨팅 학습도구의 체험이나 메이커나 발명과 같이 단순히 산출물을 만들어 내는 것만 목적을 둔 활동이 많다. 이런 활동들은 쉽게 흥미나 관심을 불러일으킬 수 있겠지만 그만큼 고등사고력을 길러주기는 역부족일 수 있다. 피지컬 컴퓨팅을 통해 CT적인 문제해결 과정과 아날로그 현상을 추상화하기 위해서는 본질에 대한 깊은 사고가 필수적이다. 자신의 생각이나 아이디어를 실현하기 위해서는 때론 다소 복잡한 조건을 가진 알고리즘을 구현하는 경험도 필요하다. 피지컬 컴퓨팅을 활용한 CT 향상을 위해서는 보다 확실한 전략을 세워 수업 설계를 할 필요가 있다. 이제 피지컬 컴퓨팅의 확산과 활용에서 더 나아가야 한다. 피지컬 컴퓨팅 도구의 단순 체험에서 벗어나 디지털과 아날로그를 아우르는 진정한 융합 활동이 될 있도록 많은 교수 전략과 학습 프로그램에 대한 연구가 필요한 시점이다.

CT교수학습방법 적용의 실제

미래인재연구소 연구원
윤 종 원
bbengjjuem@naver.com

1. 들어가며

구글에서는 CT 신장을 위하여 분해, 패턴인식, 추상화, 알고리즘(프로그래밍)의 4단계 모듈 전략을 제시하고 있다. 필자는 이 4단계 모듈을 적용한 CT요소 중심의 교수학습방법이 적용된 수업을 소개하고자 한다. CT요소 중심의 교수학습방법은 반드시 분해-패턴인식-추상화-알고리즘-프로그래밍의 절차적 단계를 밟아야 하는 것은 아니다. 상황에 따라서는 패턴인식-추상화-분해-알고리즘-프로그래밍, 패턴인식-추상화-분해-알고리즘-프로그래밍 등의 다양한 전략으로 지도가 가능하다.

2. 적용한 수업 모형

DPAAP 모형 인용-케리스 보고서

3. 컴퓨팅 융합과 STEAM교육

STEAM수업 준거틀을 3단계로 제시하여 구분하면서 동시에 컴퓨팅 융합을 하는 수업 예시로 개선

4. 수업의 실제

나만의 스마트 스프링클러 만들기

적용학년	5학년~6학년
과목 및 차시	창의적 체험활동
수업모형	D-P-A-A-P
수업전략	문제 해결 교수 학습
도구 및 자료	컴퓨터, 스크래치
중점 CT요소	■분해 ■패턴인식 ■추상화 ■알고리즘 ■프로그래밍

가. 수업의 개요

본 수업에서는 학생들이 농업에서 자동화 시스템을 갖추어야 할 필요성을 알고, 센서보드를 이용하여 나만의 스마트 스프링클러를 만드는 시간을 갖는다. 학생들은 나만의 스마트 스프링클러를 만들기 위하여 센서보드를 조작하고 스크래치로 스프링클러 프로그램을 제작한다. 프로젝트를 계획하고 실행하는 과정 속에서 학생들은 분해→패턴인식→추상화→알고리즘→프로그래밍의 CT요소를 경험한다. 이러한 과정을 통해 학생들의 CT력을 신장시킬 수 있도록 수업을 설계하였다.

나. 학습목표

농업에서 자동화 시스템의 필요성을 알고, 나만의 스프링클러를 만들 수 있다.

다. 수업 단계 및 활동

문제 상황 제시

우리나라는 이제 미국, 네덜란드와 같은 선진농업 국가로부터 농업기술을 도입해 자급자족형 농업에서 수출형 농업을 하는 시대가 되었습니다. 2000년대 들어서는 농업수출액이 점차 증가했고 이제는 아프리카와 동남아와 같은 개발도상국에 농업기술을 전수해주기에 이르렀습니다.

필리핀의 경우 과거에는 채소종자를 직파해서 심거나 모종을 TRAY에 키워 육묘하는 것이 아니고 직접 밭에 심어 모종을 만들었다고 합니다. 이제는 우리나라의 자동화 육묘시설을 이용하여 상토준비, 파종, 재배 관리 작업 등이 자동으로 이루어진다고 합니다.

자동화 시스템을 갖추면 대량생산이 가능해지고 충실한 모종을 만들 수 있어 품질이 높은 농산물을 수확할 수 있습니다. 이러한 자동화 시스템 덕분에 식량난 해결의 방법을 찾을 수 있게 된 것입니다.

-농림수산식품부 대학생 블로거 기자, 고려대학교 생명공학과 안정모

- 제시 글을 통해 학생들이 농업에서도 자동화 시스템을 갖추어야 할 필요성을 인식할 수 있게 한다.
- 농업에서의 자동화 시스템의 예로 스마트 스프링클러가 있다는 것을 소개하고 다음 단계에서 스마트 스프링클러 예제를 학생들이 자유롭게 탐색할 수 있는 시간을 갖게 한다.

스마트 스프링클러 예제 살펴보기

- 학생들이 스마트 스프링클러 예제를 자유롭게 탐색할 수 있도록 충분한 시간을 부여한다.
- 교사가 직접 설명해주기보다는 학생들 스스로가 프로그램의 작동 원리를 충분히 생각해 볼 수 있도록 지도한다.

배경지식 살펴보기

전기가 흐를 때 방해를 받게 되는데 전류의 흐름을 방해하는 것이 바로 '저항'입니다. 저항이 생기는 이유는 전자가 물질을 이동할 때 물질을 구성하고 있는 원자와 부딪히게 되는데 이러한 충돌은 전자의 이동을 방해하기 때문입니다. 저항의 크기는 물질의 종류에 따라 달라지며, 단면적과 길이에도 영향을 받습니다. 물질의 종류에 따라 구성하는 성분이 다르므로 전기적인 특성이 다릅니다.

