

Computational Thinking 개념 평가를 위한 스크래치 코드 분석 시스템 개발

김수환[†]

요 약

본 연구의 목적은 Computational Thinking(CT) 교육에서의 학습자의 CT개념 평가를 지원하기 위한 스크래치 코드 분석 시스템을 개발하는 것이다. 최근 초중등교육에서 활발하게 활용되고 있는 교육용 프로그래밍 언어인 스크래치는 이벤트 기반언어로 전체적인 프로젝트의 구조를 파악하기가 어렵게 되어 있다. 따라서 본 연구에서는 학습자들이 자신이 만든 프로젝트를 디버깅 할 때나 교사가 학습자의 코드를 보고 평가 및 피드백을 할 때, 활용할 수 있는 코드 분석 시스템을 개발하고 실제 사용성을 검증하였다. 본 연구에서 개발한 스크래치 코드 분석 시스템은 SW 교육에서 디버깅과정을 촉진하거나 평가의 도구로 활용할 수 있다.

주제어 : 평가, 스크래치 코드 분석, SW 교육, 컴퓨팅 사고력

Development of Scratch Code Analysis System for Assessment about Concepts of Computational Thinking

Soohwan Kim[†]

ABSTRACT

The purpose of this research is to develop scratch code analysis system about scratch project for assessment in computational thinking(CT) education. Recently, scratch which is event-based language is used actively in K-12 SW education but it is difficult to understand the structure of scratch project. Therefore, in this study, we developed the system that students can use it to debug their scratch project and teachers can give feedback to students or evaluate students' projects through this system, and verified its usability. We are able to use this system to facilitate students' debugging process and to evaluate students' project in K-12 SW education.

Keywords : Assessment, Scratch Code Analysis, SW Education, Computational Thinking,

[†] 종신회원 : 종신대학교 조교수
논문접수: 2015년 8월 27일, 심사완료: 2015년 10월 15일, 게재확정: 2015년 10월 26일

1. 서론

문제해결력은 현대 교육에서 추구하는 교육목표이며, 미래 사회에 필요한 역량으로 최근 Computational Thinking(이하 CT)의 중요성이 부각되고 있다. 미국, 영국을 중심으로 한 선진국에서는 미래 인재 양성을 위해 CT를 목표로 한 컴퓨팅 관련 교과를 초중등교육에 도입하고 있다. 우리나라에서도 2018년 SW교과를 초등학교와 중학교에서 필수로 가르치도록 교육과정을 개정하고 있다[1][2][3][4].

CT를 증진하기 위한 컴퓨팅 교과의 내용에는 컴퓨터 없이 컴퓨터 과학의 원리를 가르치는 언플러그드, 교육용 프로그래밍 언어를 활용한 알고리즘 및 프로그래밍 교육, 센서들과 하드웨어를 연결하고 코딩을 통해 현실세계의 문제를 해결하는 내용으로 구성된 피지컬 컴퓨팅 등으로 구성된다[5]. 이 중에서 교육용프로그래밍 언어를 통한 알고리즘 및 프로그래밍 학습은 초중등 컴퓨팅 교육에서 필수적으로 다루어지며, CT 증진을 목표로 하고 있다[4][5].

전통적인 알고리즘과 프로그래밍 학습에서는 성인학습자들이 대상이므로 산업용 언어인 C, JAVA 같은 언어를 가르치나 보편적 교육으로서의 CT 교육에서는 학습자의 수준과 흥미, 재미를 고려한 교육용프로그래밍 언어가 활용되고 있다. 최근에는 텍스트기반의 어려운 언어보다는 스크래치, Starlogo TNG, 엔트리 같은 비주얼 블록형태의 언어들이 주로 활용되고 있다[6]. 교육용프로그래밍 언어를 활용한 학습에서는 학습자들이 일정기간 동안 학습하게 되면 자신만의 프로젝트를 구상하여 만들 수 있게 되는데, 이때 학습자들은 자신의 코드를 테스트하고 디버깅하는 과정을 필수적으로 경험하게 된다[7]. 테스트와 디버깅을 위해서는 학생들은 자신의 코드와 프로젝트의 구조를 파악하고 오류를 발견, 수정할 수 있어야 하며, 교사들은 학생들의 프로젝트를 보고 어떤 부분에서 막혀있는지에 대한 파악과 작품의 향상을 위한 피드백을 해주어야 한다[8]. 이와 같은 목적을 달성하려면 교사나 학생들이 자신의 스크래치 프로젝트를 파악하고 오류를 발견할 수 있는 도구가 필요하다.

스크래치는 언어 특성상 이벤트 기반의 언어의 특성과 스프라이트 중심의 객체지향 특성을 가지고 있어서 코드가 복잡해지면 전체적인 구조를 파악하기가 어렵다. 또한 초보학습자들의 경우 대부분 점진적인 시도와 개발을 통해 코드를 확장하는 방법으로 작품을 만드는 경우가 많으므로[9], 자신이 만든 작품임에도 전체적인 구조와 방송받기 등의 관계를 파악하기가 어렵다[10]. 또한 수업 중 학생들의 코드를 보고 교사가 피드백을 주려고 할 때에도 스프라이트의 수가 많거나 코드가 많아질 경우 어느 스프라이트에서 어떤 코드가 오류가 생기는지 파악하기가 힘들다.

