휴먼 컴퓨터 인터페이스

과제 보고서 #3

구현 완성도 개선

제출일: 2018-05-27 (일)

소 속: 수학과

학 번: 2015603006

이 름: 김예지

담당교수: 이강훈 교수님

• 개요

1. 구현 완성도

(과제 #2와 동일. 빨간 글씨만 추가적으로 구현한 것)

항목	예시
1. 수식 입력	
(1) 정수, 실수, 복소수의 표현과 그 기본 연산	
- 산술 연산	+, -, *, /, %, ^, sqrt
- 비교 연산	==, !=, >, <, >=, <=
- 논리 연산	and, or, not, nor, true, false
- 비트 연산	&, |, ~, <<, >>
(2) 벡터, 행렬의 표현과 그 기본 연산	
- 벡터 & 행렬의 표현	x = [1, 2, 3], m=[1,2;3,4]
- 벡터: 내적, 외적	dot(x, y), cross(x, y) // x, y는 벡터
- 행렬: 곱셈, 역행렬, 행렬식	m*n, inv(m), det(m) // m, n는 행렬
(3) 자주 사용되는 상수 및 함수 지원	
- 상수	pi, e, i(복소수)
- 함수: 삼각함수, 지수/로그, 확률, 통계	sin, cos, tan, exp, log, mean, var, std, n!, nPr, nCr, …
2. 결과 출력	
- 올바른 입력 ➔ 수식의 결과 값	2*3 = 6
- 잘못된 입력 ➔ 오류 메시지	Parenthesis) expected
3. 변수, 함수 정의 및 사용	
- 변수: 최소 3개	x, y, z, a, b, c 포함 12개
- 함수: 최소 2개	f, g, h, f2, g2, h2 포함 12개
4. 추가 기능	
- COPY & PASTE 버튼 개선	식이나 결과를 선택해서 복사 & 붙여넣기 가능
- Redo, Undo 버튼 개선	이전 결과로 갈 수 있고 다시 돌아올 수 있음.
- Delete 버튼	문자열에서 뒤에 문자 한 개를 지움.
- Shift, alpha 버튼	버튼 재사용성.
- 그래프 기능	그래프를 그려준다.
- 도움말 기능	사용방법과 예시를 보여준다.

2. 개발 환경 & 조건

에디터	VS code
입력	Only mouse
호환성	구글 크롬 웹 브라우저

오픈소스	math.js, jquery.min.js, plotly.js
종횡비	1.78 : 1

3. 과제 #2에서 추가된 핵심적인 상호작용 방식 요약

1) **Copy / Paste 기능:** 과제 #2에서 구현했던 기능에서 추가적으로 복사하고자 하는 영역을 선택해서 복사할 수 있도록 추가적으로 구현함.

2) **Undo / Redo:** 과제 #2에서는 바로 전으로 돌아갈 수 있고 바로 뒤로 돌아올 수 있었는데 그 부분을 완벽히 구현. 처음으로 돌아갈 수 있으며, 실제 undo & redo 기능과 비슷하게 돌아간 곳에서 식을 계산하면 다시 그 곳부터 기록한다.

   예를 들어, 계산을 6번 수행해서 1~6과정까지 기록이 있다고 할 때, 3과정으로 돌아간 뒤 계산을 한다면 그 계산은 4번 과정이 되고 기존의 4~6은 삭제되는 방식이다.

   ※ 1)과 2)를 잘 사용하면 빠르고 간편하게 계산기를 사용할 수 있다.

3) **도움말:** 좌측 상단에 있는 '?'버튼을 클릭하면 사용법을 볼 수 있다. 폴더를 이용해 테마별로 분류를 해 두었다. 또한 부족한 설명을 대체하기 위해 시연 영상 링크를 담은 demon 버튼을 추가되었다.

4) **그래프:** 메인 화면에서 함수를 정의하면 우측 상단에 있는 'graph'버튼으로 들어갔을 때 정의 된 함수만 활성화 되어 있어 그래프를 그리기 원하는 함수들을 클릭하면 선택한 함수들이 무엇인지 볼 수 있는 창이 나오고, Draw 버튼을 클릭하면 선택한 함수들을 하나의 그래프에 그려준다. 버튼으로 정의역을 조절할 수 있고, 조절한 뒤 Draw 버튼을 클릭하면 사용자가 설정한 구간에서만 그래프가 그려진다.