소금은 고체 상태에서는 전류가 흐르지 않습니다. 고체 상태에서도 전기를 띤 입자는 존재하지만 고체 상태에서는 이들 입자들이 강하게 결합되어 있기 때문에 도선을 연결해 주더라도 전류가 흐르지 않습니다. 그러나 소금을 액체 상태로 만들면 상황은 달라집니다. 액체 상태에서는 구성 입자들이 움직일 수 있기 때문에 전류가 흐릅니다. 소금이 물에 녹았을 때 전기를 띤 입자인 양이온과 음이온으로 나뉘기 때문에 이 입자들이 수용액 속에서 전하를 운반할 수 있기 때문입니다. 순수한 물의 경우는 어떨까요? 순수한 물에는 전자나 이온 등 전하를 운반할 수 있는 입자가 없기 때문에 전류가 흐르지 않습니다.

-에듀넷, 네이버 지식백과

- 빛과 소리는 학생들에게 익숙하나 저항은 다소 생소한 개념일 것이다. 배경지식 살펴보기를 통해 학생들에게 수분이 적을수록 저항 값이 올라간다는 것을 인식시켜주어야 할 필요가 있다.

Decomposition(분해)

스마트 스프링클러를 만들기 위해 필요한 데이터에는 무엇이 있을지 생각해봅시다.

1. 땅의 수분 정도
2. 밝기
3. 소리의 세기

- 스마트 스프링클러를 만들기 위하여 어떠한 입력 값들이 필요한지 생각해보는 시간을 갖는다.

Pattern Recognition(패턴인식)

실험을 통해 스마트 스프링클러를 만들기 위해 필요한 데이터를 모아봅시다.

땅의 수분정도	저항 값
수분부족	80

- 센서보드를 조작하면서 입력 값과 출력 값에 대한 데이터를 수집하는 시간을 갖는다.

Abstraction(추상화)

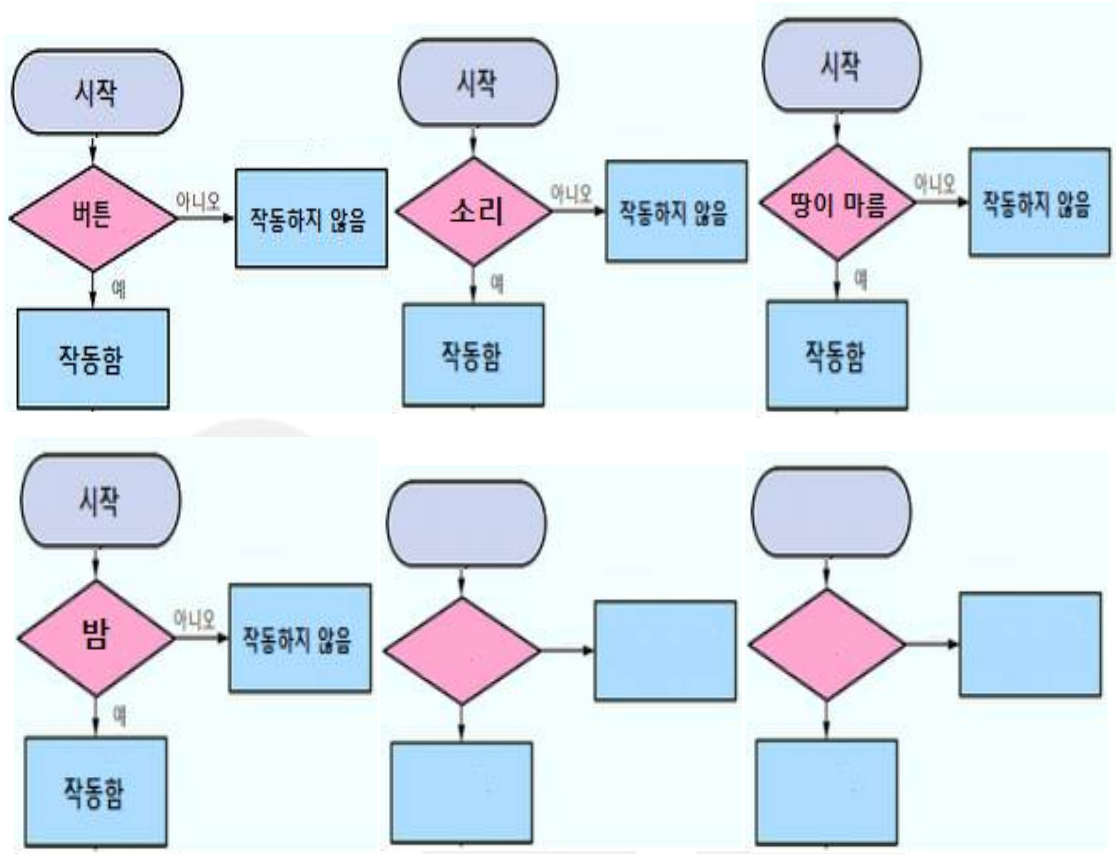
스마트 스프링클러가 작동하기 위해 필요한 조건이나 식을 정리하여 봅시다.

1. 수분 저항 값 > 80 이면 스프링클러 돌기
2.
3.

- 패턴인식 단계에서 수집한 데이터를 바탕으로 스마트 스프링클러를 만들기 위해 필요한 조건이나 식을 간단하게 작성해보는 시간을 갖는다.

Algorithm(알고리즘)

앞에서 정리한 것들을 바탕으로 알고리즘을 구상해봅시다.



