학습용 게임을 위한 한국어 인터프리터 언어의 설계 및 구현

윤경섭*, 조강현O
*인하공업전문대학 컴퓨터정보과,
O인하공업전문대학 컴퓨터정보과
e-mail: ksyoon@inhatc.ac.kr*, jokh980324@naver.comO

Design and Implementation of Korean based Interpreter Language for Learning Games

Kyung Seob Yoon*, Kang Hyeon JoO
*Dept. of Computer Science, Inha Technical College,
ODept. of Computer Science, Inha Technical College

● 요 약 ●

코딩교육과 게임을 활용한 교육에 대한 관심이 증가하고 있다. 그러나, 코딩 교육 도구는 블록형 코딩 도구에 치우쳐져 있어, 교육과 실제 프로그래밍 언어 사이에는 큰 차이가 발생하게 된다. 이 차이를 좁히기 위해 이 논문에서는 한국어 인터프리터 언어를 사용한 학습용 게임의 설계 및 구현 방법을 제공하며, 한국어 인터프리터를 사용한 코딩 교육용 게임의 장점과 기존 코딩 학습 방법과의 차이점을 제시한다. 이를 통해 자발적이고 효과적인 코딩 교육을 기대할 수 있을 것이다.

키워드: 인터프리터(Interpreter), 컴퓨터 게임(Computer Game)

I. Introduction

조기 코딩 교육의 필요성이 커지고 있는 가운데, 여러 코딩 교육 도구가 개발되었다. 코딩 교육 도구를 사용한 코딩 교육은, 학생의 컴퓨팅적 사고를 강화시킴과 동시에, 실제 프로그래밍 언어를 배우기 위한 밑바탕이 되어야 한다. 하지만, 이런 목표를 달성하기 위해서는 수업 시간에만 진행하는 것 보다는 학생 스스로가 흥미를 가지고 긴 시간 고민하며 학습하는 것이 효과적이다. 그렇기 때문에, 게임을 통한 조기 코딩 교육은 일찍이 연구되어 왔으며, 주로 블럭형 코딩을 활용해 왔다[1].

본 논문에서는 학습용 게임을 위한 한국어 인터프리터 언어를 설계 및 구현하면서 조기 코딩 교육의 도구로 사용하는 방법을 제공하고자 한다

II. Preliminaries

2.1. Related works

2.1.1 최근 동향

최근 초등학교에서 SW교육을 의무화하는 계획을 준비중일 정도로 코딩 교육의 관심이 높아지고 있다.

그에 따라서 Scratch와 같은 많은 코딩 교육 도구들이 개발되었다[2].

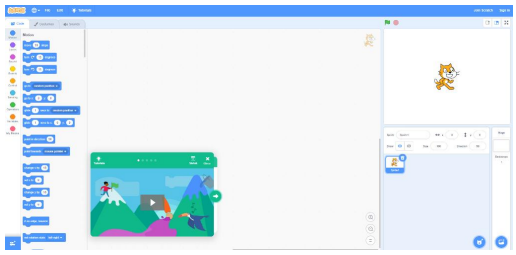

Fig. 1. Block-base Coding Tool

2.1.2 블록형 코딩

해외의 MIT 미디어 랩이 개발한 스크래치(Scratch)와 국내의 네이버 커넥트재단이 개발한 엔트리로 대표되는 블록형 코딩 형식은 코딩을 시작하는 학생이 가장 쉽게 배울 수 있다는 장점 때문에 초등학생의 코딩 교육용 도구로 사용되어 왔다[3].

III. Design & Implementation

3.1. 코딩게임을 위한 한글 인터프리터의 설계

3.1.1. 코딩게임에서의 인터프리터의 활용

게임은, 놀이이다. 그리고, 조지 산티아나는 놀이를 다음과 같이 정의한다. "플레이는 그 자체를 위해 자발적으로 행해지는 모든 것이다"[4] 장기간의 사고를 통해 컴퓨팅적 사고를 키워나가야 하는 코딩 교육에서, 학습이 끝나면 더이상 생각하지 않게 되는 기존의 교육보다 자발성을 가진 게임은 확실한 이점을 가진다[5].

게임 내부에서의 코딩은 컴파일러를 사용해야 할 정도로 최적화될 필요가 없으며, 컴파일을 위한 대기시간이 없으며, 실행 중간에 사용자에게 즉시 피드백을 줄 수 있다는 점에서 인터프리터 언어를 통해 진행하는 것이 효과적일 것이다.

3.1.2. 한국어 인터프리터

블록형 코딩 교육 도구는 충분히 연구되고, 개발되었다.[1]. 하지만, 코딩 교육은 블록형 코딩의 단계에서 멈추어서는 안 된다. 스크래치/엔트리와 앱인벤터를 거친 학생은 이후 문자를 사용하는 교육 도구를 사용하게 될 것이다. 이 구간에 사용할 수 있을 도구 또는 언어라고 한다면, Python이나 Lua스크립트를 생각해볼 수 있을 것이다. 기존 한글, 블록형 코딩 환경에서 갑작스레 영어 문자 코딩으로 넘어가는 것은 너무 급격한 난이도 상승으로 느껴질 수 있으며, 이는 코딩에 대한 관심 하락으로 이어질 수 있다.