● 본문

1. 기존 계획으로부터 변경된 부분과 그 이유

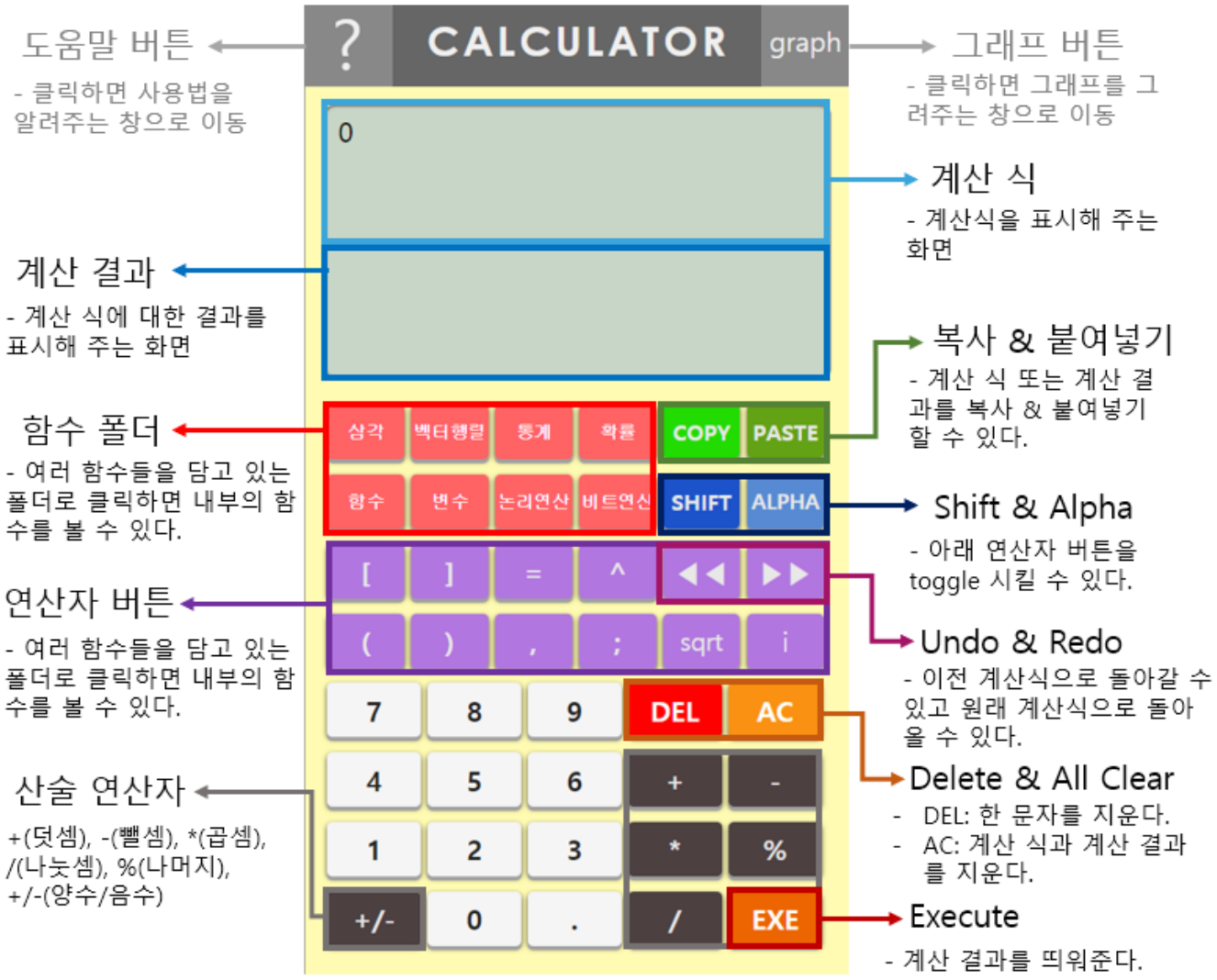
1) 메뉴 버튼

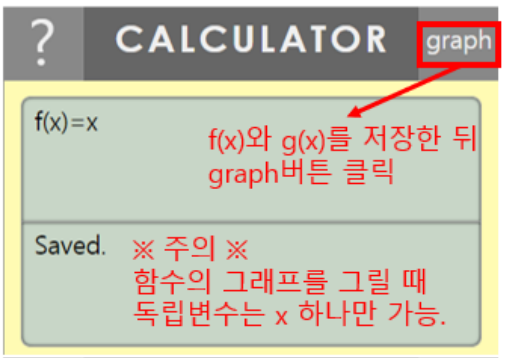
테마, 글씨체, 글자 색상, 검색 기능 등 여러 추가 기능을 메뉴 버튼에 모아 두고 싶었으나 테마나 글씨체, 글자 색상은 구현하는 것에 비해 사용자의 만족도가 그렇게 높아지지 않을 것 같았다. 그렇기 때문에 기능적인 면을 더 편리하게 바꾸는 데에 시간을 투자했다. 또한 검색 기능이나 즐겨 찾기 기능은 원래 많은 함수를 지원하고 그 중 골라서 사용자만의 계산기를 만들게 하는 것이 목적이었지만, 실제로 폴더로 묶어서 관리하다 보니 검색 기능이나 즐겨 찾기 기능의 이점이 크게 없음을 깨닫고 함께 없애게 되었다.