- 추상화 단계에서 작성한 조건이나 식을 명확하게 알 수 있도록 정리한 내용을 순서도로 표현한다.

Programming(프로그래밍)

앞에서 작성한 순서도를 바탕으로 센서보드를 이용하여 나만의 스마트 스프링클러를 만들어봅시다.

- 알고리즘 단계에서 정리한 순서도를 바탕으로 스마트 스프링클러 프로그램을 제작한다.

발표 및 공유하기 (평가하기)	- 자신(팀)이 만든 프로젝트를 발표하며 친구들과 함께 공유한다. 그 과정에서 친구들과 주고받는 질문과 답을 통해 동료평가도 동시에 실시할 수 있다.
피드백	- 앞서의 과정을 통하여 피드백을 주고받는다.
재도전하기	- 피드백을 통하여 자신의 프로젝트를 개선하기 위한 재도전이 이루어진다.

라. 평가계획

연번	평가 기준	방법
1	주어진 문제를 다양한 요소로 분해할 수 있는가?	관찰
2	주어진 문제에서 패턴을 찾을 수 있는가?	관찰
3	문제 해결 과정을 추상화 하여 순서도로 나타낼 수 있는가?	관찰
4	나만의 스프링클러 만들기 활동에 적극적으로 참여하였는가?	관찰
5	센서보드를 이용하여 나만의 스프링클러를 제작할 수 있는가?	수행평가

5. 마치며

본 수업에서는 분해-패턴인식-추상화-알고리즘-프로그래밍의 단계를 순차적으로 적용하였다. 그러나 CT신장을 위한 모든 수업들에 위의 단계들을 순차적으로 적용하기에는 다소 어려움이 있다. 교사가 의도적으로 수업에 각 단계들을 순차적으로 적용하여도 학습자들이 학습 내용과 CT요소를 이해하는데 어려움을 겪을 것이다. 수업을 준비함에 있어서 단계의 순서 바꾸기, 자료수집과 자료 분석 등의 단계 추가하기 등 다양한 전략이 필요하다.

온라인 교육 사이트를 이용한 SW교육

온라인 SW교육 사이트(Koreasw.org)는 초중등 교사들을 위하여 현장 교사들이 직접 참여하여 강좌를 만들고 운영하고 있다. 연구 기관인 만큼 SW교육에 대한 깊은 고민과 함께 학생들을 직접 가르쳐 본 경험을 바탕으로 강좌를 제작하였다.

온라인 SW교육 사이트는 창의컴퓨팅을 통해 학생들의 사고력, 즉 CT를 길러 미래를 살아갈 수 있는 역량을 길러주고자 하는 목표를 추구하고 있다.

강좌는 '컴퓨터과학, 프로그래밍, 피지컬 컴퓨팅' 세 개 영역으로 나누어져 있다. 이는 소프트웨어교육 운영지침 및 2015 개정교육과정에 맞춰 교육과정과 연계한 것으로 학교에서 교사들이 학생들과 함께 쉽게 이해하며 SW교육을 할 수 있도록 27개의 강좌를 구성하였다.

언플러그드 강좌를 통해 컴퓨터과학에 대한 이해를 하고 이를 바탕으로 스크래치와 같은 프로그래밍을 통해 추상화, 자동화 과정을 거쳐 문제를 해결해보며, 피지컬 컴퓨팅을 통해 실세계와 연결해 보는 경험을 가질 수 있다.

강좌는 교사뿐만이 아니라 교사의 도움으로 학생들이 가정에서 스스로 학습할 수 있으며, 학부모들의 SW교육에 대한 이해를 돕기 위해 학부모들도 강좌를 들을 수 있다.

강좌는 매 강좌마다 유닛별로 쪼개져 있어서 자신에게 필요한 부분을 집중해서 들을 수 있으며, 교재와 소스코드가 함께 제공되고 있다.

미래인재양성을 위한

초중등교사들을 위한

교사들이 만든 온라인 SW교육 사이트

온라인 SW교육 강좌 로드맵

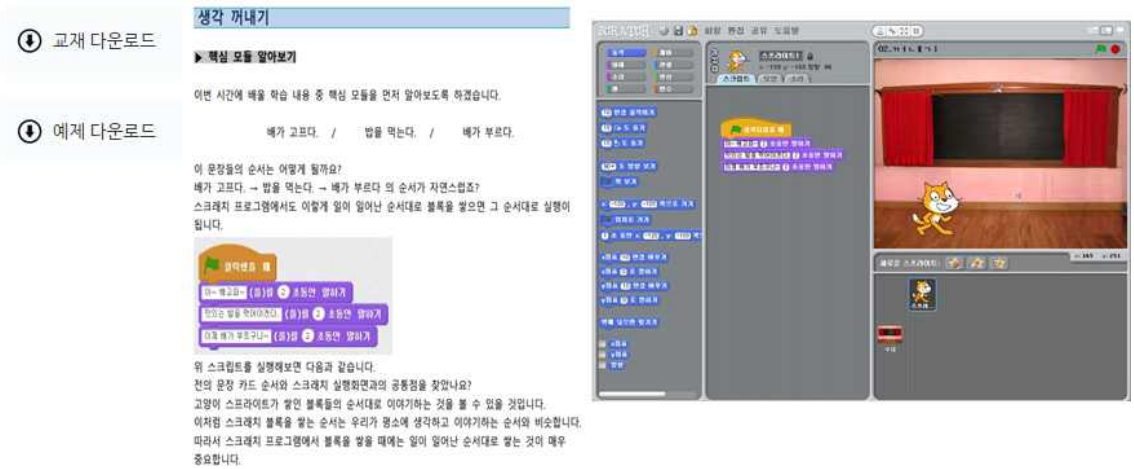
1. Koreasw.org - SW교육과 관련한 동영상 및 학습자료 제공

