

# Calculator for you

휴먼 컴퓨터 인터페이스 [이강훈 교수님]

소프트웨어학부 3학년  
2017203039 박선영  
제출일 : 2019.06.23

## [요약]

오픈소스 라이브러리 의존성  
jQueryjs, Math.js

### 추가적으로 구현한 기능

1. Intellisense
2. 키보드 마우스 입력 지원
3. 도움말 기능
4. Degree & Radian 모드
5. 히스토리 창 지원
6. 복사 붙여넣기 가능
7. 커서 위치에 따른 입력 가능

수식 입력	구현 여부
정수 , 실수, 복소수 표현	O
정수, 실수, 복소수 기본연산	O
벡터 행렬의 표현	O
벡터, 행렬의 기본 연산	O
자주 사용되는 상수 지원	O
자주 사용되는 함수 지원	O
결과 출력 (올바른 입력)	O
결과 출력 (잘못된 입력)	O
변수, 함수 정의 및 사용	O

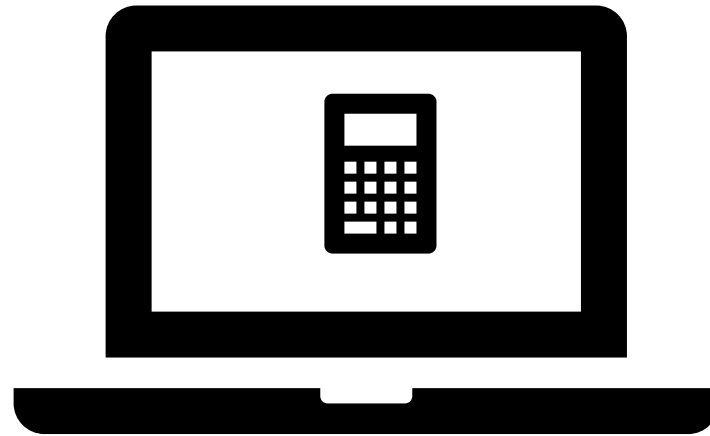
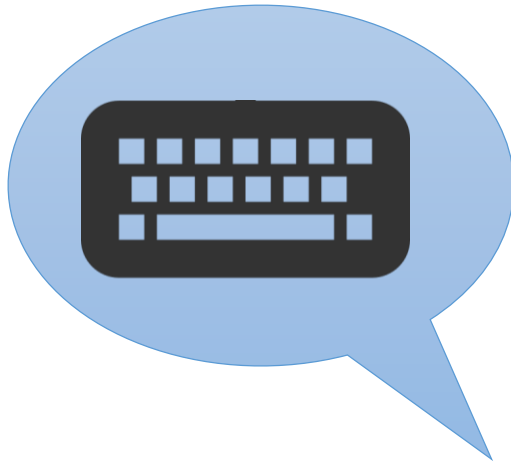
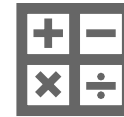


## CONTENTS

---

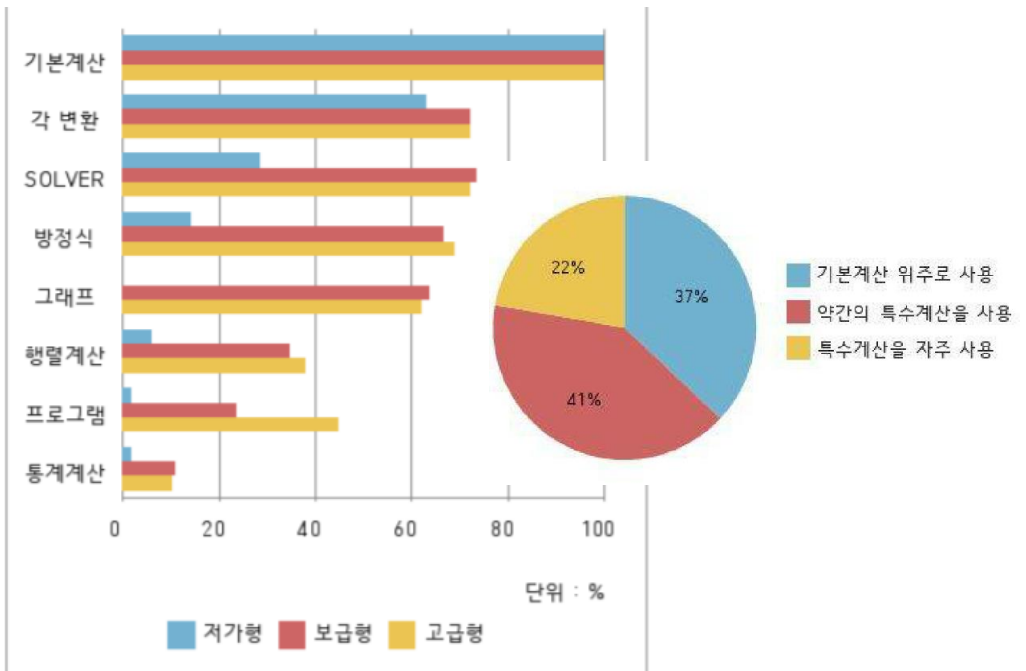
1. 프로젝트의 방향
2. 인터페이스 설계
3. 프로토타입 구현
4. 사용성 평가
5. 결론

## | 사용자 설정



키보드와 마우스 사용이 익숙한  
고급 수학을 배우는 대학(원)생

# | 과업 설정 -사용자 조사에 따른 과업 설정



[기본 계산]  
 $((237-261)*37/5+3813)-32+(36*23-326)+(7736/3+67)=?$

[각 변환]  
If  $x=1.33^\circ$ ,  $\tan x \neq \sin x =?$

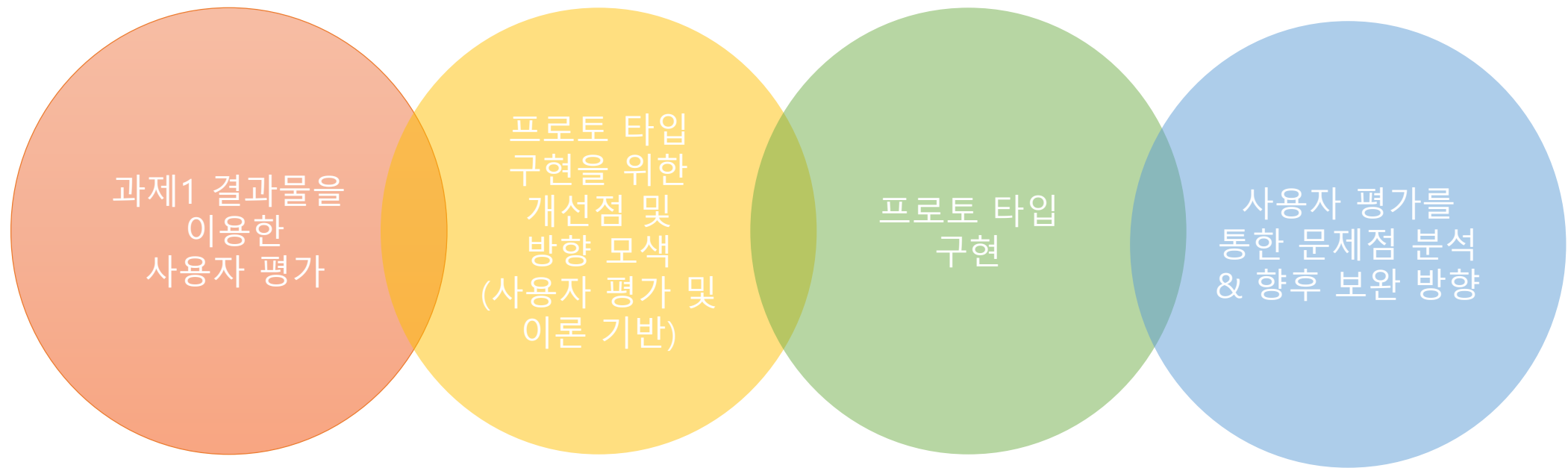
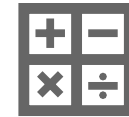
[방정식]  
 $F(x)=x^4+3*x^3+21*x^2+32$   
 $F(2)=?$