초보자들의 프로그래밍 오류는 연구자마다 다르게 분류하고 있는데, 일반적으로 어휘(Lexical), 구문(Syntax), 의미(Semantic), 논리(Logic) 오류로 분류한다[8]. 일반적으로 어휘오류는 인식할 수 없는 토큰을 입력한 경우 발생하는 오류이며 프로그래밍 과정에서 거의 발생하지 않는다. 스크래치의 경우 구문오류나 의미오류가 발생하지 않도록 설계되어 있다. 즉, 스크래치로 작품을 만들었을 때 발생하게 되는 오류는 대부분이 논리오류로 구분할 수 있다. 스크래치에서 논리오류의 경우 방송하기나 받기가 생각한대로 작동하지 않거나 스택을 제대로 만들지 못해서 원하는 결과를 얻지 못하는 경우가 많다.

또한, CT의 경우 사고력에 대한 평가이고 여러 요소를 포함하고 있어서 단순히 지필평가와 같은 방법으로 평가하기가 어렵다. 스크래치를 개발하고 다년간 교육팀을 운영해온 MIT 미디어랩에서도 CT 평가 방법으로 세 가지를 제안하고 있는데, 그중에서 학습자의 개념을 평가하기 위한 방법으로 프로젝트 분석법을 제안하고 있다[9].

따라서 본 연구에서는 학생들은 자신의 스크래치 프로젝트의 구조를 파악할 수 있고, 교사는 학생들의 작품을 살펴보고 피드백 및 평가를 할 수 있게 지원하는 스크래치 코드 분석 시스템을 개발하고 사용성을 검증하였다.

2. 이론적 배경

2.1 스크래치 코드 구조

스크래치는 MIT 미디어랩에서 만든 교육용 프로그래밍 언어로 현재 150 여 개국에서 활용하고 있다. 스크래치는 오프라인 기반인 1.4 버전과 온라인 기반인 2.0 버전이 있다. 1.4 버전은 스펙트럼 기반으로 작성되어 있어서 오프라인으로 설치가 가능하고 프로젝트의 메타데이터를 텍스트 파일로 추출할 수 있다.

메타데이터 파일의 구성은 <표 1>과 같다.

<표 1> 스크래치 1.4 메타데이터의 예

메타데이터	내용
History: 2014-5-27 23:14:03 save1 2014-5-27 23:54:29 save2 1	파일저장 히스토리
Totals: Sprites: 56 Stacks: 1184 Unique costumes: 42 Unique sounds: 33	전체 요약 스프라이트 수 스택수 모양수 사운드 수
Sprite: 무대 Costumes (6): 배경4 (480x360) 배경5 (480x360) 배경1 (480x360) 배경2 (480x360) 배경3 (480x360) 배경6 (480x360) Sounds (1): 꺼내기 (0:00:00) Stacks (8): when I receive "next" 다음 배경 end when I receive "end" 만약 ((점수) < "200") 라면 배경 "배경3" 로 바꾸기 만약 ("199" < (점수)) 라면 배경 "배경2" 로 바꾸기 end	무대 요약 모양수 사운드 수 스택의 세부 명령어 들
Sprite: easy Costumes (1): 모양1 (88x37) Sounds (0): Stacks (5): when green flag clicked 숨기기 end when easy clicked "easy" 방송하기 숨기기 end	스프라이트 요약 각 스프라이트 별로 모두 나타남. 모양수 사운드 수 스택의 세부 명령어 들

따라서 1.4 버전에서는 스크래치에서 어떤 명령어가 어떻게 사용되었는지 메타데이터로 확인이 가능하다.

2.0 버전은 플래시 기반으로 제작되어 있어서 기본적으로 웹 클라우드 방식으로 작동하나 오프라인 버전도 가능하다. <표 2>와 같이 스크래치 2.0의 경우 .sb2 파일로 저장되는데, 압축을 풀어 보면 project.json 파일에서 1.4 버전과 같이 메타데이터를 포함하고 있는 것을 알 수 있다[11].

<표 2> 스크래치 2.0 메타데이터의 예

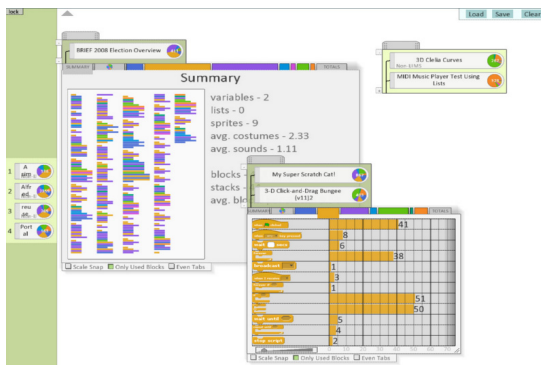
메타데이터	내용
"objName": "Stage", "variables": { "name": "P1", "value": 338, "isPersistent": false },	무대 코드 (변수, 스크립트, 사운드, 모양으로 구분됨)
"objName": "menu", "scripts": [[396, 136, ["whenGreenFlag"], ["xpos:", 0], ["ypos:", 0]], ["whenIReceive", "menu"], ["setSizeTo:", 100], ["ypos:", -180], ["show"],	스프라이트 코드 (변수, 스크립트, 사운드, 모양으로 구분됨)
"info": { "spriteCount": 9, "hasCloudData": true, "flashVersion": "WIN 18,0,0, 209", "videoOn": false, "userAgent": "Mozilla\5.0 (W indows NT 6.1; WOW64) Apple WebKit\537.36 (KHTML, like Gec ko) Chrome\44.0.2403.125 Safari \537.36", "projectId": "70359626", "scriptCount": 39, "swfVersion": "v438.1" }	파일에 대한 기 본 정보 스프라이트 개수 플래시 버전 비디오 활성화 프로젝트 아이디 스택 수 등

