

수학 용어 사전 프로그램 제작

1. 과제 개요(선정 배경 및 취지, 내용 등)

수학은 초등학교 교육과정부터 지속해서 중요시되는 중등교육과정의 필수과목이다. 하지만 현재 우리나라에서 사용하고 있는 수학용어는 과거 중국에서, 근대 이후에는 일본을 통해 받아들여 대부분 한자어로 이루어져 있다.* 따라서 학생들이 직관적으로 그 의미를 이해하는 데 어려움이 있다. 최근 위와 같은 수학용어들을 우리말로 바꾸는 작업들이 행해지고 있지만, 여전히 일부 용어들은 본래의 의미를 파악하기 어렵다. 이는 수학에 대한 이해도와 흥미를 떨어뜨려 학습을 이어나가는 데에 방해된다. 또한 기존 수학 용어 어플리케이션은 단어 자체의 뜻보다는 공식 위주의 설명 또는 너무 많은 부가적 정보까지 담고 있어 학생들이 필요한 내용을 분별해서 습득하기 어렵다.

따라서 본 과제에서 중학교 학생들이 수학용어의 본질적인 의미를 파악하여 올바르게 이해하도록 용어사전을 제작하고자 한다. 사전은 인터넷 연결 없이 사용 가능한 프로그램으로 제작하여 접근성을 높이고, 중학교 교과서에 수록된 모든 용어를 정의와 예시, 예제 등의 내용으로 구성하여 학생들이 사전을 통해 수학용어에 더 쉽게 다가갈 수 있도록 한다.

*허민, 수학용어의 개선 방향에 대한 소고, 수학교육논문집 제27집 제4호, 2013

2. 과제 수행(연구) 방법

(1) 중등과정 학년, 학기별 수학용어 목록 정리

- 비상교육, 천재, 신사고, 동아출판 교과서를 참고하여 용어 목록 정리
- 중학교 1학년, 2학년은 2015 교육과정으로 3학년은 2009 교육과정에 맞추어 단어를 선정
- 이를 python 프로그램 구현 시 사용할 수 있도록 excel로 리스트화

(2) 사전 용어별 설명 파일 제작

- 학생들의 혼란이 없도록 교과서 서술방식과 같게 서술하여 용어 풀이 작성
- 용어와 관련된 역사자료와 시각자료 및 문제 등을 탭으로 나누어 제공할 수 있도록 작성
- txt 파일 제작 ; 파이썬 프로그램을 사용할 때 제목과 내용 사진을 리스트로 인식할 수 있도록 용어별 txt 파일을 제작
- ppt 파일 제작 ; 용어별 탭마다 제작함으로써 미리 이미지화된 상태를 확인하고 내용 첨삭을 할 수 있도록 함.

(3) Python과 Pygame을 활용한 컴퓨터 프로그램 제작

- python 3.7ver 과 pygame 1.9.4 모듈을 사용하여 구현
- 검색, 북마크, 탭 등을 구현하는 함수를 작성

3. 과제 수행(연구) 결과

1) 중학교 1학년~3학년 교과서 속 용어 선정 및 조사

중학교 학생들을 대상으로 하는 수학사전 프로그램을 제작하기 위해 먼저 교과서를 기반으로 정의가 필요한 용어들을 선정하였다. 2019년 1학기를 기준으로 실제 교육현장에서 사용되고 있는 중학교 1학년, 2학년은 2015 교육과정으로 3학년은 2009 교육과정 교과서를 기반으로 1학년 1학기 53개, 1학년 2학기 38개, 2학년 1학기 16개, 2학년 2학기 17개, 3학년 1학기 16개, 3학년 2학기 10개로 총 150개의 용어를 조사하였다. 용어풀이는 되도록 교과서를 참고하되, 교과서의 내용이 외에 흥미를 줄 수 있는 내용이 있을 시에는 부가적인 내용을 첨가하였다.

사전에 들어가는 내용은 정의, 예, 활용법, 퀴즈 등이 있었으며 각각의 용어에 맞게 필요한 내용을 작성했다. 예를 들어, 소수(약수가 1과 자기 자신뿐인 수)의 경우 그림2와 같이 정의, 다른 의미의 소수, 에라토스테네스의 체, 퀴즈를 탭으로 분류했으며, 학생들이 이해가 쉽도록 간결하게 설명하고 PPT를 활용하여 제작한 이미지를 첨부했다. 용어 용어리스트는 csv 파일로 작성, 용어에 들어갈 내용과 이미지는 각각 txt, png 파일로 저장하여 프로그램에서 일괄적으로 처리할 수 있도록 하였다.