## 사용 목적 및 특수 계산 빈도

위의 조사 결과는 20-30대 자연계열, 공학계열, 학생, 연구원150 명을 대상으로 한, 설문조사 결과 중 일부를 발췌한 것이다. 위 의 설문 조사들을 통해 상위 4개 항목 중 3항목 기본 계산과 약 간의 특수 계산을 사용하여 과업을 설정했다.

출처:공학용 계산기 인터페이스 디자인의 사용성에 관한 연구, 김재현(국민대학교) p46

# | 프로젝트 진행절차



# 사용성 평가(prototype homework 1)

사용성 평가 이유

과제로 제출했던 homework1의 계산기를 사용하면서 개선할 점과 사용자 평가를 받아, 조금 더 좋은 ux를 줄 수 있는 계산기를 만들기 위해서 에스노테라피 기법을 이용한 사전 사용성 평가를 진행했다.

실험 환경

장소 : 홍대 카페 드x탑  
일시 : 2019년 6월 2일 저녁 7시경

사용자 선정

이공계 분야를 전공으로 하는 공학용 계산기를 주로 사용하는 대학생 중 1명을 사용자로 선정했다.

과업

[기본 계산]  
 $((237-261)*37/5+3813)-32+(36*23-326)+(7736/3+67)=?$

[각 변환]  
If  $x=1.33^\circ$ ,  $\tan x + \sin x = ?$

[방정식]  
 $F(x)=x^4+3*x^3+21*x^2+32$   
 $F(2)=?$

사용자 정보

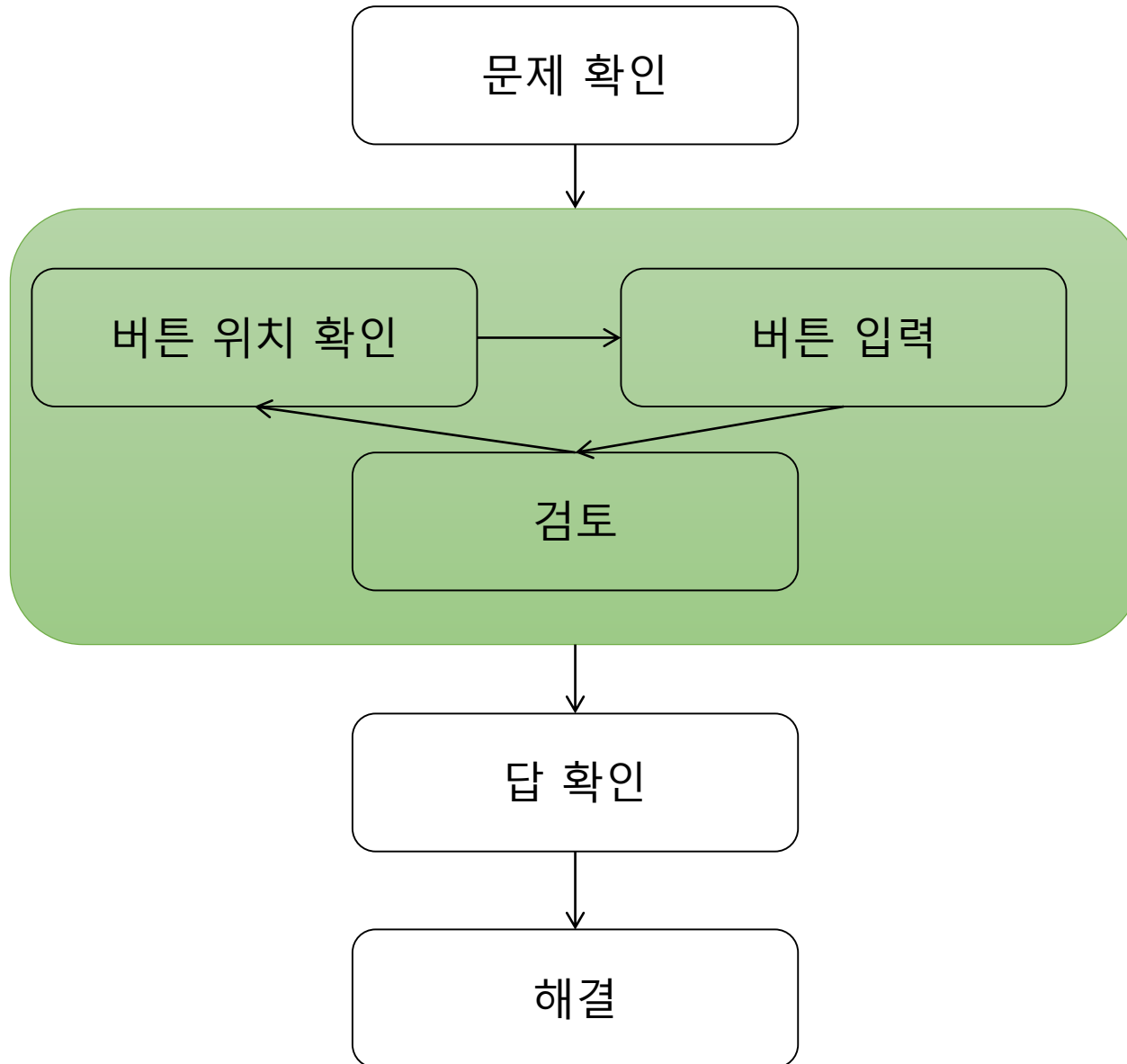
이름 : 김x배  
나이 : 23세  
직업 : 대학생  
전공: 컴퓨터 소프트웨어  
사용자 성격 : 신중하고 책임감이 강하다.  
특기 : 공학 수학 및 화학 문제 해결

사전 인터뷰

Q 자기소개 부탁드립니다.  
A 안녕하세요. 저는 광운대학교 컴퓨터 소프트웨어학과 2학년 재학중인 16학번 김x배입니다.  
Q 공학용 계산기를 사용해보신 적이 있으신가요?  
A 네 있습니다. 1학년 때 대학 화학을 들으면서 자주 사용했습니다.  
Q 얼마나 자주 사용하셨습니까?  
A 수업을 들을 때마다 사용했으니, 주 2회 정도 사용했습니다.  
Q 주로 어떤 계산을 할 때 공학용 계산기를 사용하셨나요?  
A 농도 계산, 몰 계산과 같은 약간 복잡한 단순계산을 할 때 자주 사용했습니다.  
Q 공학용 계산기의 이점은 무엇이라고 생각하세요?  
A 미지수도 사용할 수 있고, 함수도 사용할 수 있는 것이 좋고, 큰 수를 계산하기 쉬운 점입니다.  
Q 공학용 계산기의 단점은 무엇이라고 생각하세요?  
A 조작법이 어렵다는 점 같습니다.