스크래치에서는 디버깅을 위한 도구로 ‘단계별 실행’이라는 메뉴를 제공하여 어떤 블록에서 막혀

있는지를 확인하는 것이 가능하나, 스프라이트 간의 방송 구조를 알 수 있는 기능이 없어서 전체적인 구조를 파악하기가 어렵다.

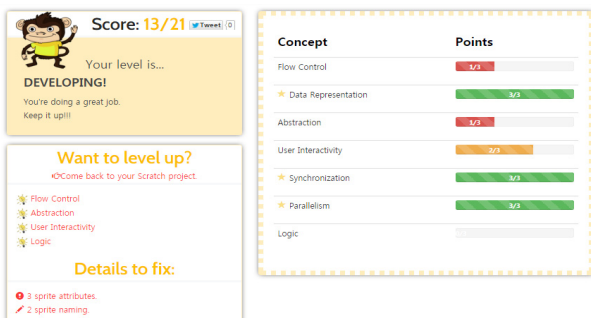
2.2 프로젝트 코드 분석툴

스크래치는 8개의 카테고리로 명령어 블록이 분류되어 있는데, 각 블록들의 사용횟수 파악을 통해 코드 분석이 가능하다. Brennan & Resnick [9]의 연구에 의하면 프로젝트 분석법은 CT의 개념을 평가하는데 활용할 수 있으며, 프로그램의 소스 코드를 분석하여 평가하는 방법은 SW 공학에서도 사용하고 있는 방법이다[12]. 스크래치의 코드를 분석할 수 있는 도구에는 Scrape, Drscratch 등이 있다[13][14]. Scrape는 <그림 1>과 같이 각 블록의 횟수와 저장횟수, 스프라이트 수, 블록수, 스택수 등의 분석이 가능하다[13].



<그림 1> Scrape 분석 화면

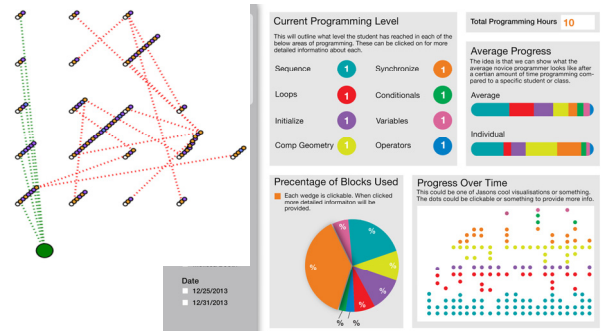
<그림 2>와 같이 Drscratch는 CT의 속성을 데이터 표현, 논리적 사고, 사용자 상호작용, 플로우 제어, 추상화, 병렬화, 동기화로 분류하고 각 점수를 판별하여 등급으로 보여주고 코드 복사, 스프라이트 이름변경, 죽은 코드, 스프라이트 속성을 보여준다[14].



<그림 2> Drscratch 분석 화면

등급에는 베이직, 개발자, 숙련자의 3등급이 있고 함께 점수로 판별된다.

<그림 3>은 Martin & Fields[15]가 개발한 도구로 스크래치 활용 학습에서 방송의 중요성에 초점을 맞추어 각 스프라이트 간의 방송 주고 받기를 한눈에 알아 볼 수 있는 부분을 제공한다. 기본적으로 다른 프로그램과 같이 블록빈도나 퍼센트 같은 정보도 제공해 주며 전체 학생들의 블록 빈도수와 개인 프로젝트의 빈도수를 비교해서 보여준다. 또한 시간에 따른 블록의 사용상황도 보여주도록 구성되어 있다. <그림 3>은 프로토타입으로 개발된 것으로 현재 서비스가 되지 않고 있다.



<그림 3> Fun 툴 분석 화면

스크래치는 이벤트 기반의 언어로 일정 수준을 지나면 스프라이트의 수가 증가하게 되고, 스프라이트들의 방송 주고 받는 구조가 복잡하게 된다. 이상에서 개발된 도구들은 스크래치의 구조를 한눈에 파악하기 어려우며, 2.0 버전의 소스를 분석하는데 어려움이 있다. 또한 평가를 위한 CT 개념 점수는 Drscratch만 제공되어 있다. 또한 실제 교육환경에서 사용가능한 사용성 테스트 결과도 없어서 현장적용에 대한 연구가 필요한 실정이다. 따라서 본 연구에서는 CT 개념 평가에 활용할 수 있는 Drscratch와 같은 규칙과 점수가 포함되고, 디버깅에 활용할 수 있는 스프라이트 구조도가 포함된 웹기반의 코드 분석 시스템을 개발하였다.