◆ 소수는 무엇일까?

다음 표는 자연수 2, 3, 4, 5, 6, 7의 약수를 나타낸 것이다.

자연수	2	3	4	5	6	7
약수	1, 2	1, 3	1, 2, 4	1, 5	1, 2, 3, 6	1, 7

1 표의 빈칸을 알맞게 채우시오.
2 위의 자연수 중에서 약수가 1과 자기 자신뿐인 수를 모두 말하시오. 2, 3, 5, 7

위의 개념 열기에서 2, 3, 5, 7의 약수는 1과 자기 자신뿐이다.

이와 같이 1보다 큰 자연수 중에서 약수가 1과 자기 자신뿐인 수를 **소수** 한다. 그리고 소수가 아닌 1보다 큰 자연수를 **합성수**라고 한다.

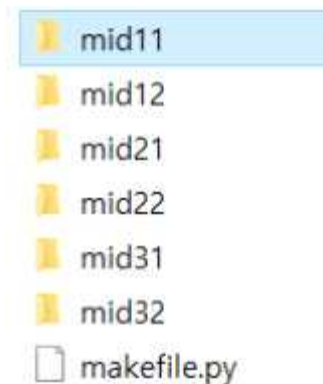
<그림1> 비상교육 중학 수학 1 교과서 중

mid11_1 - 메모장

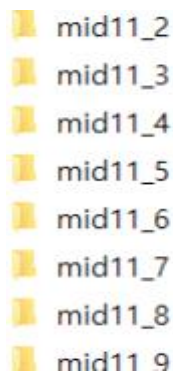
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

정의, 1보다 큰 자연수 중에서 약수가 1과 자기 자신뿐인 수, 0
다른 의미의 소수, 초등학교에서 배운 $0.1 < c < 0.02$ 등과 헷갈리지 않도록 한다. $<n> <n>$ 중학교에서 나오는 소수는 $<n> 1$ 보다 큰 자연수 중에서 약수가 1과 자기 자신 뿐인 수이다, 0
에라토스테네스의 체, 고대 그리스의 수학자인 에라토스테네스가 고안한 소수를 찾는 방법 $<n>$ 마치 체로 걸러 내듯 소수만 걸러 낸다고 하여 $<n>$ '에라토스테네스의 체'라고 한다., 1, mid11_1_1

<그림2> '소수' txt 파일 (mid11_1.txt)



<그림3> wordfie 파일



<그림4> mid11 파일 일부

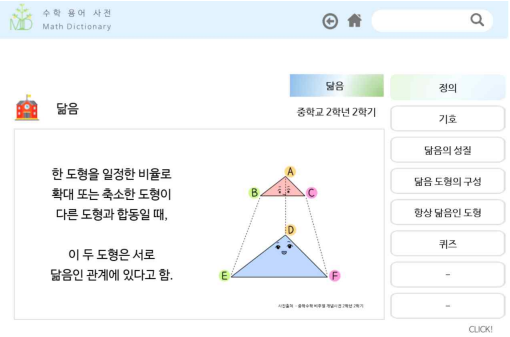



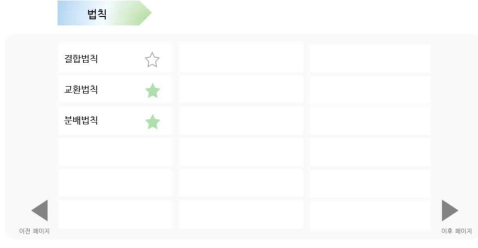


<그림5> mid11_2 파일 일부

2. 프로그램 구현

프로그램에 필요한 사전내용은 일괄적으로 빠르게 불러올 수 있도록 png와 txt 파일의 이름을 동일한 양식으로 작성했다. 텍스트 파일(txt)은 메모장, 이미지 파일(png)은 Power Point를 이용하여 제작했다. 이후 Python을 통해 사전 기능을 하는 프로그램을 구현했다. 사용되는 프로그램은 Python 3.7 ver 이며 모듈은 Pygame 1.9.4를 사용했다. 다음은 실제로 '수학용어 사전'을 실행했을 때의 모습을 캡처한 사진과 설명이다.