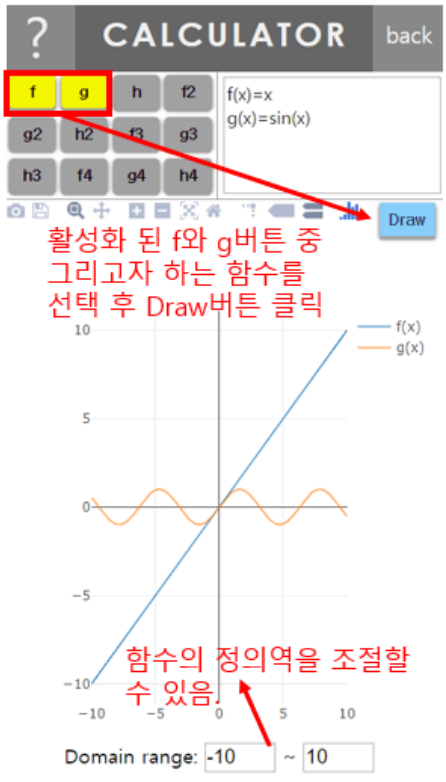
2) 도움말 버튼

각각의 기능마다 도움말 버튼을 달아 놓고 사용법과 예제를 보여주고 싶었으나, 그러기 위한 적절한 공간도 없었고, 핸드폰 터치로만 사용이 가능해야 하기 때문에 도움말을 보여주기 위한 쉽고 적절한 이벤트를 찾기 어려웠으며, 그렇게 많은 기능이 있지 않기 때문에 각각 보여주는 것 대신 폴더로 묶어 사용법과 예제를 한 번에 보여주고, 원하는 기능을 찾기 쉽도록 만들었다.

2. 구성 요소 및 사용방법


도움말 버튼
- 클릭하면 사용법을 알려주는 창으로 이동
CALCULATOR
graph
그래프 버튼
- 클릭하면 그래프를 그려주는 창으로 이동
0
계산 식
- 계산식을 표시해 주는 화면
계산 결과
- 계산 식에 대한 결과를 표시해 주는 화면
복사 & 붙여넣기
- 계산 식 또는 계산 결과를 복사 & 붙여넣기 할 수 있다.
함수 폴더
- 여러 함수들을 담고 있는 폴더로 클릭하면 내부의 함수를 볼 수 있다.
삼각
벡터행렬
통계
확률
함수
변수
논리연산
비트연산
COPY
PASTE
SHIFT
ALPHA
Shift & Alpha
- 아래 연산자 버튼을 toggle 시킬 수 있다.
연산자 버튼
- 여러 함수들을 담고 있는 폴더로 클릭하면 내부의 함수를 볼 수 있다.
[ ] = ^
( ) , ; sqrt i
Undo & Redo
- 이전 계산식으로 돌아갈 수 있고 원래 계산식으로 돌아올 수 있다.
7 8 9 DEL AC
4 5 6 + -
1 2 3 * %
+/- 0 . / EXE
Delete & All Clear
- DEL: 한 문자를 지운다.
- AC: 계산 식과 계산 결과를 지운다.
산술 연산자
+(덧셈), -(뺄셈), *(곱셈), /(나눗셈), %(나머지), +/-(양수/음수)
Execute
- 계산 결과를 띄워준다.


CALCULATOR
graph
f(x)=x
f(x)와 g(x)를 저장한 뒤 graph버튼 클릭
Saved.
※ 주의 ※
함수의 그래프를 그릴 때 독립변수는 x 하나만 가능.
CALCULATOR
graph
g(x)=sin(x)
Saved.


CALCULATOR
back
f
g
h
f2
g2
h2
f3
g3
h3
f4
g4
h4
f(x)=x
g(x)=sin(x)
Draw
활성화 된 f와 g버튼 중 그리고자 하는 함수를 선택 후 Draw버튼 클릭
f(x)
g(x)
함수의 정의역을 조절할 수 있음
Domain range: -10 ~ 10

삼각 버튼 클릭

- 스크롤 바를 내리면 더 많은 함수가 나옴.

EX) sin(30), sin(pi/2), ...

벡터행렬 버튼 클릭

- 스크롤 바 없음.

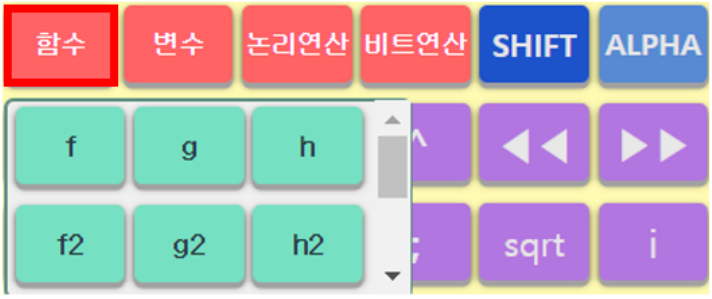
EX) **(벡터)**

- dot([1, 2], [-5, 9]) // 내적
  x=[1,2,3], y=[-1,0,4] 일 때,
- cross(x, y) // 외적 (3차원 벡터만 가능)
- norm(x), norm([3,7,10]) // 크기