2.3 휴리스틱 평가

닐슨(Nielsen)이 제안한 휴리스틱 평가 방법 [16][17]은 유저 인터페이스 디자인에서 사용성에

대한 문제를 찾아내기 위한 방법이다. 휴리스틱 평가는 해당분야의 전문가들이 개발된 시스템을 사용해보고 어떤 문제점이 있는지, 치명적인 수정 사항이 있는지 점수를 매겨서 시스템의 사용성을 테스트 하는 방법이다.

닐슨의 연구에 의하면 해당 분야의 전문가 5-6 명의 피실험자를 통해 시스템의 85% 정도의 발견 점을 얻을 수 있다고 하였다[17]. 닐슨이 제시한 휴리스틱 평가 요소는 현재 상태 시각화, 현실세계 부합, 적절한 통제권, 일관성과 표준성, 실수방지, 적은 인지노력, 유연한 사용, 심미적이고 간결, 오류과약 수정, 충분한 도움말이다.

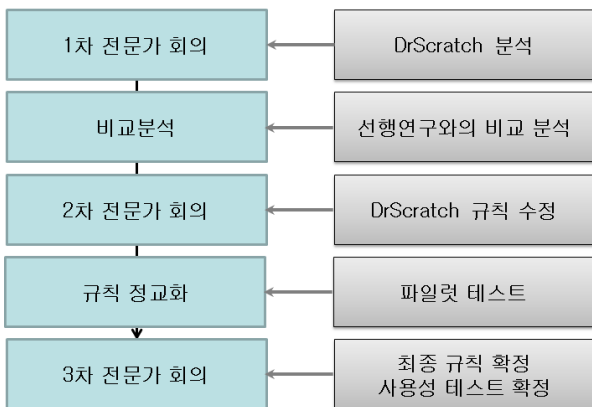
본 연구에서는 휴리스틱 평가법을 적용하여 개발된 시스템의 사용성을 평가하였다.

3. 코드 분석 시스템 설계 및 구현

3.1 코드 분석 알고리즘 개발

본 연구에서의 코드 분석 알고리즘은 Drscratch에서 사용한 코드분석 알고리즘을 수정하여 <표 3>과 같이 설정하였다.

코드분석 알고리즘은 컴퓨터교육 교수 1명, 컴퓨터교육 박사 1명, 석사 1명, 교사 1명, 교육용프로그래밍 언어 개발 전문가 2명이 참여하여 전문가 회의를 진행하였으며, 최종 규칙 개발절차는 <그림 4>와 같다.



<그림 4> 코드분석 알고리즘 개발 절차

알고리즘의 정확도를 높이기 위해 <그림 4>와 같이 전문가 회의 3회를 거쳐 규칙을 테스트하고 최종규칙을 <표 3>과 같이 설정하였다.

<표 3> 코드 분석 알고리즘

CT 개념	1점	2점	3점
절차	스택 수 4개 죽은코드 4개	스택 수 8개 죽은코드 2개	스택 수 10개 죽은코드 1개
반복	AC ^ 반복 (3회)	AC ^ 중첩 반복 (2회)	AC ^ 조건반 복 (2회)
분기	AC ^ if (3회)	AC ^ if else (2회)	AC ^ if ^ 논리 연산 (2회)
상호 작용	스프라이트 클릭	키보드, 마우 스 상호작용	물고 답하기 사용
데이터 표현	변수 사용	변수 ^ 연산	변수 ^ 연산 ^ 리스트
추상화	스프라이트 3 개 이상	스프라이트 5 개 이상 ^ 방송 사용	스프라이트 7 개 이상 ^ 방 송 ^ 사용자 블록
병렬 및 동기화	시작하기 블 록 2회(한 프라이트 안 에)	시작하기 블 록 2회 ^ 방 송 V 기다리 기 V 중복 이 벤트(키입력 동시, 클릭 동 시)	시작하기 블 록 2회 ^ 방 송 2회 V 무 대변환 V 스프 라이트 사라지 기(스테이지 개념)

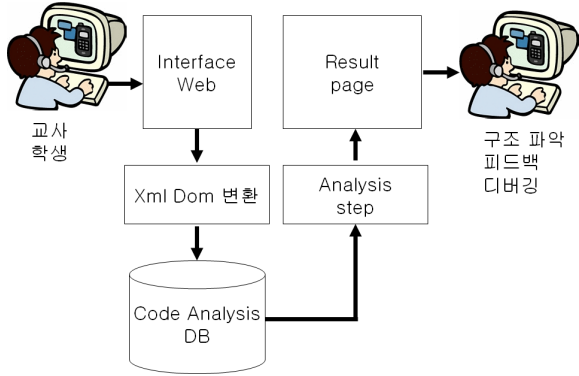
개발된 코드분석 알고리즘은 Drscratch의 규칙을 수정, 보완한 것으로 Brennan & Resnick[9]의 평가요소와 CT의 개념요소를 반영한 것이다. 기존의 Drscratch[14]에서는 데이터 표현, 논리적 사고, 사용자 상호작용, 플로우 제어, 추상화, 병렬, 동기화로 구분되어 있었으나 본 연구에서는 절차, 반복, 분기, 상호작용, 데이터 표현, 추상화, 병렬 및 동기화로 수정, 보완하였다. 점수부여 방식은 기존의 Drscratch의 방법과 동일하게 사용하였다.

3.2 코드 분석 시스템 구조

본 연구의 분석 시스템의 구조는 <그림 4>와 같이 설계하고 개발하였다.