## 분석

### 문제 해결방식





## 분석

### 발견된 문제점

1. 마우스 조작만으로 숫자와 수식을 입력해야 해서, 사용자가 지쳐 보였고, 오랜 시간이 수식과 숫자 입력에 사용되었다.
2. Sin과 tan 등의 함수를 사용할 때 사용자가 쉽게 찾지 못하고 헤매는 모습을 발견했다.
3. 잘못 입력한 경우, error로만 출력되게 했는데, 이때 어떠한 종류의 에러가 발생한 지를 몰라서, 다시 수식을 입력할 때 이전 식의 내용을 다시 유심히 보는 모습을 발견했다.
4. 벡터와 행렬 계산을 쉽게 하기 위해서 사용한  $\blacktriangleright$  이 기호와  $\leftarrow$  숫자나 수식을 하나씩 지울 수 있는 이 기호를 헷갈려 혼동해 사용했다.
5. 팝업 메뉴를 사용해서 클릭 후 드래그해 입력하는 방식을 사용하는 것에 실패가 잦았다.
6. 각도 계산의 경우 degree를 따로 입력하지 않고, 당연히 degree 모드로 되고 있다고 생각하는 경우가 있다.
7. 버튼을 입력 후, 재검토 하는 과정을 거친다.

## 분석

### 사후 인터뷰

Q 방금 사용한 계산기는 어땠나요?

A 숫자 쓰는게 불편해서, 화학보다 수학같은 계산에서는 편할 거 같다. 화학처럼 숫자를 많이 쓰는 계산은 힘들 것 같고, 기호가 잘 되어있어서, 사용하기 편했다.

Q 메뉴의 구성이나 개수는 어떤가요?

A 메뉴가 많아서 숫자를 잘못 누르거나, 숫자를 하나씩 누를 때마다 확인해야 해서 불편했다. 함수나 변수를 팝업메뉴로 구현이 되어있었는데, 따로 두는 것보다는 편한 것 같다.

Q 각도 계산에서 deg를 사용하는게 찾는데 어려움을 겪으신 것 같은데, 항상 degree radian을 따로 입력하는 건 어떻게 생각하시나요?

A deg를 항상 입력하지 않아도 모드로 바뀌서 하면 좋을 것 같다.

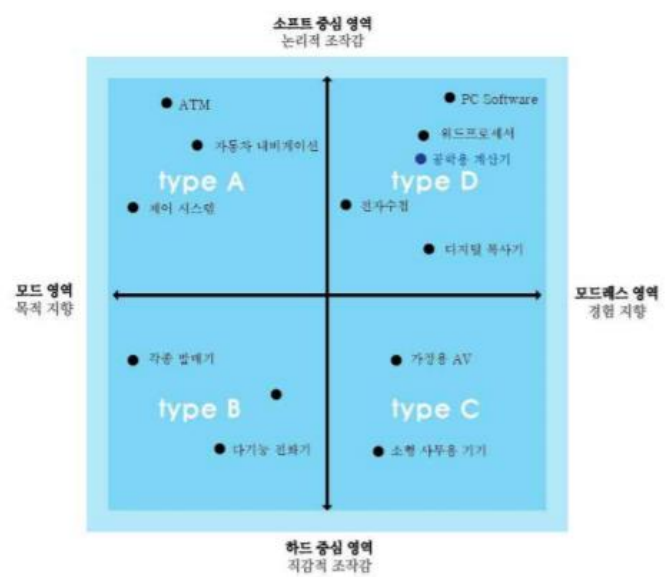
Q 긴 수식을 입력하실 때 되게 오래 걸리시고, 어려워 보였는데 왜 그러신거죠?

A 숫자가 많아서 마우스로만 하기 오래 걸리고, 버튼의 위치를 찾기 힘들어서 sin tan같은 건 아는데, 위치가 어디 있는지 모르니까 오래 걸렸다.

Q 무엇이 개선 되었으면 하시나요?

A 키보드 입력도 되었으면 좋겠고, 기호가 적당히 있었으면 좋겠다. 이전에 계산했던 기록들도 남았으면 좋겠다.

# 공학 계산기 분석



- 초심자
- 중급자
- 숙련자

## 공학 계산기의 UI 분석

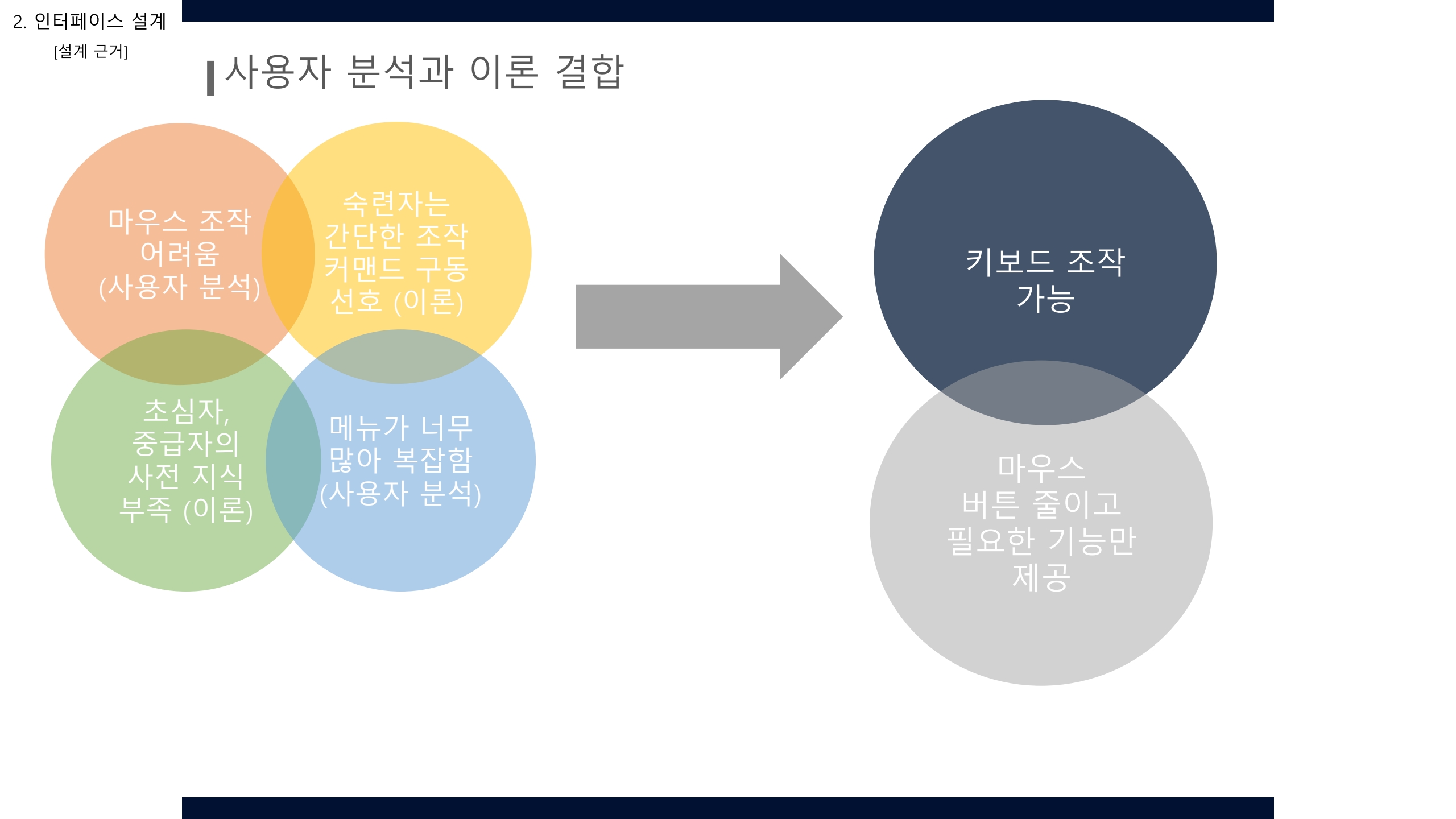
위의 그림은 UI를 지향하는 방향에 따라 분류한 것인데, 공학 계산기의 경우 type D에 속한다. Type D는 사용하기 쉬운 점과 목적에 따라 사용법을 달리할 수 있는 유연성이 장점이지만, 처음 사용할 경우 어느 정도 지식이 필요하여 사용이 어렵다는 문제가 있다.