(행렬)

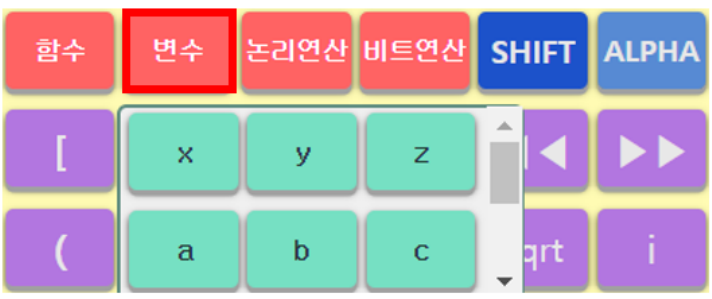
m=[1,2;3,4](=[[1,2],[3,4]])일 때,

- flatten(m) // 고차원 행렬을 1차원 벡터로 변환
- det(m) // determinant: 행렬식 (정사각행렬만 가능)
- trans(m) // transpose: 행렬의 행과 열을 바꿈
- inv(m) // inverse: 역행렬 (정사각행렬만 가능)



함수 버튼 클릭

EX) f(x)=x, g(a)=2*a, f2(x,y)=x+y, ...
f(3)=3, g(4)=8, f2(-4,3)=-1, ...



변수 버튼 클릭

EX) x=3, y=2*5, z=[1,2,3], m=[1,2;3,5], ...

통계 버튼 클릭

EX) x=[3,5,2,9,10] 일 때,

- mean(x) // 평균
- var(x) // 분산
- std(x) // 표준편차
- max(x) // 최댓값
- min(x) // 최솟값
- sum(x, y) // 합
- median(x) // 중간값
- mode(x) // 최빈값
- flatten(m) // 곱

확률 버튼 클릭

EX)

- permutations(n,r) // nPr: 순열
- combinations(n,r) // nCr: 조합
- 3!, 10! // factorial
- gcd(3,5) // 최대공약수
- lcm(3,5) // 최소공배수

논리연산 버튼 클릭

EX)

- and(true,false) // AND(&&)연산
- or(true,false) // OR(||)연산
- not(true) // NOT(!)연산
- nor(true,false) // NOR(^)연산
- true, false // 0: false, 0이 아닌 수: true

Default 상태

EX) 5^2, sqrt(3), 3+4i, ...

SHIFT 버튼 클릭

EX) e, pi, 3<5 (=true), 5<=3 (=false), ...

ALPHA 버튼 클릭

EX) exp(5)=e^5, ceil(5.2)=6, floor(5.2)=5,
abs(-9)=9, log(e)=1, log(100,10)=2,
round(3.2284)=3, round(3.2284, 3)=3.228, ...

(log는 두 번째 인자가 밑, 기본 값은 e,
round는 두 번째 인자가 소수점 및 자리 수,
기본 값은 0)

비트연산 버튼 클릭

EX)

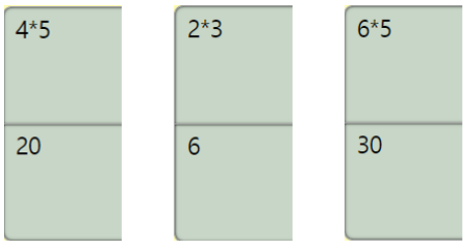
- 2&5 // 비트 and 연산
- 2|5 // 비트 or 연산
- ~7 // 비트 not 연산
- 3<<5 // 비트 shift 연산 (*2)
- 7>>2 // 비트 shift 연산 (/2)

- ◀◀를 누르면 이전 계산 식과 결과 화면으로 돌아 갈 수 있음.
- ▶▶를 누르면 다시 원래 식과 결과 화면으로 돌아올 수 있음.

- 실제 undo, redo 기능처럼 ◀◀로 이전 결과로 돌아간 상태에서 새로 식을 입력하고 계산을 하면 redo가 불가능해 짐.

- EXE를 눌러서 계산이 된 결과를 기준으로 함. 즉, EXE를 누르지 않은 경우는 undo, redo에 결과로 포함되지 않음.

EX) ① 1*2, 2*3, 3*4, 4*5를 순서대로 계산
② ◀◀를 두 번 눌러 2*3으로 돌아 감
③ 그 상태에서 *5를 눌러서 계산
④ ▶▶를 눌러도 아무 일이 일어나지 않음.



- COPY는 계산 식과 계산 결과 2가지의 문자를 그대로 복사할 수 있다.
- COPY를 한 상태에서 PASTE를 누르면 복사한 문자가 계산 식 화면에 나온다.
- 방법: 복사하고 싶은 영역을 클릭한 후 COPY 버튼을 클릭.