분석과정을 살펴보면 먼저 교사나 학생들은 자신들이 만든 프로젝트 파일을 웹 인터페이스에 업로드 한다. 둘째, 파일을 업로드 하면 웹서버에서 파일의 구조를 파악하고 xml Dom 구조로 변환하여 DB에 저장한다. 셋째, DB에 저장된 요소들을 Drscratch 알고리즘을 수정한 분석규칙에 따라 현재 프로젝트의 분석 결과를 보여준다. 넷째,

교사나 학생들은 분석된 결과를 보고 자신의 프로젝트가 어떤 구조로 어떤 스프라이트와 방송을 주고 받는 지에 대해 파악할 수 있다.

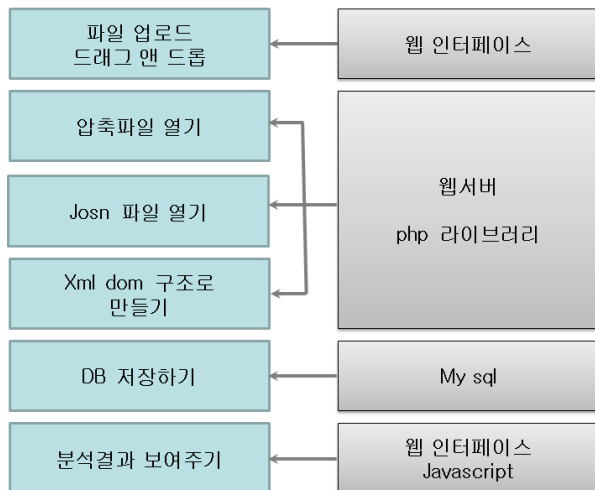


<그림 4> 시스템 구조도

교사들은 결과 분석 페이지를 통해 파악한 프로젝트의 구조를 보고 학생들에게 피드백 및 평가를 할 수 있으며, 학생들은 자신의 프로젝트에서 어떤 부분이 제대로 실행되지 않는지에 대한 디버깅의 힌트를 얻을 수 있다.

3.3 코드 분석 시스템 구현

앞 절의 시스템 설계에 따라 코드 분석 시스템을 <그림 5>와 같은 절차로 구현하였다.



<그림 5> 코드 분석 절차

<그림 6>은 스크래치 메타 데이터가 xml Dom 구조로 변환된 결과를 보여준다. xml 구조로 파싱한 데이터는 DB에 저장한다.

```

<scratch>
  <stage>
    <stack xpos="80" ypos="419">
      <block category="Control" name="doWaitUntil" type="Wrap">
        <param type="label"/>
        <param type="label"/>
      </block>
    </stack>
    <stack xpos="101" ypos="42">
      <block category="Events" name="whenGreenFlag" type="Hat"/>
      <block category="Control" name="doForever" type="Wrap">
        <param type="label">
          <block category="Sound" name="doPlaySoundAndWait" type="Stack">
            <param type="label">ParagonX9 - Chaos Fantasy.mp3</param>
          </block>
        </param>
      </block>
    </stack>
    <stack xpos="116" ypos="142">
      <block category="Events" name="whenIReceive" type="Hat">
        <param type="label">start</param>
      </block>
      <block category="Data" name="setVar:to:" type="Stack">
        <param type="label">Combo</param>
        <param type="number">1</param>
      </block>
    </stack>
  </stage>
</scratch>
    
```

<그림 6> 스크래치 코드 변환 xml Dom 구조

DB에 저장된 데이터는 <표 3>에서 제시한 코드 분석 알고리즘에 따라 분석된 결과를 보여준다. <그림 7>과 같이 각 항목별 점수를 제시하고 전체 점수에 따라 현재 학습자의 수준을 보여준다.

Pane은 초보자의 프로그래밍 시스템의 사용성 평가기준을 제시하면서 프로그램이 어떻게 진행되고 있는지, 쉽게 이해할 수 있는 피드백이 제공되어야 한다고 하였다[18]. 따라서 본 시스템에서는 스크래치 프로젝트의 구조를 쉽게 파악할 수 있도록 여러 가지 정보를 제공한다.

분석결과 화면의 처음부분에는 분석규칙에 따른 CT 개념 점수가 표시된다. 아래 부분에는 가장 기본적인 블록 수, 스택 수, 스프라이트 수와 같은 정보를 알 수 있도록 해준다. 또한 기존의 연구에서 밝혀진 중요한 요소인 변수, 리스트, 방송, 복사된 스택에 대한 빈도수를 보여준다. 다음 부분에서는 각 블록의 사용빈도수와 스크래치 블록 모양에 따른 사용빈도를 보여준다. 스크래치 블록의 종류는 이벤트를 발생하는 모자형태, 주요 명령어인 스택 형태, 불린 블록, 값을 표현하는 리포터 블록, C 형태인 C블록, 끝내는 블록인 캡 블록이 있다[19]. 맨 아래 부분에는 각 스프라이트 간의 관계를 파악할 수 있는 스프라이트 관계도 영역이 있다.