## 숙련도에 따른 사용자의 특징

숙련도에 따라 사용자는 세 그룹으로 나뉘는데, 초심자의 경우 사용시 쉽게 흥미를 잃을 수 있고, 상당한 지식이 필요한 조작에 대해서 매뉴얼대로 따라 한다면 조작 가능하다고 생각하며, 중급자의 경우 불완전한 지식으로 실수를 반복하여 에러 복구와 오 조작에 대한 저항성을 중요시 한다. 숙련자의 경우 복잡한 조작을 싫어하여 간단한 조작방법을 제 공하고 메뉴보다 단축키를 이용한 커맨드 구동을 선호한다.

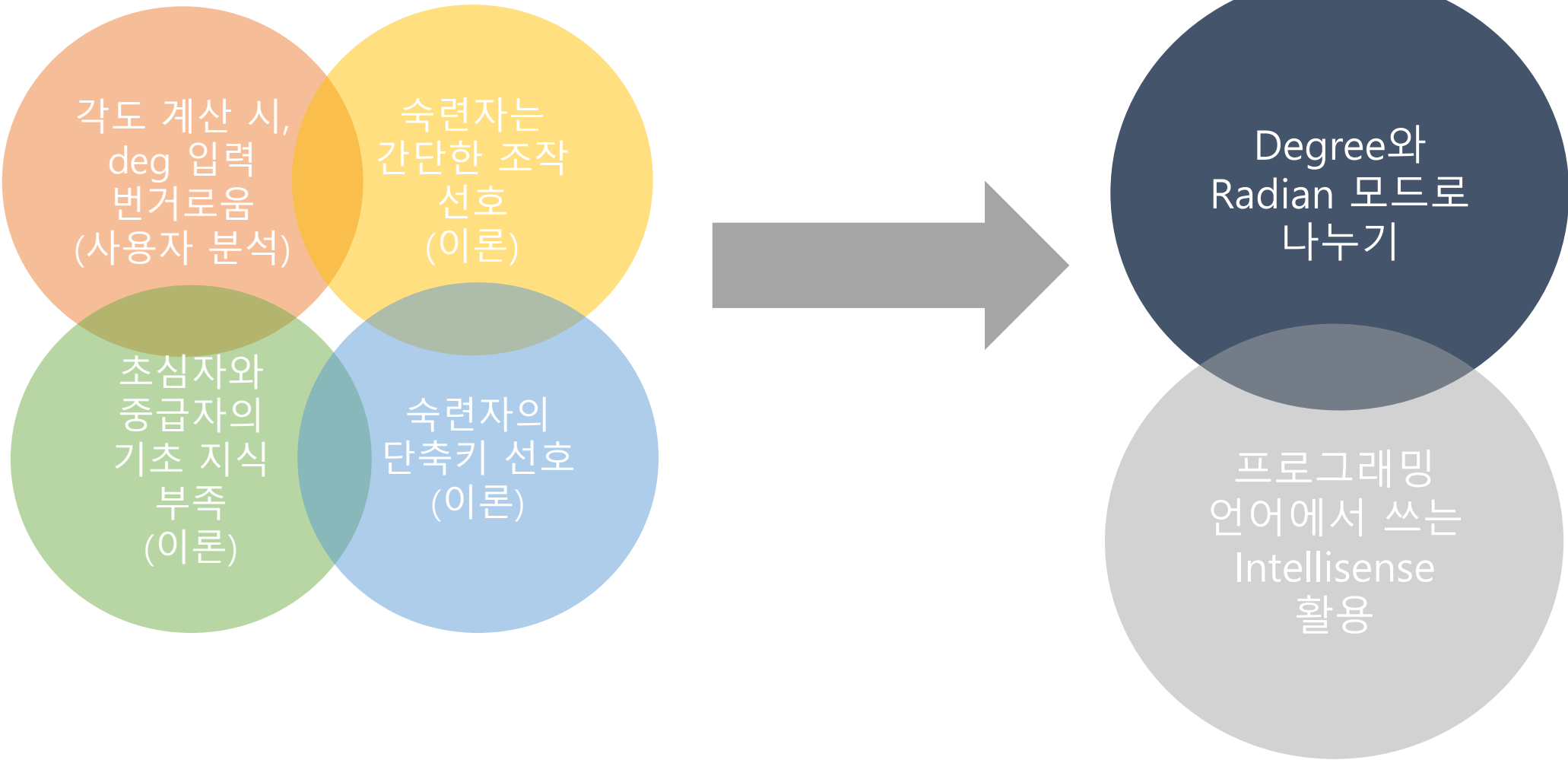
출처:이진호,GUI 디자인 가이드, 언그래픽스,2002,p47

출처:공학용 계산기 인터페이스 디자인의 사용성에 관한 연구, 김재현(국민대학교),p33

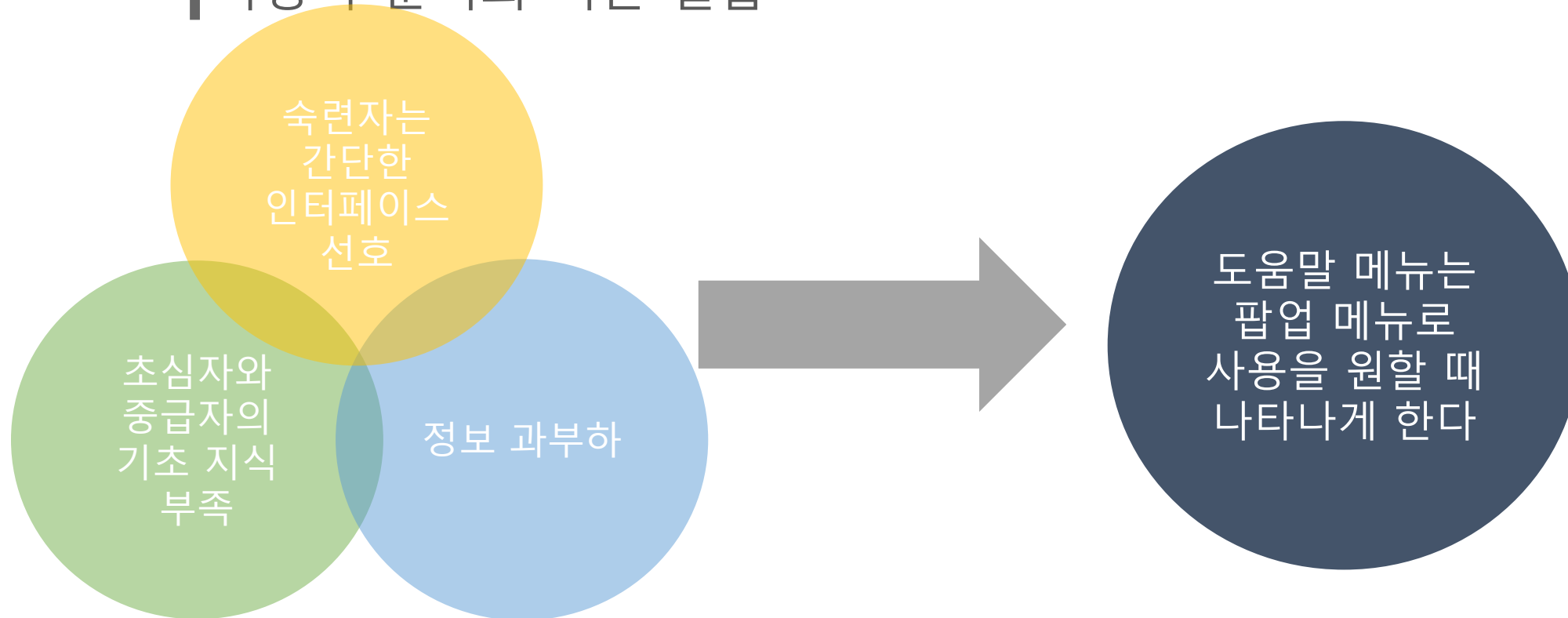


[설계 근거]

# 사용자 분석과 이론 결합



## ■ 사용자 분석과 이론 결합



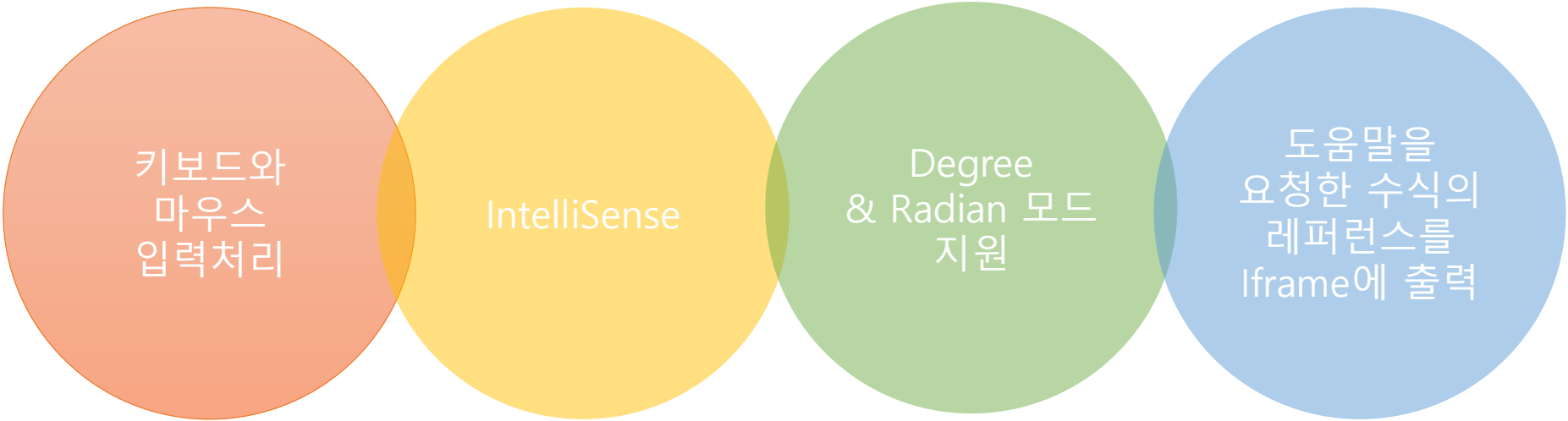
정보 과부하란? 1. 정보 과잉 현상