2. 온라인 교육을 통한 학습

- 교사, 학생, 일반일을 대상으로 자신의 관심과 수준에 맞추어 학습 가능
- 자발적인 학습이 가능하며, 학교에서의 SW교육 시수가 부족한 현실에서 플립러닝 학습 가능
- 학습 → 놀이 → 평가 → 수료증 형태로 전개되어 학습에 대한 동기 부여

3. 교재 및 예제 자료 제공

- 각 강좌 차시별로 좌측 하단에서 다운 받을 수 있음



4. 창작 활동과 평가 체제

5. 뱃지 및 수료증 제공

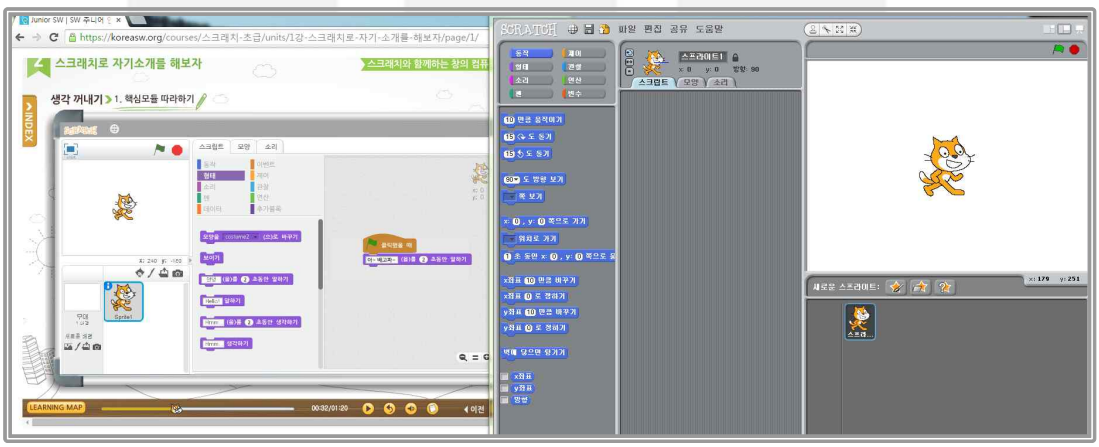
6. 온라인 SW교육 사이트 현장 활용팁!

교실에서는 이렇게 해보세요.

교실에서의 SW교육 학습은 학교마다, 교사마다, 교실마다 그 상황이 다 다를 수 있다. 아직 2015개정 교육과정이 적용되기 전이므로 SW교육 학습은 연구학교, 선도학교에서 주로 시행이 되고 있다. 하지만 시범학교가 아닌 경우에도 학교나 교사의 재량에 따라 SW교육은 학급 교육과정 안에서 얼마든지 이루어질 수 있다. 창의적 체험활동, 방과후수업, 동아리 활동 등을 통해 SW학습 시간을 확보해서 운영할 수 있다. 여기에서는 무료 온라인 SW교육 사이트를 활용해서 학습할 수 있는 경우를 가정학습과의 병행 여부로 나누어 안내해 보고자 한다.

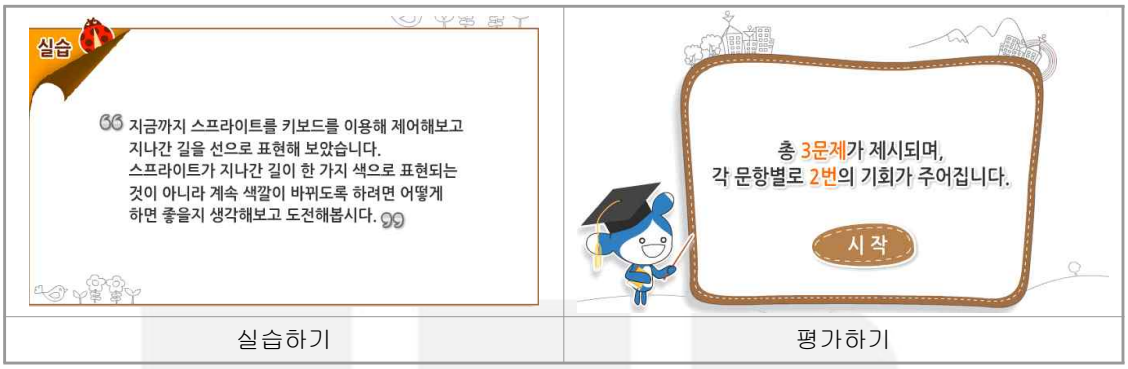
가. 가정학습과 병행하지 않는 경우

① 창의적 체험활동, 방과후 수업, 동아리 등의 시간을 활용하여 여러 강좌 중 하나를 선택하여 운영한다.
② 예를 들어, 학생들에게 스크래치를 알려주고자 할 때 스크래치 초급을 선택하여 14강에 대한 학습계획을 학급 교육과정에 맞게 구성한다.(차시별 제공되는 교재 참고)
③ 컴퓨터실에 가서 https://scratch.mit.edu/ 에서 스크래치 1.4 또는 2.0을 다운받는다.
④ koreasw.org 사이트에서 해당강좌를 열고, 스크래치 창을 동시에 실행시킨다.
⑤ 강좌를 보며, 직접 스크래치 프로그램을 열어 따라해 볼 수 있다.