CT개념 점수: 14/21

CT개념 등급: 개발자

분야별 CT개념:

분야	점수
절차(Procedure)	2/3
반복(Repetition)	2/3
분기(Branch)	3/3
상호작용(UserInteractivity)	2/3
데이터표현(DataRepresentation)	2/3
추상화(Abstraction)	2/3
병렬 및 동기화(Parallelization and Synchronization)	1/3

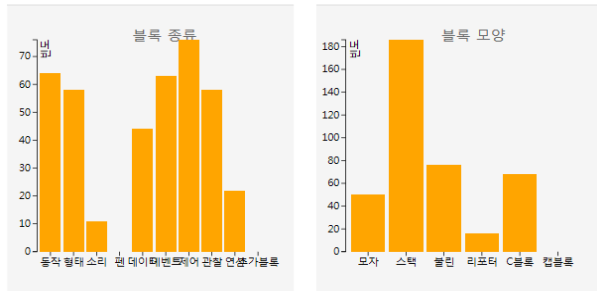
기본정보:



주요정보:



블록 빈도:

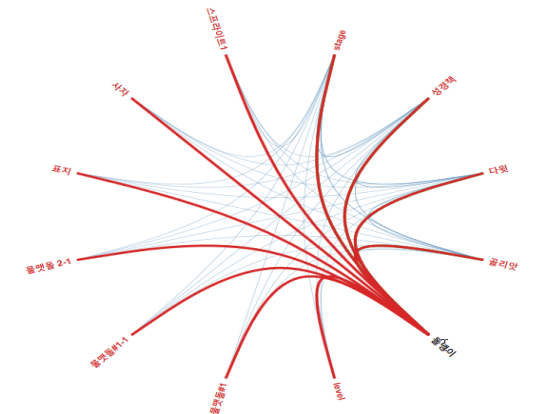
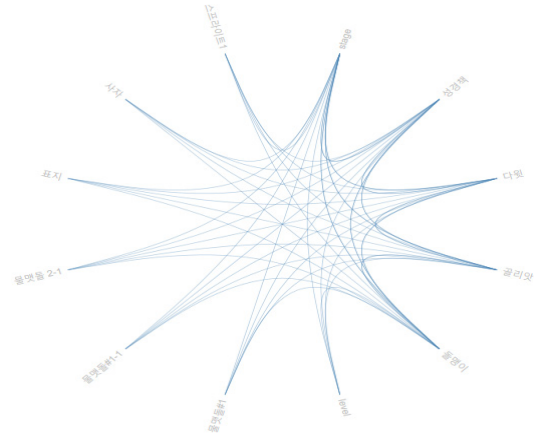


<그림 7> 코드 분석 결과 화면

<그림 8>과 같이 방송을 받는 스프라이트 간의 관계가 선으로 연결되어 있다. <그림 8>의 하단과 같이 스프라이트 이름을 클릭하면 방송을 받는 스프라이트의 연결선이 갈색으로 나타나 어떤 스프라이트가 방송을 받고 있는지 확인할 수 있다.

관계도 화면을 통해 학생들은 스프라이트들 간의 구조를 파악하여 디버깅에 활용할 수 있다. 스크래치의 경우 이벤트 기반이면서 OOP의 속성도 가지고 있으므로, 스프라이트들 간의 관계가 중요하다. 코드의 양이 많아질 경우 스프라이트들 간의 구조를 파악하기가 힘들며, 어떤 스프라이트가 연결되어 방송을 받는지 파악하기가 힘들다. 따라서 학생들이 자신의 프로젝트에서 스프라이트들

간의 구조를 파악하여 디버깅하는데 도움을 줄 수 있다. 또한 교사들이 학생들의 코드를 파악하여 피드백을 주는 도구로도 활용이 가능하다.



<그림 8> 스프라이트 관계도 화면

기존의 Scrape나 Drscratch와의 차이점은 다음과 같다. 첫째, 본 시스템은 한글 프로젝트나 한글 블록, 한글 스프라이트 명의 처리가 가능하여 우리나라의 교육환경에 적합하다. 둘째, 웹 인터페이스로 개발되어 언제 어디서나 사용이 가능하다. Scrape의 경우 PC용이고 메타데이터를 추출해서 사용해야 하는 단점이 있다. Drscratch는 웹 용량의 한계가 있어서 큰 용량의 프로젝트를 처리하지 못한다. 셋째, Scrape와 Drscratch에서 제공하지 않는 스프라이트 관계도를 제공한다.

4. 사용성 테스트

본 연구에서는 개발된 시스템의 사용성과 적합성 평가를 위해 실제 프로젝트 점수와의 비교 방

법과 휴리스틱 평가법을 적용하여 Triangulation을 실행하였다.

4.1 사용성 테스트

본 연구에서는 스크래치를 이용한 SW 교육 경험이 1년 이상 있는 6명의 교사들을 대상으로 휴리스틱 평가를 실시하고 <표 4>와 같이 5점 척도로 사용성 결과를 분석하였다. 각 전문가들은 시스템을 사용해 보고 실제 평가나 학생 피드백에 사용 가능한지 판단하여 기능성, 인지성, 편이성, 유용성, 활용성 측면에서 점수와 문제점을 기술하게 하였다.

사용성 테스트의 영역은 닐슨의 휴리스틱 평가의 영역과 장진철 외(2014)[20], 서영석(2007)[21]의 연구를 토대로 기능성, 인지성, 편이성, 유용성, 활용성을 평가 요소로 선정하였다.