	
<p>[페이지1] 로딩 프로그램을 시작하기 전 2초간 로딩페이지가 뜬다.</p>	<p>[페이지2] 메인 학년별로 용어를 나누어 보거나, 전체보기가 가능하며 북마크 기능을 사용할 수 있다. 검색기능도 사용가능하다. (검색기능 - 페이지8)</p>
	
<p>[페이지3] 용어리스트 학년 또는 전체보기 버튼을 클릭하게 되면 용어리스트들이 나온다. 각각의 용어 중 표시해두고 싶은 용어가 있을 시에는 용어 오른쪽에 있는 별표를 클릭하여 표시할 수 있고, 표시해 둔 용어는 따로 모아볼 수 있다. (북마크 - 페이지 4)</p>	<p>[페이지4] 북마크 용어리스트 페이지에서 표시해둔 용어만 따로 모아서 볼 수 있다.</p>

	
<p>[페이지5] 용어 정의</p> <p>각각의 용어를 클릭하면 먼저 간단한 정의와 예시 또는 그림이 표시된다.</p>	<p>[페이지6] 용어 탭</p> <p>각각의 탭을 누르면 선택된 탭이 변경되면서 탭에 맞는 내용과 이미지가 표시된다.</p>
	
<p>[페이지7-1] 퀴즈</p> <p>탭 중에서 퀴즈 탭을 클릭할 경우 각 용어에 맞는 퀴즈가 나온다. 답을 입력한 뒤 확인버튼을 클릭하면 정답과 오답을 표시한다.</p>	<p>[페이지7-1] 정답/오답</p> <p>퀴즈페이지에서 입력한 답이 정답일 경우 왼쪽, 오답일 경우 오른쪽과 같이 표시되며, ⊗를 클릭한 후 정답을 재입력 할 수 있다.</p>
	<p>[페이지8] 검색</p> <p>메인페이지에서 오른쪽 상단에 있는 '돋보기' 아이콘을 클릭한 후 원하는 용어를 입력하면 검색한 용어를 포함하거나 일치한 용어들을 볼 수 있다. 왼쪽 예시는 '법칙'을 검색하여 나온 용어들이다.</p>

3. 프로그램 세부 코드

위와 같은 프로그램을 구현하기 위해 여러 함수가 사용되었다. main.py파일 내부에 있는 각 함수의 역할은 다음과 같다.