※ 이 활동은 교사가 칠판 스크린에 강좌를 열어주고 따라해 볼 수 있도록 수업을 구성할 수 있다. 하지만 이 경우에는 학생의 수준에 따라 학습 속도에 있어 차이가 많이 나므로 개별 학습으로 진행할 수 있도록 구성하는 것을 권장한다. 이해가 어려운 학생은 교사 또는 그 활동을 완료한 친구의 도움을 받을 수 있도록 팀티칭으로 구성할 수 있으며, 학습을 빨리 끝낸 학생의 경우에는 그 차시에서 배운 것을 토대로 자신만의 프로젝트를 만들어 볼 수 있는 시간을 주도록 한다.

⑥ 실습 과제 해결하기(실습 해법도 강좌에서 안내가 되므로 교사의 부담이 없음)
⑦ 평가하기를 통해 자신의 성취수준 확인하기



실습하기 | 평가하기

⑧ 학습정리하기
⑨ 자신이 만든 프로젝트는 학급 홈페이지(학급 상황에 맞추어)에 저장하기
※ 저장한 프로젝트를 다른 친구들과 함께 공유하며 칭찬 또는 새로운 아이디어를 추가해줄 수 있는 답글을 달아줄 수 있는 분위기를 형성하여 격려해주면 좋다.

⑩ 강좌를 다 들으면 '완료'버튼 누르고 나오기

⑪ 3~4개의 강의를 듣고서, 학생들이 만든 프로젝트 발표 및 공유 시간 갖기

⑫ '스크래치 초급' 강좌를 다 들으면, 온라인상에서 뱃지를 받고, 수료증을 출력하여 수여하기(마이 페이지 - 나의 강의실 - 수강 완료한 강좌 - 수료증)

나. 가정학습과 병행하는 경우

SW교육을 위한 학습 시간을 학급교육과정에서 확보하기 어려운 경우 가정학습을 통해 학습을 권장할 수 있다.

하지만 앞서 <가정학습과 병행하지 않는 경우>에서 설명한 과정은 학습 시작 전에 미리 안내가 되어야 한다. 사이트 주소를 알고, 회원 가입을 하고, 스크래치 프로그램을 다운 받는 방법을 익히게 되면 학생은 과제 형태로 학습을 하게 된다.

간단한 학습 진도표를 작성하여 학생 스스로 이수한 강의에 대해서는 체크를 하고, 교사는 이에 대한 적절한 보상제도를 마련하여 학습활동이 지속적으로 이루어질 수 있도록 강화시켜 주면 좋다.

한 달에 한 번 정도는 학생들이 만든 프로젝트를 발표하는 시간을 가져보며 자신의 작품에 대해 공유할 수 있는 시간도 만들어 준다.

한 강좌에 대한 모든 수강이 끝나면 수료증을 출력하여 교사가 학생에게 수여해 준다.

TIP!

가정학습의 경우, 교사나 친구와의 상호작용 없이는 혼자서 지속적으로 학습하기는 어렵다. 학습과정에서 궁금한 사항이 생길 수도 있고, 자신이 만든 프로젝트를 친구들과 공유하는 것은 참으로 중요하다.

이 때, 교사는 학급홈페이지를 통해 SW교육과 관련된 학습방을 하나 만들어서 학생들이 프로젝트 공유 및 궁금하거나 도움을 필요로 하는 글을 서로 올리며 해결할 수 있는 장을 마련해주어야 한다.

학생들이 서로 소통하고 공유하는 장이 SW교육에서 꼭 필요한 이유는 자신이 만든 것을 뽐내며 친구들에게 칭찬받아 서로 동기부여가 되는 측면도 있지만, 친구가 만든 프로젝트를 보며 자신 또한 아이디어를 얻거나 새로운 것을 배울 수 있는 기회를 가질 수 있기 때문이다. 새로운 것을 처음부터 만들기 힘든 학생들은 친구가 만든 프로젝트를 그대로 다운받아 요리조리 만져보는 과정을 통해 배우기도 하며, 그 안에서 많은 생각을 하게 된다. 이 때, 학생들은 자신의 창의성을 풍부하게 쏟아낸다.

학교에서는 이렇게 해보세요.

학교에서는 'SW페스티발'이라는 시간을 통해 SW에 대한 관심과 교육의 효과를 높일 수 있다. 해마다 가을이면 학급 또는 학년 단위로 발표회를 갖는다. 그간 학급에서 학생들이 프로젝트를 개인 또는 팀을 이루어 만들어낸 최종 산출물을 친구와 부모님 앞에서 발표하는 시간을 가져보는 것이다. 일명 '우리반 SW 축제', 'SW 페스티발' 이라는 다양한 이름으로 학생들을 칭찬해주고 학생들이 미래를 꿈꿔볼 수 있는 장을 마련해 주는 것이다.

> 강좌 선택 → 강좌 수강 → 한 달에 한 번 자신의 프로젝트 발표(강좌에 대한 지속적인 피드백을 위해 필요한 과정, 횟수는 재량) → 강좌 수강이 끝나면 해당 강좌에 대한 뱃지 받고, 수료증 수여 → 반별(학년별)로 프로젝트 주제 설정 → 개인 또는 팀을 이루어 프로젝트를 만들고 친구들과 함께 공유하며 지속적으로 프로젝트 수정하기 → 페스티발에서 프로젝트 발표 및 공유하기

[예시 활동 안내]