2. 인간의 정보처리 능력을 넘어설 정도로 정보의 양이 많은 상태

3. 잘못된 결정을 하거나 정보 자체를 무시하거나 정보 처리를 포기하는 부작용

사용자가 감당할 수 없는 많은 정보를 주는 것은 오히려 사용자에게 부작용을 일으키게 되는데, 이러한 부작용을 막고 숙련자와 초심자 중급자 모든 사용자를 고려할 수 있는 방안이 무엇일까? 생각하다가 도움말 메뉴를 만들되, 사용자가 도움말을 사용한다고 요청했을 경우에만, 도움말을 띄워 정보량을 조정하기로 하였다.

# | 기능적 기대효과



키보드와 마우스 모두 사용자  
입력을 지원하여, 원하는  
인터페이스를 사용할 수 있도록  
제공함으로써 UX를 높인다.

입력한 문자로 시작하는 수식의  
리스트들을 사용자에게 보여주  
어, 이용시간을 단축할 수 있다.  
또한 공학계산기의 UI가 D type에  
속해, 지식이 없는 사용자의 경우  
사용이 어려우나 IntelliSense가 그  
러한 어려움을 덜어준다. 방향키  
와 엔터키 마우스 조작으로 추가  
가 가능하게 하여 사용자에게 간  
단한 조작방법을 제시한다.