<표 4> 휴리스틱 평가 결과

영역	내용	점수
기능성	이 시스템에서 스크래치 프로젝트의 구조를 파악할 수 있는가?	3.67
	이 시스템은 스크래치 프로젝트의 코드를 분석하기에 적절한가?	3.67
인지성	이 시스템에서 기능에 대한 요약적인 정보를 제공하는가?	4.0
	이 시스템의 레이아웃 구성이 적절한가?	3.67
	이 시스템의 구조가 직관적이고 일관적인가?	3.83
편이성	이 시스템의 사용법은 쉬운가?	5.0
	이 시스템의 사용법은 편리한가?	4.67
유용성	이 시스템은 스크래치를 이용한 학습에서 활용 가능한가?	3.83
	이 시스템은 스크래치를 이용한 학습에서 평가를 위한 도구로 활용 가능한가?	3.67
	이 시스템은 스크래치를 이용한 학습에서 학생 피드백을 위한 도구로 활용 가능한가?	3.83
활용성	스크래치를 활용한 학습에서 이 시스템의 사용을 추천하겠는가?	4.17
	스크래치를 활용한 학습에서 이 시스템을 활용하여 학생들의 코드를 파악하거나 평가에 활용한 생각이 있는가?	4.17

휴리스틱 평가결과 활용성, 편이성에서는 매우 긍정적인 반응을 보였으나, 유용성과 기능성 부분에서는 상대적으로 낮은 점수를 보였다.

특히 기능성 부분에서 나타난 문제점은 각각의 용어에 대한 설명이 없고 각 수치가 무엇을 분석하는지에 대한 내용을 알 수 없다는 것이었다. 또한 유용성 부분에서는 실제 수업과정에서 어떻게 활용할 수 있는지에 대한 안내가 미흡한 점이 문제점으로 나타났다.

4.2 시스템 평가의 적합성

본 연구에서 제시한 평가규칙이 실제 학습자의 작품 수준과 일치하는지 정확도를 평가하기 위해 교수자에 의한 작품 평가점수와 정확도를 검증하였다. 기존의 스크래치 프로젝트 작품 평가의 결과를 시스템의 평가 점수 결과와 상관관계가 있는지 분석하였다.

실험 프로젝트 작품은 2015년 1학기에 서울 C 대학 컴퓨터 비전공 학생들을 대상으로 실시한 스크래치 활용 교육에서 산출된 결과물을 활용하였다. 먼저 교수자 평가 점수를 상, 중, 하로 구분하고 구분된 영역별로 다섯 작품씩 무작위로 추출하였다. <표 5>와 같이 추출된 작품의 교수자 평가 점수와 시스템을 통해 평가된 점수와 비교 분석 하였다.

<표 5> 시스템 평가점수 비교

번호	교수자평가	시스템평가
1	30	17
2	28	14
3	28	15
4	30	14
5	28	15
6	26	9
7	25	13
8	25	18
9	26	16
10	24	15
11	18	2
12	20	3
13	22	13
14	18	7
15	20	12

교수자 평가점수의 경우 학생들의 참여나 태도 점수가 포함되어 있어서 시스템 점수와 직접적으로 비교할 수 없으므로 상관관계 분석을 실시하였다. <표 6>에서와 같이 실제 교수자 평가와 시스템 평가 결과에 대한 상관관계가 .741로 강한 상관관계가 나타나(유의수준 .01) 실제 스크래치 코드 평가에 활용할 수 있는 수준으로 나타났다.

<표 6> 상관관계 분석

		교수자평가	시스템평가
교수자평가	Pearson 상관계수	1	.741**
시스템평가	Pearson 상관계수		1

**< .01

5. 결론 및 제언

본 연구에서는 스크래치를 활용한 알고리즘 및 프로그래밍 학습에서의 디버깅 및 평가를 지원하기 위한 코드 분석 시스템을 개발하고 사용성을 검증하였다.

본 연구를 통해 개발된 시스템에 대한 결과는 다음과 같다.

첫째, 스크래치 코드 분석 시스템은 스크래치를 활용한 CT 교육에서 활용 가능하다[22]. 휴리스틱 사용성 테스트에서 실제 학습에서 활용성이 높게 나타났다. 따라서 본 시스템은 테스트와 디버깅 과정에서 오류를 파악하고 수정하는데 활용할 수 있으며, 학생들의 작품을 피드백하는 근거로 활용할 수 있다.

둘째, 스크래치 코드 분석 알고리즘은 학습자의 CT 개념 학습에 대한 평가도구로 활용할 수 있다. 본 시스템에서 제안한 코드의 수준 판별 알고리즘은 실제 스크래치 프로젝트 15개를 분석한 결과 교수자 평가 점수와 높은 상관관계가 나타나 학생들의 CT 개념 평가를 위한 도구로 사용이 가능함을 알 수 있다.

셋째, 개발된 시스템의 교육현장 활용을 위한 사용성 평가에서 휴리스틱 평가결과, 모든 영역에서 3.6이상 만족도를 보여 사용성이 양호하게 나타났다. 시스템의 보완 사항으로는 각 점수에 대한 설명과 실제 수업에서의 활용 방법에 대한 안

내가 부족하다는 점이 밝혀졌다.

본 연구에서 개발한 스크래치 코드 분석 시스템의 활용을 위해 다음과 같이 제언한다.

첫째, 본 시스템은 학생들의 스크래치 작품 코드만을 분석하는 것이므로 학생들이 직접 제작한 작품을 대상으로 해야 한다. 수업시간에 학생들이 직접 만든 작품을 분석하는 도구로 사용하는 것이 좋다.