1) 계산 식을 복사할 경우

2) 계산 결과를 복사할 경우

3. 새로 추가한 특징적인 상호작용 방식에 대한 세부 구현 방법

[Copy / Paste 기능]

```
$('#input').click(function(){
    is_input = 0;
})
$('#result').click(function(){
    is_input = 1;
})
```

```
else if (textVal == 'COPY') {
    if (is_input == 1)
        copyString = preResult;
    else if (is_input == 0)
        copyString = preValue;

    copied = true;
    $(this).css('background-color', '#c6ff3b');
    $(this).css('color', 'gray');
    is_input = -1;
}
```

// 영역을 선택한 뒤 그에 맞는 값을 copyString에 추가한다.

[Undo / Redo 기능]

```
var textVal = $(this).text();
if (textVal == 'EXE') {
    try {
        while (preValues.length > 0 &&
            preValues[preValues.length - 1] != preValue) {
            preValues.pop();
            preResults.pop();
        }

        preValue = displayValue;
        preValues.push(preValue);
        var Functions = displayValue;
        displayValue = parser.eval(displayValue).toString();
        var tokens = displayValue.split(' ');
        if (tokens[0] == 'function'){
            var splits = Functions.split("=");
            dict[splits[0]] = splits[1];

            displayValue = "Saved.";
            var elements = document.getElementsByClassName('func_key');
            for(var i = 0; i < 12; i++)
                if($(elements[i]).text() == splits[0].split("(")[0]){
                    $(elements[i]).addClass('func_key_available');
                    break;
                }
        }

        preResult = displayValue;
        preResults.push(preResult);
        $('#result').val(displayValue);
        displayValue = '0'; resultValue = '';

        afterEXE = true;
    }
    catch (e) {
        preValue = displayValue;
        displayValue = '0';
        if (displayValue != 'function')
            $('#result').val(e);

        preResult = '';
    }
}
```

먼저, EXE를 하면 계산 식을 모아두는 list인 preValue를 이용해 현재 위치 이후의 계산 식을 다 지운 뒤 현재 계산 식과 결과를 추가한다.

그래프를 그릴 때 필요한 함수를 dictionary로 관리하기 위해 함수를 '=' 을 기준으로 나누고 앞 부분은 key, 뒤 부분은 value로 넣는다.

그리고 정의 된 함수는 활성화 해주기 위해 func_key_available이라는 클래스를 추가한다.

```
else if (textVal == '◀◀') {
    displayValue = '';
    if (preValues.indexOf(preValue) > 0) {
        preValue = preValues[preValues.indexOf(preValue) - 1];
        preResult = preResults[preResults.indexOf(preResult) - 1];
        $('#input').val(preValue);
        $('#result').val(preResult);
    }
}
else if (textVal == '▶▶') {
    displayValue = '';
    if (preValues.indexOf(preValue) < preValues.length - 1) {
        preValue = preValues[preValues.indexOf(preValue) + 1];
        preResult = preResults[preResults.indexOf(preResult) + 1];
        $('#input').val(preValue);
        $('#result').val(preResult);
    }
}
```

Undo: 현재 index에서 하나 전의 계산식과 결과를 보여준다.

Redo: 현재 index에서 하나 뒤의 계산식과 결과를 보여준다.

[도움말 기능]

```
<div id="container">
    <div id="top">
        <span id="question" class="extra_btn">?</span>
        <span id="title">Calculator</span>
        <span id="graph" class="extra_btn">graph</span>
    </div>
    <div id="help">
        <span class="help_btn">calculator</span>
        <span class="help_btn">graph</span>
        <span class="help_btn">folder</span>
        <span class="help_btn">undo/redo</span>
        <span class="help_btn">copy/paste</span>
        <span class="help_btn">shift/alpha</span>
        <span class="help_btn">
            <a href="https://youtu.be/3j2YKiefaVM">demon</a></span>
        <div id="calc_help" class="hidebtn">
            <img src="./Img/right.png" alt="help-image" style="width: 100%">
            <img src="./Img/left.png" alt="help-image" style="width: 100%">
        </div>
        <div id="graph_help" class="hidebtn">
                <img src="./Img/graph1.png" alt="help-image" style="width: 100%">
                <img src="./Img/graph2.png" alt="help-image" style="width: 100%">
        </div>
        <div id="folder_help" class="hidebtn">
            <img src="./Img/folder1.png" alt="help-image" style="width: 100%">
            <img src="./Img/folder2.png" alt="help-image" style="width: 100%">
            <img src="./Img/folder3.png" alt="help-image" style="width: 100%">
            <img src="./Img/folder4.png" alt="help-image" style="width: 100%">
        </div>
        <div id="undo_help" class="hidebtn">
            <img src="./Img/undo.png" alt="help-image" style="width: 100%">
        </div>
        <div id="copy_help" class="hidebtn">
                <img src="./Img/copy1.png" alt="help-image" style="width: 100%">
                <img src="./Img/copy2.png" alt="help-image" style="width: 100%">
        </div>
        <div id="shift_help" class="hidebtn">
            <img src="./Img/shift_alpha.png" alt="help-image" style="width: 100%">
        </div>
    </div>
```