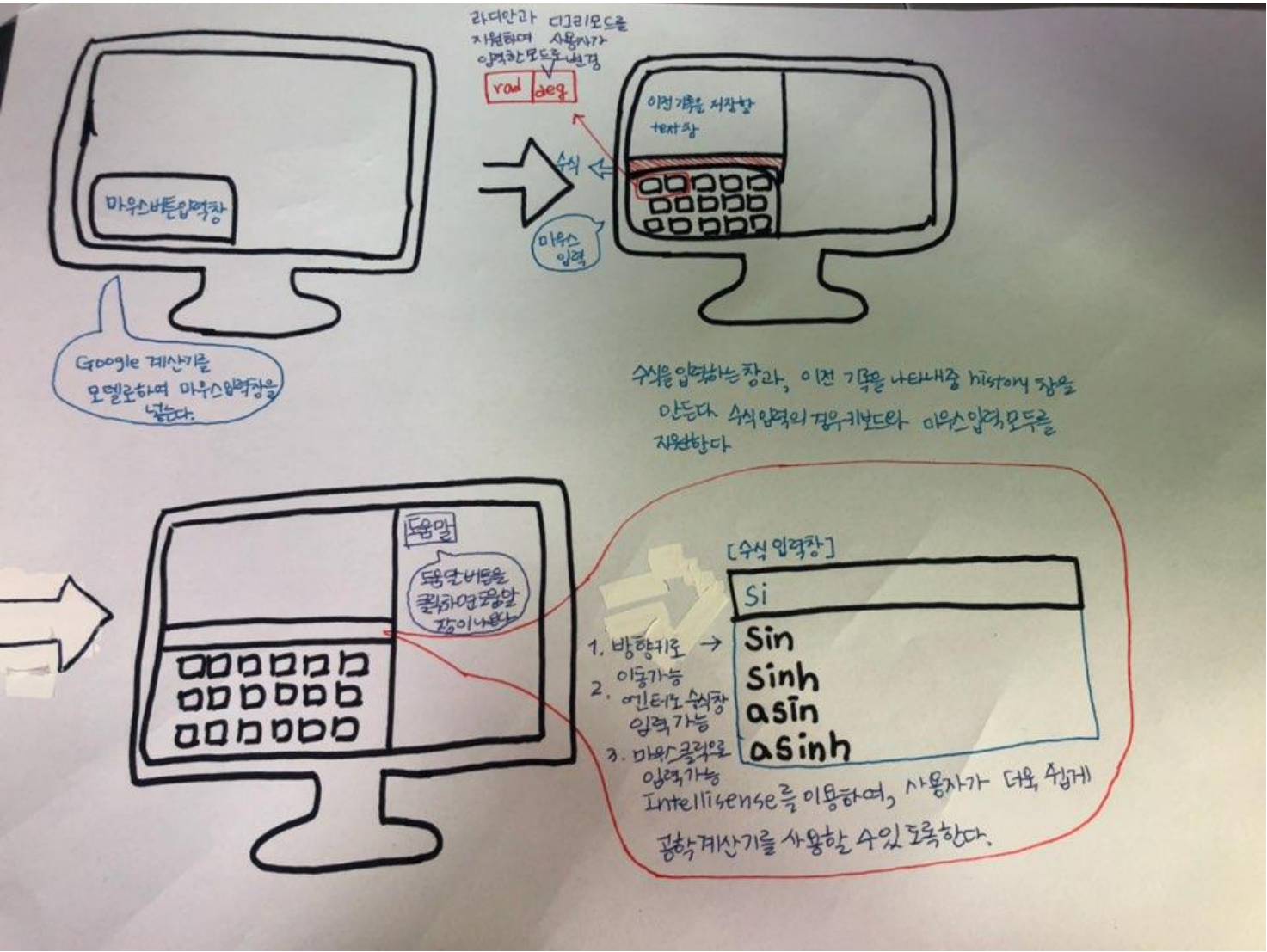
매번, 삼각함수 등에서 deg를 붙  
이거나, 별도로 계산을 하는 것이  
사용자에게는 꽤나 번거로운 과  
정이다. 이러한 번거로움을 없애  
기 위해 degree와 radian 모드를 지  
원하여, 모드를 바꾸기만 하면 매  
번 입력하지 않아도 되도록 하여,  
편의성을 높였다.

공학 계산기에 무지한 초심자나, 아  
직 불완전한 지식을 가진 중급자를  
위해 iframe에 요청한 수식의 도움말  
을 출력해주어, 사용자가 쉽게 공학  
계산기를 사용할 수 있도록 돕는다.  
이러한 기능은 공학 계산기가 가진  
지식이 없는 사용자가 어려움을 겪  
는다는 단점을 보완할 수 있다.



몰입형 인터페이스 : 일상적인 행동으로 쉽게 사용하고, 몰입하  
게 만들 수 있는 혁신적인 사용자 인터페이스

# 스토리 보드





# 전체 UI 구성

History창

이전의 수식은 여기에 계속 누적하여 나타난다.  
창을 넘어설 경우, 스크롤로 조정하여 넘어가더라도 상관없도록 구현했다.

수식입력창

raddegx!( ( ) % AC CL

Inv sin ln 7 8 9 /

pi cos log 4 5 6 \*

e tan sqrt 1 2 3 -

Ans EXP xy 0 . , EV +

마우스 버튼 입력창

도움말

도움말 창

도움말 버튼을 클릭하면, 아래에 math.js function 페이지가 뜨며, 이전실행과 이후 실행을 수행해주는 키를 만들어 ui를 더 높였다.

마우스나 키보드를 통해 수식을 입력할 수 있으며, 커서 위치에 따라 수식을 앞뒤로도 이동할 수 있고, 복사 붙여넣기 기능도 제공한다.

위의 버튼을 마우스로 클릭하여도 키보드 동작과 동일하게 수행되며, UI와 메뉴 구성은 사용자들이 많이 사용하는 구글 기본 계산기의 UI에서 차용했다.

### 3. 프로토타입 구현

[구현 방법]

## 수식 입력창 및 히스토리 창

### [마우스 입력처리]

```
$('.key').click(function(index, key){
  if($('#expression').val() == '0') $('#expression').val() = '';
  if($(this).text() == 'EV')
  { ... }
  else
  {
    if($(this).text() == 'CL')
    { ... }
    else if($(this).text()=='AC')
    { ... }
    else if($(this).text()=='rad')
    { ... }
    else if($(this).text()=='deg')
    { ... }
    else
    {
      var txtArea = document.getElementById('expression');
      var txtValue = txtArea.value;
      var selectPos = txtArea.selectionStart; // 커서 위치 지정
      var beforeTxt = txtValue.substring(0, selectPos); // 기존텍스트 ~ 커서시작점
      var afterTxt = txtValue.substring(txtArea.selectionEnd, txtValue.length);
      var addTxt = $(this).text(); // 추가 입력 할 텍스트
      txtArea.value = beforeTxt + addTxt + afterTxt;
      selectPos = selectPos + addTxt.length;
      txtArea.selectionStart = selectPos; // 커서 시작점을 추가 삽입된 텍스트 이후로
      txtArea.selectionEnd = selectPos; // 커서 끝지점을 추가 삽입된 텍스트 이후로 지
      $('#expression').focus();
    }
  }
});
```

버튼들을 key class로 선언하여 key가 눌릴 때 마다 다른 동작을 해 주도록 했고, 커서의 위치를 받아와서 커서의 이전과 이후 텍스트를 각각 저장한 뒤 key의 텍스트와 합쳐 주는 방식으로 커서의 위치까지 구현했다.