그렇기 때문에, 블록형 코딩 도구와 영문 프로그래밍 언어 사이의 간극을 매울 한국어 프로그래밍 언어의 필요성이 생긴다.

3.1.3. 세부 설계

교육용 게임을 위한 한국어 인터프리터 언어는 Python등의 실제 프로그래밍 언어의 학습 이전에 적용하는 것이 가장 효과가 좋다고 판단하였다. 그렇기 때문에, Python과 형태적으로 유사한 형태로 설계하였다.

이를 위해서 공백 들여쓰기를 통해 블럭을 구분하고, 함수와 제어문의 인수는 소괄호로 묶도록 하였다.

```
변수1=1
반복(변수1 <5)
   함수(인수1,인수2)
   변수1=변수1+1
```

Fig. 2. Interpreter Design

3.1.4. 인터프리터 구성

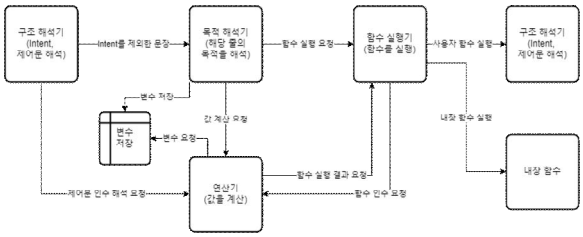

Fig. 3. Interpreter Request

Fig. 3은 인터프리터 내부의 구성 요소들과 한 줄을 실행하기까지의 호출 관계를 나타낸다.

3.2. 구현

3.2.1. 구조 해석기

구조 해석기는 입력된 전체 스크립트를 각 줄로 분리하고, 들여쓰기와 제어문을 처리하는 영역이다. 처리할 라인이 제어문이면 다음 줄에 들여쓰기가 증가하여야 함을 기록하고, 제어문에 필요한 처리를 실행하여, 들여쓰기 스택(Stack)에 들여쓰기가 증가한 원인을 기록한다. 그 외의 문장이라면, 들여쓰기를 분리한 실제 실행할 문장만을 목적 해석기로 전송한다.

들여쓰기의 증가는 새 블럭으로의 진입을 의미하며, 들여쓰기가 증가하기 위해서는 제어문이 필요하다. 제어문이 없는 불필요한 들여쓰기증가는 오류를 발생시켜 사용자에게 코드를 고칠 것을 요구한다.

들여쓰기의 감소는 블럭에서의 탈출을 의미하며, 들여쓰기 스택에서 들여쓰기가 증가했던 원인을 불러와 적절한 처리를 하게 된다.

3.2.2. 목적 해석기

목적 해석기는 해당 문장의 목적을 순수 함수 실행과 변수 생성/대입으로 지정한다. 변수 생성/대입의 경우, 변경될 변수를 기억하고, 변수 목록에 새 변수를 생성할 것을 요구하거나, 변수의 대입에 대비하여, 들어와야 할 변수의 타입을 예측한다. 이 경우, 뒤의 문장을 연산기로 전송하며, 연산기에서 반환된 값을 변수에 저장한다.

함수 실행의 경우, 변환기를 거치지 않고 함수 실행기로 직접 내용을 전달하여, 변수의 조작 없이 함수의 실행만을 처리하도록 한다.

3.2.3. 연산기

연산기는 단일 수식의 단일 결과값을 반환하는 장치로, 수식을 확인하여, 함수가 있다면 함수 실행기로, 변수가 있다면 변수 테이블에 요청하여, 연산 가능한 원시 자료형(Primitive Type)으로만 이루어진 수식을 만들어 수식을 계산한 뒤, 결과값을 반환한다.

3.2.4. 함수 실행기

함수 실행기는, 들어온 함수의 몸체를 식별하고, 인수를 분리하여, 인수를 원시 자료형으로 전달하기 위해, 각 값을 연산기에 요청하여, 내장 함수나 사용자 지정 함수에 원시 자료형 값을 인수로 전달할 수 있도록 가공하는 역할을 수행한다.

또한, 함수가 사용자 지정 함수인 경우, 함수 저장소에서 문자열로 이루어진 함수를 추출하여, 이를 실행할 새로운 해석기 객체를 생성하고, 새 객체의 구조 해석기에 지정 함수의 실행과 결과값을 요청한다.