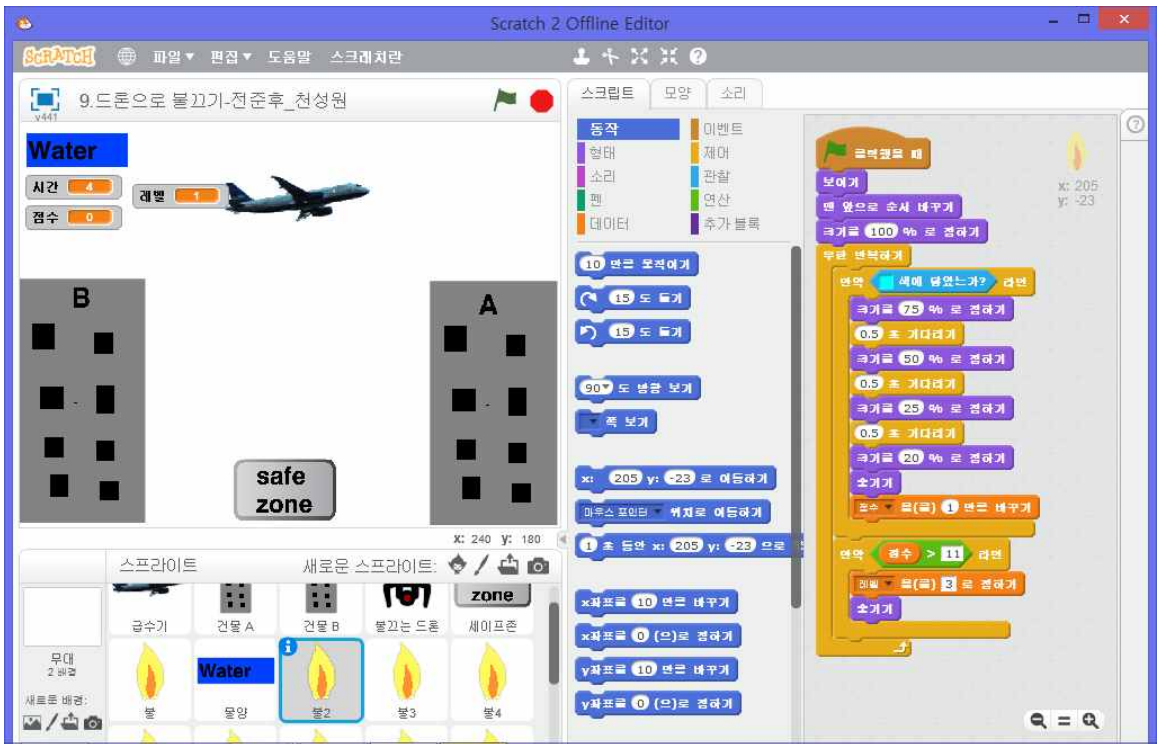
2015년 SW심화 캠프에서 실시하였던 활동을 안내하고자 한다.

심화캠프의 주제는 '위험으로부터 인간을 보호하라'였다. 위험으로부터 인간을 보호하기 위한 문제를 해결하는데 필요한 다양한 창의 컴퓨팅 프로그래밍과 센싱 도구(센서통합형보드, 센서분리형보드, 모션인식장치, 아두이노)를 활용하여, 학생 자신의 일상생활에서 느끼고 있는 문제를 위에서 제시한 주제에 맞추어 다양하게 프로젝트 산출물을 만들어내며 해결방안을 제시하였다.

CT신장을 위한 창의컴퓨팅

창의컴퓨팅 교육은 컴퓨터과학의 이론적 기초와 함께 코딩 능력, 프로그래밍 능력, 정보처리 능력을 이용하여 소프트웨어를 창의적으로 산출하는 교육활동을 주요 내용으로 다룬다. 초중등 교육에서는 이러한 컴퓨팅 과정을 통하여 소프트웨어개발 전문 지식과 기능을 목표로 하기 보다는 문제해결력, 창의적 사고, 팀 프로젝트를 통한 협업능력 등 초중등 교육에서 추구하는 교육의 목표를 추구한다. 즉, 컴퓨팅 과정을 통하여 자신의 소질과 적성을 발견하여 미래비전을 달성하는 전인적 인간을 양성하고자 한다. 실생활과 연계한 과정 속에서의 CT(컴퓨팅 사고), 프로그래밍 과정에서의 CT(컴퓨팅 사고)를 위한 창의컴퓨팅 활동이다.

[발표자료1 : 모션 인식 장치 인터페이스, 사람을 위한 창의적인 융합]



드론을 이용하여 화재를 진압하자

발표한 학생들은 위험으로부터 인간을 구하기 위한 상황을 화재를 진압하는 것으로 설정하였고, 모션인식장치를 활용하여 드론을 조정하였으며 화재가 난 곳에 드론이 다가가서 물을 뿌리며 화재를 진압하며 위기에 빠진 사람을 구하는 문제를 해결하였다.

[발표자료2 : 센스에 센서를 더하다, 사람을 위한 창의적인 융합]

장애인들을 위한 특수장치들(학생들이 쓴 발표자료)

안녕하세요? 저희가 만든 프로그램은 장애인을 위한 프로그램입니다.

(동영상보고)첫 번째는 시각 장애인이 앞을 보지 못해 사고가 발생할 수 있는 것을 대비하는 방법입니다. 사고가 생길 것을 대비하여 빛 감지 센서를 이용하였습니다. 앞에 있는 물건이나 사람 등이 앞에 있으므로 어두워져 빛이 감지되어 팝 소리가 나오게 됩니다. 그래서 시각 장애인이 그 소리를 듣고 피해가서 사고가 발생할 경우가 줄어들 것입니다.