### [키보드 입력처리] 및 히스토리 창

```
$(document).ready(function(){
  $('#rad').css("color", "red");
  $('#deg').css("color", "black");

  let parser = math.parser();
  let eval_func = function(e) {
    expression = $('#expression').val();
    if(expression!=""){
      try
      {
        result = parser.eval(convertRD(expression)).toString();
        let tokens = result.split(' ');
        if(tokens[0] == 'function')
        {
          result = tokens[0];
        }
        $('#result').val($('#result').val()+expression+"="+result+"\n");
      }
      catch (e)
      {
        if(result != 'function')
        {
          $('#result').val($('#result').val()+expression+"="+e+"\n");
        }
      }
    }
  }
});
```

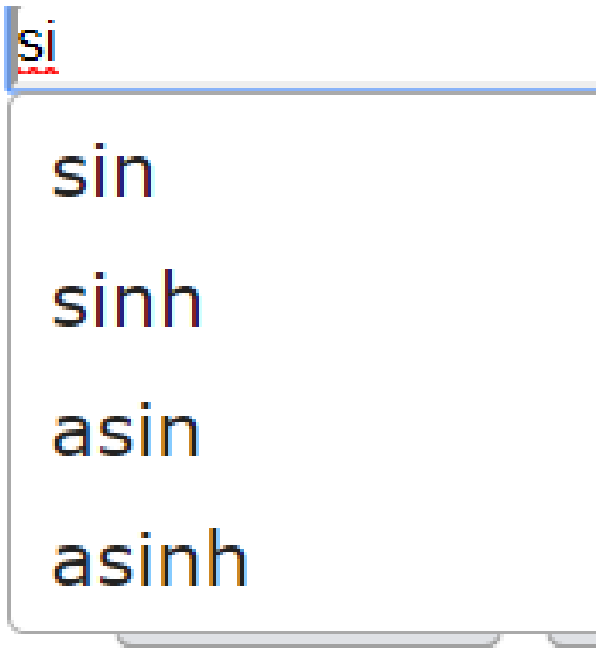
키보드 입력의 경우 수식 입력창이 input으로 선언되어 있기 때문에 계산을 할 경우에만 수식 입력창의 텍스트를 변환하여 result를 도출하는 식으로 구현했다. 또한, 결과값을 계산하게 되면, result라는 히스토리 창에 결과값 혹은 오류 값을 포함한 수식을 출력해주도록 했다.

### 3. 프로토타입 구현

[구현 방법]

## Intellisense

[구현 모습]



Si를 입력하면 si로 시작하거나 si를 포함하는 함수들의 리스트가 아래에 뜨게 된다.

[구현 방법]

```
$(function () {
    var availableTags = [
        "sin",
        "cos",
        "tan",
        "sinh",
        "cosh",
        "tanh",
        "asin",
        "acos",
        "atan",
        "asinh",
        "acosh",
        "atanh",
        "sec",
        "sech",
        "asech",
        "csc",
        "csch",
        "acsch",
        "cot",
        "coth",
        "acoth",
        "det",
        "dot",
        "cross",
        "inv",
        "pow",
        "abs",
        "cbrt",
        "exp",
        "log",
        "mod",
        "sqrt",
        "Ans"
    ];
    $('#expression').autocomplete({
        source: availableTags,
        select: function (event, ui) {
            //아이템 선택시 처리 코드
        },
        selectFirst: true,
        minLength: 2,
        open: function () {
            $(this).removeClass("ui-corner-all").addClass("ui-corner-top");
        },
        close: function () {
            $(this).removeClass("ui-corner-top").addClass("ui-corner-all");
        }
    });
});
```

함수들의 리스트를 선언해서, expression이라는 수식 입력창 밑에 입력한 텍스트를 포함할 경우, 뜨게 한다.

## Degree Radian 모드 전환

### [구현된 모습]



Radian과 degree 모드를 선택할 수 있게 해서 사용자가 쉽게 계산을 할 수 있도록 했다. 빨간색으로 나타낸 모드가 선택된 모드이다.

### [구현 방법]

```
function convertRD(e) {  
  var result = replaceAll(e, 'sin(', 'sin(' + raddeg + '*');  
  result = replaceAll(result, 'cos(', 'cos(' + raddeg + '*');  
  result = replaceAll(result, 'tan(', 'tan(' + raddeg + '*');  
  result = replaceAll(result, 'asin(', 'asin(' + raddeg + '*');  
  result = replaceAll(result, 'acos(', 'acos(' + raddeg + '*');  
  result = replaceAll(result, 'atan(', 'atan(' + raddeg + '*');  
  return result;  
}
```

```
let eval_func = function(e) {  
  expression = $('#expression').val();  
  if(expression!=""){  
    try  
    {  
  
      result = parser.eval(convertRD(expression)).toString();  
    }  
  }  
}
```

convertRD라는 삼각함수 변환 함수를 만들어 주어, radian과 degree중 선택된 모드로 변환해 주도록 했다. 그리고, 결과 값을 계산하는 코드에서 이렇게 변환된 값을 사용하도록 했다.

3. 프로토타입 구현

[구현 방법]

# 도움말

[구현 모습]



[구현 방법]

```
function help_click(){
  if( $("#help").html() == "도움말" ) {
    $("#main_frame").attr("src","https://mathjs.org/docs/reference/functions");
    $("#help").html('도움말 닫기');
    $(".hiddenmenu").show();
  }
  else{
    $("#main_frame").attr("src","");
    $("#help").html('도움말');
    $(".hiddenmenu").hide();
  }
}
function back_click(){
  history.back();
}
function forward_click(){
  history.forward();
}
```

도움말 버튼을 클릭할 경우, 도움말 닫기 버튼으로 바꾸게 해주었고, 프레임의 src를 처음에는 none인 상태로 두었다가, 도움말 버튼을 클릭할 경우에만 math.js 페이지를 띄워주게 하였다. 그리고, 방향키 버튼들도 도움말을 클릭할 경우에만 보여주게 하고, 각각 history back과 history forward 명령을 클릭시 실행 하도록 해주었다.

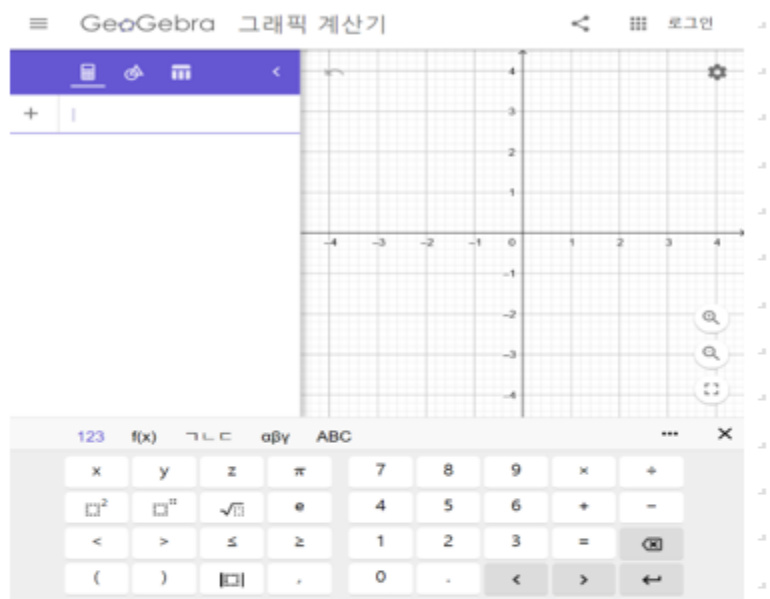
도움말 버튼을 클릭하면 math.js function 페이지가 뜨게 되고 ◀와 ▶ 키를 이용하여, 이전과 이후 페이지로 이동할 수 있다.