3.2.5. 구현 예시

Fig. 4. Implementation Example

Fig. 4는 Intent와 한글로 이루어진 반복문이 포함된 예시 스크립트와 실행 결과이다.

```
변수1=0
변수2=0
반복(변수2 작다 11)
   변수1=변수1+변수2
   변수2=변수2+1
```

Fig. 5. Example Code

또한, 비교연산자 등을 한글로 변환할 수 있으며, 화면상의 캐릭터를 조작하는 명령어 등을 추가하여, 명령어로 화면상의 캐릭터나 실제 로봇 등을 조작할 수 있다.

3.2.6. 게임에의 적용

해당 인터프리터 언어를 사용하여, 다음과 같이 적용할 수 있다.

1) 타이쿤/퍼즐 게임

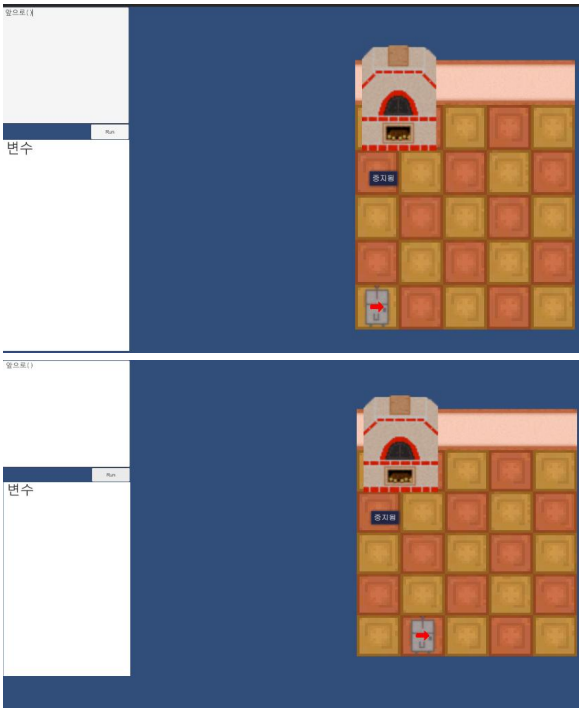

Fig. 6. Tycoon Game Example

플레이어는, 기계 장치들을 코딩하고, 코딩된 장치들을 사용하여 생산물을 제작한다. 플레이어는 이 제작물을 판매하여 게임 내 재화를 획득하고, 게임 내 재화를 소모하여 새로운 제작법을 획득한다.

처음부터 장치에 복잡한 코딩을 요구하게 되면, 플레이어는 게임이라기 보다는 학습 소프트웨어로 느낄 가능성이 클 것이다. 그렇기 때문에, 처음에는 장치들이 플레이어의 입력에 의존하여 움직이도록 시작한다. 하지만, 게임 중반에 접어들게 되며, 조작해야 할 장치들이 늘어나게 되면, 플레이어 스스로가 코딩과 자동화를 선택지에 두게 될 것이다.

플레이어가 자신이 코드를 짜지 않고, 누군가의 코드를 복사하여 시작하는 것을 선택할 수도 있다. 하지만, 이는 서로 다른 각자의 환경에서 정확히 적용되지 않을 것이며, 플레이어에게 자연스럽게 코드의 분석과 수정을 요구하게 될 것이다.

2) MMORPG 마이룸 확장

최근 MMORPG에서 개인의 표출은 플레이어들이 중요하게 생각하는 요소 중 하나로 자리잡았다. 이는 아바타등의 캐릭터 커스터마이징 외에도, 마이룸과 같은 형태로 개인의 공간을 꾸미고 타인을 초대하는 형태도 있다. 이 개인 공간에 있는 장치들에 스크립팅 기능을 넣는다면, 좀 더 자유로운 확장과, 유저가 스스로 콘텐츠를 생성하는

선순환이 이루어짐과 동시에, 사용자들에게 자연스럽게 코딩 교육 효과를 나타낼 수 있을 것이다.

IV. Conclusions

본 논문에는 블럭형 코딩 교육 이후의 코딩 교육의 도구로 게임을 위한 한국어 인터프리터 언어를 제시하고 구현하였다. 게임에 적용된 한국어 인터프리터는 코딩교육의 보조 교제로써 자발적인 코딩 학습의 효율성을 높이는데 도움이 될 것이다.

또한, 학습용 게임 외에도, 퍼즐 게임, 메타버스 게임 등 인게임 코딩이 필요한 여러 분야에서 응용할 수 있는 분야는 무궁무진할 것이다.

REFERENCES

[1] Chang-Ho An, Kibbm Lee, and Seok-Jae Moon, "Programming learning method for beginner based on Entry Block-base/Text-based coding"

[2] Mit Media Lab, "Scratch "https://scratch.mit.edu/

[3] Naver Connect, "Entry "https://playentry.org/

[4] Jesse Schell, "The Art of Game Design", Acorn Trans. p72, 2008.

[5] Hong Soo-bong and Park Jae-hyun, "Analysis of learning effects of classes applied with gamification." Video Culture Content Research, 2019.