// HTML 코드 – span으로 버튼을 관리하고 div로 나오는 이미지를 관리한다.

```
$('#question').click(function(e){
    if(flag == 0){
        $('#calculator').hide();
        $('#help').show();
        flag = 1;
    }
    else{
        $('#calculator').show();
        $('#help').hide();
        flag = 0;
    }

    $('#graph_area').hide();
    $('#graph').text('graph');
    $('#graph').css('padding-left','5px');
    $('#graph').css('padding-right','5px');
})
```

// 도움말 & 그래프 화면을 적절히 보여주기 위한 코드

```
var manager = [0,0,0,0,0,0];
$('.help_btn').click(function(){
    $('.hidebtn').hide();
    var text_name = $(this).text();
    if(text_name == 'calculator'){
        if(manager[0] == 0){
            $('#calc_help').show();
            manager[0] = 1;
        }
        else{
            $('#calc_help').hide();
            manager[0] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 0) continue;
            manager[i] = 0;
        }
    }
    else if(text_name == 'graph'){
        if(manager[1] == 0){
            $('#graph_help').show();
            manager[1] = 1;
        }
        else{
            $('#graph_help').hide();
            manager[1] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 1) continue;
            manager[i] = 0;
        }
    }
    else if(text_name == 'folder'){
        if(manager[2] == 0){
            $('#folder_help').show();
            manager[2] = 1;
        }
        else{
            $('#folder_help').hide();
            manager[2] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 2) continue;
            manager[i] = 0;
        }
    }
    else if(text_name == 'undo/redo'){
        if(manager[3] == 0){
            $('#undo_help').show();
            manager[3] = 1;
        }
        else{
            $('#undo_help').hide();
            manager[3] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 3) continue;
            manager[i] = 0;
        }
    }
    else if(text_name == 'copy/paste'){
        if(manager[4] == 0){
            $('#copy_help').show();
            manager[4] = 1;
        }
        else{
            $('#copy_help').hide();
            manager[4] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 4) continue;
            manager[i] = 0;
        }
    }
    else if(text_name == 'shift/alpha'){
        if(manager[5] == 0){
            $('#shift_help').show();
            manager[5] = 1;
        }
        else{
            $('#shift_help').hide();
            manager[5] = 0;
        }
        for(var i = 0; i < 6; i++){
            if(i == 5) continue;
            manager[i] = 0;
        }
    }
})
```

>> 도움말에 버튼에 해당하는 이미지를 띄우기 위해 배열로 관리한다.

[그래프]

```html
<div id="graph_area">
    <div id="btn_container">
        <div id="func_button">
            <span class="func_key">f</span>
            <span class="func_key">g</span>
            <span class="func_key">h</span>
            <span class="func_key">f2</span>
            <span class="func_key">g2</span>
            <span class="func_key">h2</span>
            <span class="func_key">f3</span>
            <span class="func_key">g3</span>
            <span class="func_key">h3</span>
            <span class="func_key">f4</span>
            <span class="func_key">g4</span>
            <span class="func_key">h4</span>
        </div>
        <div id="show_list">
            <textarea id="func_list" class='display' type="text" autocomplete="off"
            spellcheck="false" readonly="true"></textarea>
        </div>
    </div>
    <span id="draw">Draw</span>
    <div id="plot"></div>
    <div id="domain">
        Domain range:
        <input id="min" class="range" type="number" value="-10"/>
        ~
        <input id="max" class="range" type="number" value="10"/>
    </div>
</div>
```

```javascript
$('#graph').click(function(e){
    if($(this).text() == 'graph'){
        $('#calculator').hide();
        $('#graph_area').show();
        $(this).text('back');
        $(this).css('padding-left','10px');
        $(this).css('padding-right','8px');

        $('#plot').empty();
        draw();
    }
    else{
        $('#calculator').show();
        $('#graph_area').hide();
        $(this).text('graph');
        $(this).css('padding-left','5px');
        $(this).css('padding-right','5px');
    }

    $('#help').hide();
    flag = 0;
})
```

```javascript
$('.func_key').each(function(index, key){
    $(this).click(function (){
        if($(this).hasClass('func_key_available')){
            if($(this).hasClass('func_key_clicked'))
                $(this).removeClass('func_key_clicked');
            else
                $(this).addClass('func_key_clicked');
        }

        var elements = document.getElementsByClassName('func_key');
        $('#func_list').val("");
        for(var i = 0; i < 12; i++){
            var element = elements.item(i);
            if($(element).hasClass('func_key_clicked')){
                var func_name = $(element).text() + "(x)";
                if(func_name in dict){
                    var Text =  $('#func_list').val() + func_name + "=" +
                    dict[func_name] + "\n";
                    $('#func_list').val(Text);
                }
            }
        }
    })
})
```