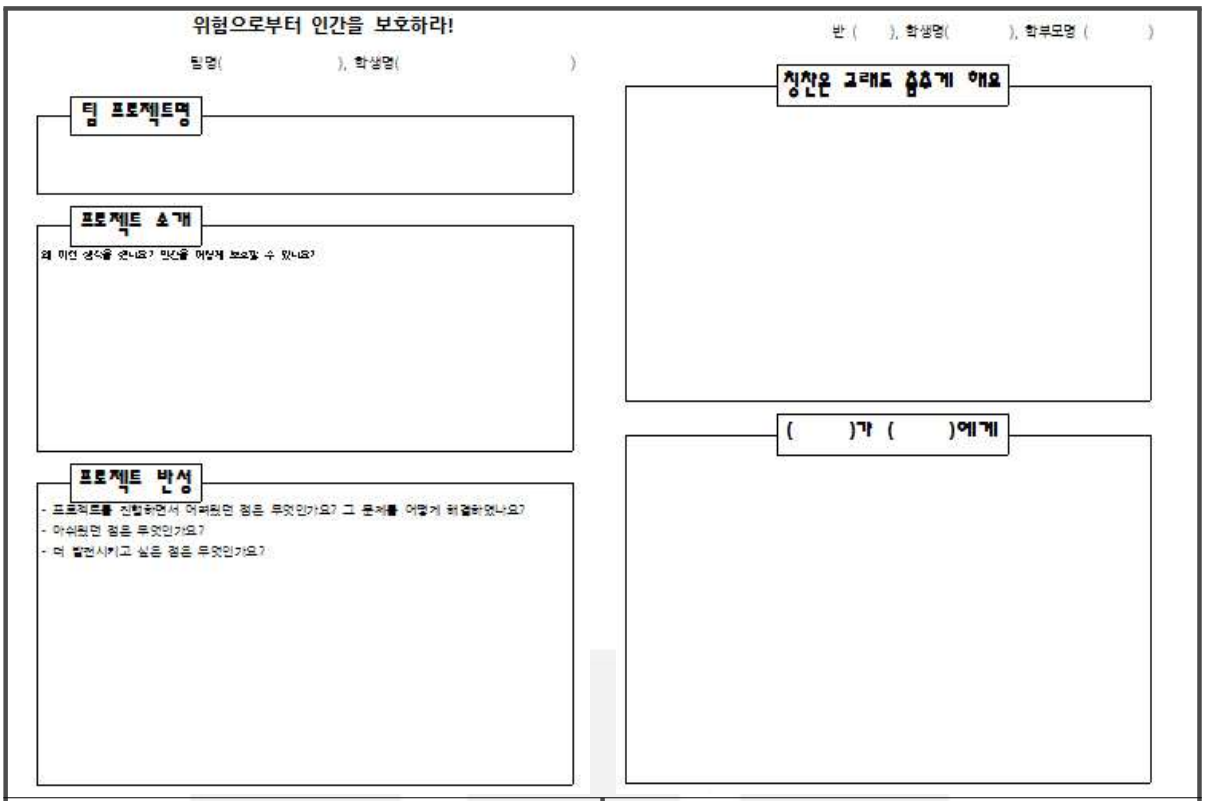
(동영상보고)두 번째는 청각장애인을 위한 장치입니다. 청각장애인은 다른 사람의 말을 듣지 못합니다. 그것을 방지하기 위해 소리 센서를 이용해 미리 정해둔 음량에 맞은 음량이 소리 센서에 감지된다면, 상대가 자신을 부르고 있다는 것을 컴퓨터가 알려줍니다.

(동영상보고)세 번째는 말을 못하는 장애인을 위한 것입니다. 말을 못하는 장애인도 생각이 있기 때문에 전할 말이 있습니다. 하지만 말을 못하는 장애를 가진 사람은 그것을 다른 사람에게 전하지 못합니다. 그런 것을 방지하고자 이런 장치를 만들었습니다. 전하고 싶은 전하고 싶을 때, 버튼을 누르면 '무엇을 말할 것인가요?'라는 말이 나옵니다. 장애인은 전하고 싶은 말을 키보드를 이용해 적으면 그 말의 높낮이를 슬라이더를 이용해서 조절 하고, 그 말을 컴퓨터가 말을 하지 못하는 사람을 대신하여 말해주는 것입니다.

네 번째는 호흡기 장애인이 되는 것을 방지하는 것입니다. 습도가 높으면 호흡기 장애인이 될 수도 있다고 합니다. 그래서 저항을 이용해 습도가 높은지 낮은지, 위험한지 위험하지 않은지를 알게 해줍니다. 그것을 보고 사람들은 호흡기장애를 예방할 수 있습니다.

다섯 번째는 시각 장애인들이 앞이 보이지 않아 지팡이를 사용하는 경우를 흔하게 볼 수 있습니다. 지팡이를 사용한다고 하더라도 시각 장애인이 길을 잘 갈수 있는 것은 아닙니다. 그래서 그것을 조금 더 보안하고자하여 지팡이의 끝에 저항 장치를 달아 비가 온 뒤 축축해진 땅을 장애인이 알아 피해갈 수 있게 하였습니다.

※이와 관련한 자료(SW기초캠프교재, SW심화캠프 교재, 학생 발표 영상 등)는 computing.or.kr 사이트에서 다운받아 볼 수 있습니다.