make_wordlist(grade, testword)	만들고자 하는 list의 종류를 grade로 받아 grade = "전체"인 경우 allwordlist.csv파일의 모든 단어를 불러오고, grade가 학년인 경우 각 학년에 해당하는 단어를, 북마크인 경우 bookmarklist에 들어있는 단어들과 일치하는 것들을 불러온다. 검색을 하는 경우 pygame에서 한글 입력을 인식하지 못하기 때문에 단어를 음소별로 입력받아 allwordlist2.csv에서 일치하는 단어를 찾고, 불러온다. 불러온 단어들을 wordlist에 입력하고, 이를 return한다.
bookmarktest(word, x, y)	입력받은 단어 (word)가 bookmarklist에 있는지 확인하여 있으면 북마크 표시 아이콘을 star_button2.png로, 없으면 star_button.png로 return한다.
word_search()	키보드가 눌리는 것을 인식하여 눌리는 경우 해당 하는 자판의 글자를 searchword에 추가한다. Backspace 자판이 눌리는 경우 searchword의 글자를 뒤에서부터 하나씩 제거하고, enter 자판이 눌리는 경우 typed = True로 하여 word_list('검색', searchword)가 실행되도록 한다. cursortime을 구현하여 화면에 searchword를 출력할 때, searchword의 맨 뒤에 깜빡이는 cursor가 함께 구현된다.
quiz(number)	퀴즈가 나오는 화면을 구성한다. word_search()에서 구현한 cursor와 유사하게 사용자의 답을 입력받고, 이를 기존 단어의 txt파일에 저장되어있던 정답과 비교하여 일치하는 경우 right_popup.png을 화면에 띄우고, 일치하지 않는 경우 wrong_popup.png을 띄운다. worddata, worddetail과 마찬가지로 tablist를 화면에 출력되도록 하여 다른 탭으로 이동할 수 있다.
worddetail(number)	화면을 3가지의 경우로 나누어 구성한다. 입력받은 값이 0인 경우 사진이 없이 설명글만 출력되도록 하고, 1인 경우 상단에 설명글, 하단에 가로이미지가 출력되도록 한다. 2인 경우 좌측에 설명글, 우측에 정사각(세로) 이미지가 출력되도록 한다. 설명글의 경우 1. 2. 3. 등 숫자+의 형태로 시작하는 문장에 대해서는 해당 문장들끼리 좌측정렬을 이루고, 이외의 문장들에 대해서는 가운데정렬을 이루도록 하여 출력한다. 전체 화면의 우측에는 tablist를 출력하여 다른 탭으로 이동할 수 있는 버튼을 구현하였고, 현재 위치한 탭의 경우 tab_button2.png가, 이외의 탭은 tab_button.png가 출력된다. 글상자 좌측상단에는 탭의 명칭이 출력된다. 글상자의 우측상단에는 모든 탭에서 동일하게 단어명이 출력되고, 그 하단에 해당 학년과 학기가 함께 출력된다. 전체 화면의 우측상단에는 뒤로가기와 홈 버튼이 있고, 각각의 경우 list로 돌아가거나 메인 화면으로 이동하도록 구현한다.
worddata(word)	worddetail(number)와 대부분 같은 방법으로 구현되어있으나, 글상자의 좌측상단에 탭의 명칭인 '정의' 대신 해당 단어명이 출력된다. 또한 입력받은 word를 wordnow라는 변수로 설정하고 global 변수로 지정하여 worddata(word)함수의 밖에서도 계속하여 사용할 수 있다.
wordlist(grade, testword)	make_wordlist()함수를 통해 grade에 따라 만들어진 list를 화면에 구현하는 함수이다. page라는 변수를 통해 한 화면에 최대 18개의 단어가 표시되도록 하였으며, left_button과 right_button을 구현하여 이들을 통해 page를 이동할 수 있도록 하였다. 화면에 표시된 각각의 단어는 우측에 bookmark 아이콘이 함께 출력되는데, 이는 bookmarktest(word, x, y)함수를 통해 bookmarklist안에 있는 단어인지 아닌지를 판단하여 화면에 출력해준다. bookmarklist안에 있는 단어의 아이콘을 클릭하게 되면, 해당 단어는 bookmarklist에서 삭제되고, 반대로 bookmarklist안에 없는 단어의 아이콘을 클릭하면 해당 단어는 bookmarklist에 추가된다.
startGame()	프로그램을 실행하면 등장하는 첫 화면을 구현하는 함수이다. 가운데에 중학교 수학1, 중학교 수학2, 중학교 수학3의 버튼을 구현하여 클릭되면 wordlist(grade, testword)함수를 통해 각각의 학년에 해당하는 list로 이동한다. '북마크 모아보기'와 '용어 전체 보기'를 클릭하면 bookmarklist에 들어있는 단어들만 있는 list나 전체 용어가 모두 들어있는 list를 확인할 수 있다. 이 화면에서 우측상단에 있는 검색버튼을 누르면 word_search()함수가 실행되어 커서가 깜빡이며 검색할 단어를 입력받는 것을 확인할 수 있다.

<표2> maim.py파일 내부 함수별 기능

main.py파일 이외에도 button.py, sentence.py에서는 button과 sentence class를 구현하였고, variable_sh.py와 image_path_sh.py에서는 공통으로 불러오는 변수, 이미지들을 지정하였으며 initGame.py에서는 프로그램이 화면에 구현되기 전에 필요한 코드들을 행한다. makelist.py에서는 각 단어의 txt파일을 열고, 읽어 list화 한다. handle.py파일에서는 pygame으로 구현되지 않는 한글을 화면에 구현하기 위해 알파벳으로 입력받아진 단어를 한글단어로 변형하는 함수 등이 구현되어 있다. 코드는 각각의 기능에 맞게 분리하여 함수로 저장하고, main.py에서 불러와 사용한다.

```

13 screen = pygame.display.set_mode((pad_width,pad_height))
14 back_button = Button((400,25), 'back_button.png')
15 home_button = Button((50,25), 'home_button.png')
16 search_button = Button((445,25), 'search_button.png')
17
18 os.chdir(file_path)
19 if os.path.isfile('bookmark.txt'):
20     file = open('bookmark.txt','r')
21     for lineContent in file:
22         bookmarkList.append(lineContent.strip().split(','))
23     file.close()
24     bookmarkList = bookmarkList[0][1:-1]
25
26 def make_wordlist(grade, testword = ''):
27     os.chdir(wordlist_path)
28     fileMatrix = []
29     wordlist = []
30     if grade == '중1':
31         with open('allwordlist1.csv','r') as file:
32             for word in file:
33                 fileMatrix.append(word.strip().split(','))
34                 for i in range(len(fileMatrix)):
35                     if testword in fileMatrix[i][0]:
36                         wordlist.append(fileMatrix[i][1])
37     else:
38         with open('allwordlist.csv','r') as file:
39             for word in file:
40                 fileMatrix.append(word.strip().split(','))
41                 for i in range(len(fileMatrix)):
42                     if grade == grade_list[i][0]:
43                         if fileMatrix[i][1] in bookmarkList:
44                             wordlist.append(fileMatrix[i][1])
45                     elif grade == grade_list[i][0]:
46                         wordlist.append(fileMatrix[i][1])
47                 else:
48                     for j in range(9):
49                         if grade == grade_list[i+j][0]:
50                             if fileMatrix[i+j][1] == grade_list[i+j][1] or fileMatrix[i+j][1] in bookmarkList:
51                                 wordlist.append(fileMatrix[i+j][1])
52                     break
53     return wordlist
54
55 if __name__ == '__main__':
56     make_wordlist(grade, testword)
57
58 PyCharm: 16.874 Lines: 404 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CRLF) UTF-8