둘째, 평가에 활용할 때에는 본 평가도구의 결과로만 학생들의 역량을 평가하지 않는 것이 좋다. 본 평가도구에서 보여주는 결과는 학생들이 만든 코드의 수준을 판단하는 데에 활용하거나 작품향상을 위한 피드백의 자료로 활용해야 한다. 즉, 현재의 도구로 학생들의 CT 역량의 전반을 평가하기는 어려우므로 여러 평가방법과 함께 사용하는 것이 좋다.

셋째, 본 연구에서는 시스템 개발에 초점이 맞춰져 있어서 초중등 학생들의 코드 수준의 판별을 위한 많은 양의 데이터 분석이 이루어지지 않았다. 따라서 본 연구에서 제시한 수준 판별 알고리즘을 일반화하기 위해서는 많은 양의 데이터로 검증해야 한다.

참고 문헌

- [1] Nacce & CAS (2014). Computing in the national curriculum: a guide for secondary teachers. 2015. 7. 14일 검색 http://www.computingatschool.org.uk/data/uploads/cas_secondary.pdf
- [2] Tucker, A. et al. (2006). *A Model Curriculum for K-12 Computer Science*: New York, CSTA.
- [3] 김경훈 외 (2015). 정보과 교육과정 시안 개발 연구. 한국교육과정평가원. 연구보고 CRC 2015-17
- [4] 김영애 외 (2015). 소프트웨어 교육 운영 지침 개발 연구. 한국교육학술정보원. 연구보고 CR 2015-3

- [5] 김수환, 채정병 (2014). 교육용 프로그래밍 언어의 동향 분석 및 교수-학습 사례. 한국 교육학술정보원. 연구자료 RM 2014-12
- [6] 김수환, 이원규, 김현철 (2009). 개정된 정보 교육과정에서 교육용프로그래밍언어의 교육 적 적용방안. **컴퓨터교육학회논문지**, 12(2), 23-31.
- [7] Sylvia Libow Martinez & Gary Stager. (2013) Invent to learn : making, tinkering, and engineering in the classroom. Torrance, Calif. : Constructing Modern Knowledge Press
- [8] 장혜선, 최숙경, 전수진, 염용철, 이원규 (2007). 에러 피드백 기반의 초보자를 위한 프로그래밍 학습 지원 시스템. **한국컴퓨터교육학회논문지**, 10(6). 1-10.
- [9] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Paper presented at annual American Educational Research Association meeting*, Vancouver, BC, Canada.
- [10] 최정원, 이영준 (2014). 프로그래밍 학습에서 학습자의 어려움 분석. **한국컴퓨터교육학회 논문지**, 17(5), 89-98.
- [11] 스크래치 위키 2015. 8. 24일 검색 http://wiki.scratch.mit.edu/wiki/Scratch_Wiki_Home
- [12] Kirkov, R., & Agre, G. (2010). Source Code Analysis - An Overview. *BULGARIAN ACADEMY OF SCIENCES CYBERNETICS AND INFORMATION TECHNOLOGIES*, 10(2), 60-77.
- [13] Scrape. 2015. 8. 14일 검색 <http://happyanalyzing.com/downloads/scrape-Local>
- [14] Drscratch. 2015. 8. 14일 검색 <http://drscratch.programamos.es/>
- [15] Martin, T., & Fields, D. (2014). Macro data for micro learning: Developing FUN! for Automated Assessment of Computational Thinking in Scratch. *Poster presented at the 2014 Cyberlearning Summit, Madison, WI.*
- [16] Nielsen J. "Ten Usability Heuristics" USEIT.com, 1994. http://www.useit.com/papers/heuristic/heuristic_evaluation.html
- [17] Nielsen J. (1994). "how to conduct a Heuristic Evaluation" USEIT.com, http://www.useit.com/papers/heuristic/heuristic_list.html
- [18] Pane, J. F. & Myers, B. A. (1996). Usability Issue in the Design of Novice Programming Systems. Technical Report, Carnegie Mellon University.
- [19] 스크래치 블록 종류 2015. 8. 25일 검색 http://wiki.scratch.mit.edu/wiki/Blocks#Block_Shapes
- [20] 장진철, 이슬기, 정석환, 정치훈, 윤완철 (2014). WeStudy : 집단지성 기반 학습자료 북마킹 서비스 구현 및 사용성 평가. **한국지능정보시스템학회 춘계학술대회 논문집**, 257-263.
- [21] 서영석 (2007). **웹기반 학습 사이트 사용성 평가 도구 개발**. 박사학위논문, 한양대학교 대학원
- [22] 김수환 (2015). Computational Thinking 역량에 대한 학습자 평가를 위한 스크래치 코드 분석. **한국컴퓨터교육학회논문지**, 18(6). 25-34.



김수환

1999 인천교육대학교(교육학학사)
 2006 경인교육대학교
 컴퓨터교육과(교육학석사)
 2011 고려대학교
 컴퓨터교육과(이학박사)
 2013~2014 경인교육대학교 겸임교수
 2014~현재 총신대학교 교양교직과 조교수
 관심분야: 컴퓨터교육, EPL, 컴퓨터적사교(CT),
 CSCL, Computational Literacy

E-Mail: skim@csu.ac.kr