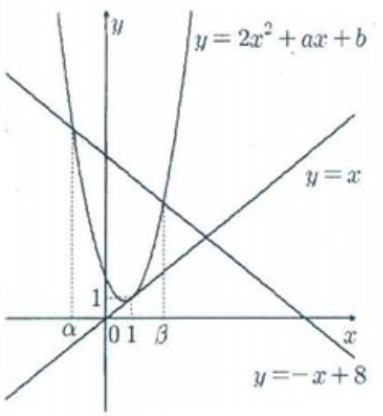
// 도움말 & 그래프 화면을 적절히 보여주기 위한 코드

// 활성화 된 함수의 버튼을 누른 경우 그 함수의 내용을 textarea에 출력하는 코드

```
function draw(){
    try{
        var min = $('#min').val();
        var max = $('#max').val();
        var xValues = math.range(min, max, 0.1).toArray();
        var data = [];
        var layout = {
            width: 398.5, height: 535
        };

        var elements = document.getElementsByClassName('func_key');
        for(var i = 0; i < 12; i++){
            var element = elements.item(i);
            if($(element).hasClass('func_key_clicked')){
                var func_name = $(element).text() + "(x)";
                if(func_name in dict){
                    var expr = math.compile(dict[func_name]);
                    var yValues = xValues.map(function(x){ return expr.eval({x: x});

                    var trace = {
                        x: xValues,
                        y: yValues,
                        mode: 'lines',
                        name: func_name,
                        type: 'scatter',
                        line: {width: 1}
                    };

                    data.push(trace);
                }
            }
        }
        Plotly.newPlot('plot', data, layout);
    }
    catch(err){
        console.error(err);
        alert(err);
        $('#domain').hide();
    }
}
```

// plotly.js에 있는 newPlot이라는 함수를 적절히 사용하여 그래프를 그려준다.

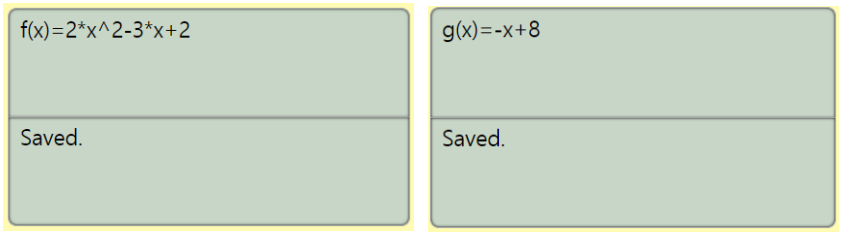
4. 추가 구현에 대한 실제 문제 사용 예시



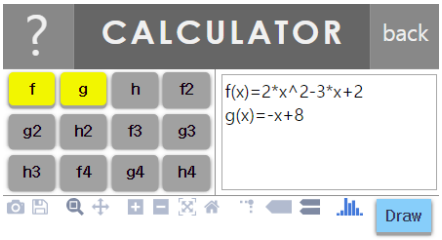
문제) 이차함수 $y=2x^2+ax+b$의 그래프는 직선 $y=x$와 점(1,1)에서 접하고 직선 $y=-x+8$과 두 점에서 만난다. 이차함수 $y=2x^2+ax+b$의 그래프가 직선 $y=-x+8$와 만나는 두 교점의 x좌표를 그림과 같이 α, β라 할 때, $(\beta-\alpha)^2$의 값은?[11]

풀이) $y'(1)=1$, $y(1)=1$을 이용해 a와 b를 구하면, $a=-3$, $b=2$가 된다.

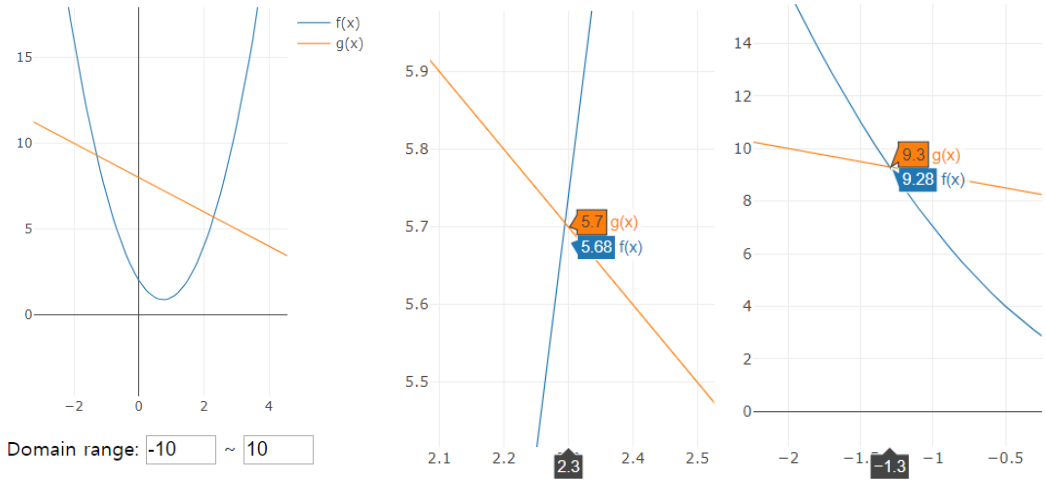
$f(x)=2x^2-3x+2$와 $g(x)=-x+8$을 그래프로 그려서 α와 β를 찾아보자.