학생용 학습지

(공유 및 발표 시간에 활용할 수 있는 학습지 예시자료임)

학부모 칭찬노트

(학생들 프로젝트 발표를 보고 난 뒤 칭찬의 글을 써서 게시판에 붙여 학생들로 하여금 자부심을 가질 수 있도록 할 수 있음)

소프트웨어 교육사이트 목록

2016 미래인재연구소 분석 자료(2016. 3. 31)

Code
주니어를 위한 코딩 교육 사이트
https://code.org/learn

Codecademy
다양한 프로그래밍 언어를 실습을 하며 배워가는 코딩교육 사이트
https://www.codecademy.com/learn

KhanAcademy
동영상 기반의 컴퓨터과학 및 프로그래밍을 위한 쉬운 강의 사이트
https://www.khanacademy.org/computing/computer-programming

CodeChef
자바를 비롯한 다양한 스크립트 언어 프로그래밍 교육
https://www.codechef.com/ide

CodingGround
리눅스 운영체제 내에서 쉘프로그래밍 기반으로 프로그래밍 교육
http://www.tutorialspoint.com/compile_c_online.php

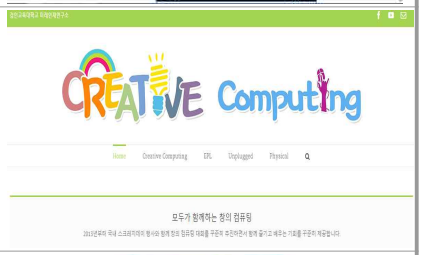
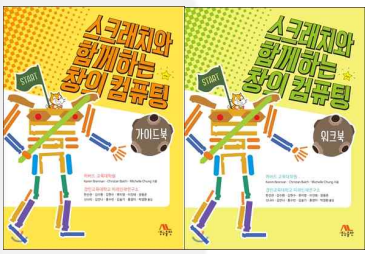
Visualize programs
Python, Java, JavaScript, TypeScript, Ruby, C, and C++ 프로그램의 시각적 실행 교육
http://pythontutor.com/

MIT open courseware
MIT대학에서 제공하는 무료오픈 프로그래밍 언어 교육 강좌
http://ocw.mit.edu/

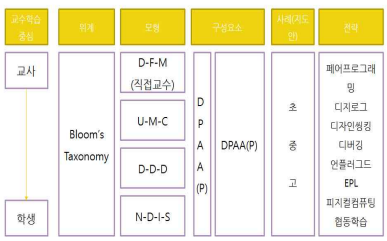
HTML5 Rocks
HTML5를 학습하기 위한 Google project 사이트
http://www.html5rocks.com/en/

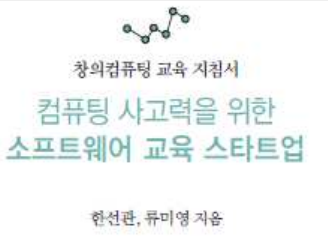
Lynda
프로그래밍과 컴퓨터과학을 배우기 위한 유료사이트
http://www.lynda.com/

미래인재연구소의 이모저모

시기	활동내용	
2015.1.1. ~현재	2년차 온라인 SW교육 사이트 운영 :미래창조과학부에서 주관하고 미래인재연구소에서 운영하는 무료 SW온라인 교육 사이트(KoreaSW.org)	
2014.1.1. ~현재	컴퓨팅 사이트 운영 : 미래인재연구소에서 창의컴퓨팅을 보급하기 위해 다양한 교육이론과 실천사례를 제공하는 커뮤니티(computing.or.kr)	
2015.1. ~현재	다양한 컴퓨팅 책 출판 -로렌과 함께 하는 유저랜드의 비밀, 사이언스주니어 -스크래치와 함께 하는 창의컴퓨팅 가이드북, 워크북, 생능출판사 -스크래치 창의컴퓨팅, 성안당 -스크래치 마법레시피 20, 성안당	  
2015.3. ~5.	언플러그드 번역작업 http://computing.or.kr/category/unplugged/ 자료탑재	

시기	활동내용	
2015.5 ~현재	스냅 매뉴얼 번역 작업	
2015.6.7 ~ 6.25	UC 버클리대 스냅팀 컴퓨팅 교육협력 및 실리콘 밸리 탐방	
2015.5. 9	Scratch Day in Korea(경인교육대학교 경기캠퍼스)	
2015.7~ 12	SW창의캠프 기초캠프(강원권 원주 상지대, 수도권 경인교대, 영남권 대구교대, 심화캠프 실시	
2015.7.27 ~ 7.28.	인천 초등학생 정보문화캠프	
2015.8.3 ~ 8. 4.	인천 SW교육 초등 교사연수	
2015.8.6.	한국정보교육학회 스크래치데이 행사 발표	

시기	활동내용	
2015.8.11 ~8.12.	한영 공동세미나 및 워크샵	
2015.4 ~12.	찾아가는 창의컴퓨팅 교육 민간지원활동	
2015.8 ~2016.2.	플루이드 인터페이스 기반 융합콘텐츠 개발 3~4학년 : 플루이드 인터페이스 히스토리, 마음으로 소통해요 5~6학년 : 디지털과 아날로그의 만남, 교실 안의 메이커 운동	
2015.5 ~2015.7	각종 매체 창의컴퓨팅 영상 소개(EBS, 다나와 등)	
2015.12.4 ~ 5	SW교육 주간행사 교사워크숍 실시 - 미국 버클리대 SNAP 개발교수 초빙	
2015.12 ~1	SW교수학습모형 개발 연구(Keris)	

2016.3	computing.or.kr 사이트에 무료 배포를 위한 SW교육자료 탑재 - 언플러그드, EPL, 피지컬 컴퓨팅 - 교사연수자료 - 학생캠프자료 - 온라인 SW교육사이트(KoreaSW.org) 사이트 활용 안내 - 구글CT자료(번역)	
2015년~ 현재	2015.1. 8. 이슈리포트 1호 발간 2015.8.17. 이슈리포트 2호 발간 2016.4. 1. 정기리포트로 발간	
2016.4	창의컴퓨팅 도서 출판	

창의컴퓨팅 정기리포트 3호 2016.4

배포일 2016년 4월 1일

배포인 한 선 관

편집인 류 미 영

배포처 경인교육대학교 미래인재연구소

http://in.re.kr http://computing.or.kr

주 소 인천광역시 계양구 계산동 경인교육대학교

전 화 032-540-1299

팩 스 032-548-0288

이메일 han@gin.ac.kr