```

<그림3> main.py 파일 일부

<그림4> 'doklip3' 파일 내부

```

1 import pygame
2 import os
3 import handle
4 from variable_sh import *
5 from image_path_sh import *
6 from button import *
7 from sentence import *
8 from initGame import *
9 from makelist import *
10 from screenshot import *
11
12

```

<그림5> 예시-initgame와 같은 함수

4. 결론 및 제언

현재 우리나라에서 사용하고 있는 수학용어는 대부분 한자어로 이루어져 있어 직관적인 의미 파악이 어렵고, 많은 내용을 담고 있는 기존 사전에서 필요한 내용만을 분별해서 습득하기 어렵다. 따라서 본 연구는 학생들이 쉽게 수학 용어의 본질적인 의미를 파악하여 올바르게 이해하도록 중학교 과정 수학용어 프로그램을 제작하였다. 어플리케이션의 기능 및 장점은 다음과 같다.

첫째, python을 이용하여 제작된 프로그램이기에 한 번의 설치만으로 인터넷 연결 없이도 지속해서 이용 가능하다.

둘째, 용어 정의와 해당 용어와 관련된 부가적인 내용(공식이 나오게 된 과정, 그리고 단어에 관련한 역사 등)을 각각 탭으로 분류하여 빠르게 핵심 정보를 습득할 수 있다. 또한 4개의 출판사(신사고, 동아출판, 천재, 비상교육)의 교과서와 대한수학회 자료를 기반으로 제작하여 학생들이 혼란 없이 신뢰할 수 있는 정보로 학습에 도움이 되도록 하였다.

셋째, 용어 검색과 북마크 기능으로 편리하게 프로그램을 사용할 수 있다.

본 프로그램은 인터넷 연결이 불가능한 상황에서도 실행할 수 있고, 학년별로 분류되어있기 때문에 교사 혹은 학생들이 수업, 자습 시 편리하게 활용할 수 있을 것으로 기대한다. 또한 프로그램이 학생들의 수학 학습에 직접 활용되어 본질적인 의미를 이해하는 데에 도움이 되고, 더 나아가 수학에 대한 흥미와 학업성취도 향상에 도움이 되길 기대한다. 추가로 사전 제작 시 중학교 1, 2학년은 2015 개정 교육과정을 기준으로 제작되었지만, 중학교 3학년의 경우 2020년부터 2015 개정 교육과정이 적용될 예정이기 때문에 2009 개정 교육과정을 기준으로 하였다. 따라서 이 과제를 기반으로 추후 중3 2015 개정 교과서가 발간된 후 그 내용을 수정 및 발전시킬 필요성이 있으며, 초등과 고등학교 과정도 제작되길 기대한다.

5. 참고문헌 및 조사기관

- 신사고 중학교 수학1, 2, 3 교과서
- 동아출판 중학교 수학1, 2, 3 교과서
- 천재 중학교 수학1, 2, 3 교과서
- 비상교육 중학교 수학1, 2, 3 교과서
- 대한수학회 수학백과
- 허민, 수학용어의 개선 방향에 대한 소고, 수학교육논문집 제27집 제4호, 2013