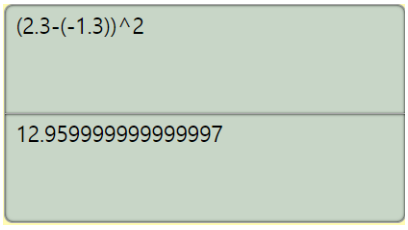


-> 함수 정의



>> 그래프 그리고 근사 해 찾기





즉, 근사적으로 $\alpha \fallingdotseq 2.3$, $\beta \fallingdotseq -1.3$이 나온다. $(\beta-\alpha)^2 = 12.959999999999 \fallingdotseq 13$ 이므로 답은 13이 나온다.

● 논의

1. 구현 측면에서 성공적인 부분과 실패한 부분

1) 성공적인 부분

- 먼저, 과제 #2에서 미처 완성하지 못했던 도움말 기능과 복사 & 붙여넣기 기능, Undo & Redo 기능을 완성했고, 그로 인해 사용자가 더 편하고 빠르게 계산기를 사용할 수 있다고 생각한다.
- 이번 과제의 핵심인 그래프 그리기도 성공적으로 잘 마쳤다고 생각한다. 그래프를 그리는 부분에서 오픈 라이브러리를 잘 사용해서 필요에 맞게 고쳐서 원하는 대로 구현이 되었고, 정의역을 설정하는 부분에서 사용자가 임의로 정할 수 있도록 숫자 자판을 따로 만들어야 하나 고민했지만 그러다가 input 태그의 number라는 타입을 이용해서 키보드 없이도 마우스로 조작할 수 있다는 것을 알아내서 쉽게 구현할 수 있었다.

2) 실패한 부분

- 할당 된 함수와 변수를 보여주고, 최근의 자주 쓰이는 변수 같은 경우에는 따로 만들어 주는 것이 좋을 것 같다고 생각했지만 시간이 오래 걸릴 것 같아 시도하지 못해서 아쉽다. 대신 거의 마지막에 수정해서 보고서나 영상에는 없지만, x라는 가장 많이 사용하는 변수를 밖에 하나의 버튼으로 위치시켜 접근성을 높이고 시간을 단축시켰다.
- 개인적으로 계산기에서 중요하다고 볼 수 있는 연립방정식 풀이 기능을 구현하지 못하였다. 그래프로 교점을 보고 소수점 첫째 자리까지 근사값을 찾을 수는 있지만 정확한 해는 알아낼 수 없기 때문에 정확한 해를 구해주는 기능이 없다는 것이 많이 아쉽다.

2. 사용성 측면에서 긍정적인 측면과 부정적인 측면

1) 긍정적인 측면

확실히 그래프가 추가되고 많은 기능이 생기면서 계산기를 이용해 할 수 있는 것들이 많아졌다. 또한 사용자의 편리성을 위해 만들고자 했던 copy & paste 기능과 undo & redo 기능이 있기 때문에 이 둘을 적절히 잘 사용한다면 계산기 사용 효율이 확실히 높아질 것이다. 처음 사용하는 사람을 위해 도움말 기능을 넣어 놨는데, 이미지를 통해 자기가 필요한 기능만 펼쳐볼 수 있도록 만들었기 때문에 이 점에서 사용자가 불편함을 느끼는 것이 줄어들 것이라 예상한다.

2) 부정적인 측면

현재 웹으로 만든 계산기이고, 마우스로만 조작할 수 있기 때문에 평소 계산기처럼 손으로 빠르게 누르는 것이 아니기 때문에 조작법이 불편할 수 있을 것 같다. 터치 스크린으로 조작한다면 훨씬 빠르고 쉽겠지만 마우스 클릭과 드래그, 스크롤을 통해서만 조작해야 하기 때문에 비교적 불편하고 느리게 느껴질 것 같다.

3. 과제에 대한 전반적인 자체 평가 및 향후 개선 계획

1) 처음 프로토타입을 구상할 때 구현하려고 했던 기능은 대부분 구현이 되었고, 추가적으로 그래프 기능까지 오류 없이 작동하기 때문에 전반적으로 굉장히 만족스럽다고 생각한다. 향후에 개선할 점은 연립 방정식 부분과 할당 된 함수나 변수의 값을 볼 수 있는 정도면 될 것 같다. 또한 개인적으로 계산기의 가장 큰 장점인 미분, 적분 기능까지 구현할 수 있었다면 더 좋았을 것 같다.

★ **유튜브 시연 동영상 URL:** https://youtu.be/3j2YKiefaVM