# 평가 계획

1. 사용성 평가.

1-1 비교 대상 공학용 계산기.

- 일 공학용 계산기 'Geo Gebra'.



1.2 사전 준비.

[과업 정의]

문제1.	$((237-261)*37/5+3813)-32+(36*23-326)*(7736/3+67)^{\circ}$
문제2.	If $x=1.33^{\circ}$ , $\tan x + \sin x = ?^{\circ}$
문제3.	$F(x)=x^4+3*x^3+21*x^2+32$ $f(2)=?$

[가설 수립]

- H1 : 마우스로만 입력이 가능한 계산기보다 사용시간이 줄어든 것이다.
- H2 : 필요한 버튼만을 제공할 때 사용자가 더 쉽게 입력을 할 것이다.
- H3 : 이전 계산 결과와 식을 보여주면 재사용성이 높을 것이다.
- H4 : degree와 radian 모드를 제공할 경우, 더 계산을 쉽게 느낄 것이다.

환경 구축 : geo gebra와 프로토 타입 모두 삼성 노트북 9 always에서 구성.

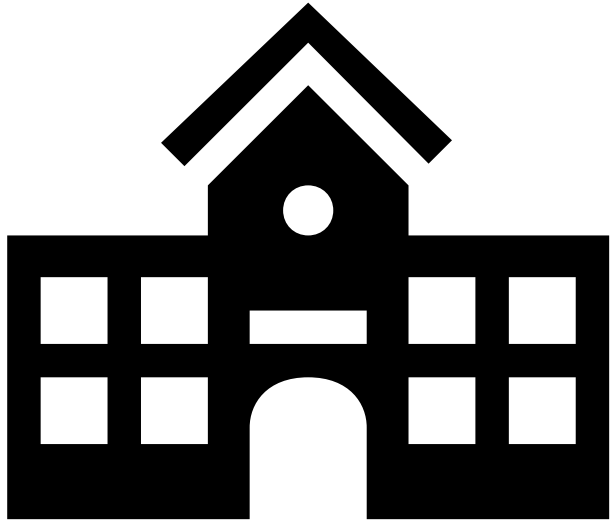
참가자 모집 : 총 4명.

- 실험자 1 : 김x배 (23세) 컴퓨터소프트웨어학과.
- 실험자 2 : 김x황 (21세) 소프트웨어학부.
- 실험자 3 : 김x석 (21세) 소프트웨어학부.
- 실험자 4 : 손x민 (20세) 소프트웨어학부.

1.3 실험 진행 순서.

번호	내용
0.	실험시 주의 사항 및 동의 여부 파악. 실험 참가 동의서 배부 및 서명. : 동의서는 담당자와 실험자 모두가 서명.
1.	실험 전 실험자 교육. Geo gebra의 사용법 교육. : 기능 설명과 시연. 프로토 타입의 사용법 교육. : 기본적인 기능 설명과 간단한 예시 시연. 과업(문제)에 대한 교육. : 한 문제를 두고 두개의 계산기를 사용하여 과업 진행.
2.	과업 진행. 첫번째 과업. : 프로토 타입을 이용하여, 과업을 진행하고 걸린 시간을 스탑 위치로 켜다. 두번째 과업. : geo gebra를 이용하여, 과업을 진행하고 걸린 시간을 스탑 위치로 켜다.
3.	과업 점검. -실험자가 도출한 답이 맞는지 확인하고, 스탑 위치를 종료. 결과를 체크 리스트에 기재한다.
4.	실험자 피드백을 위해 설문지를 배부하고 작성하게 한다.

## ■ 실험 환경

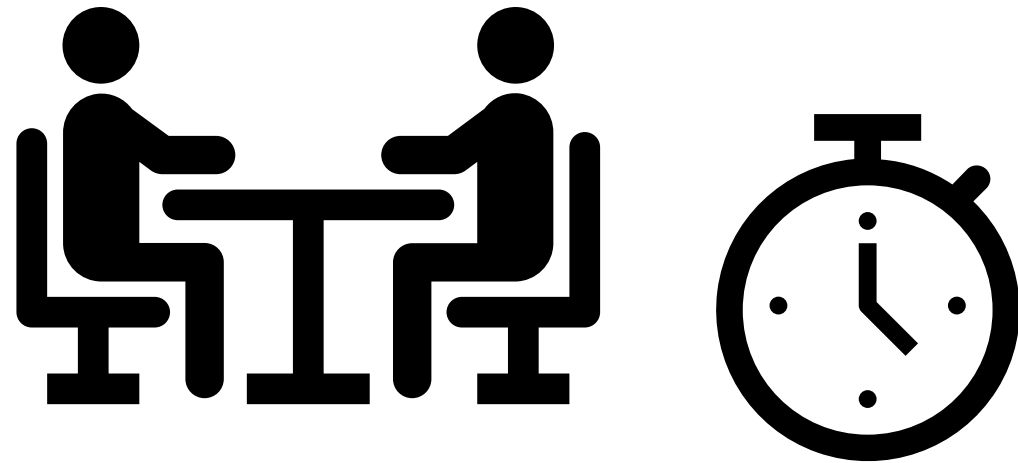


실험 장소 : 광운대학교 도서관  
(오픈 열람실)

실험 대상 : 컴퓨터 사용에 능숙한  
대학생 4명

실험 일시 : 2019.6.21-22 오후

실험 방법 : 개별 설명 후 실험 진행  
스탑워치로 해결시간 기록  
제약 사항 : 마우스와 키보드 사용



# 사용자 평가 결과

## 2. 사용자 평가 결과

	내용	답변1	답변2	답변3	답변4
1	이 계산기에 공학적 계산에 필요한 필수 기능이 충분히 있습니까?	5	5	5	5
2	이 계산기를 통해 수식이 올바르게 입력되었는지 확인할 수 있었습니까?	5	4	5	5
3	이 계산기의 각 버튼의 기능을 쉽게 알아볼 수 있었습니까?	4	5	5	5
4	잘못 작성된 수식을 수정하기에 편리했습니까?	5	3	5	5
5	이 계산기의 UI는 구성이 일관적입니까?	4	4	4	5
6	과거에 작성한 수식을 다시 사용하기 편리했습니까?	5	5	4	3
7	이 계산기의 UI의 디자인은 미적이고 단순합니까?	5	4	4	5
8	이 계산기는 잘못된 계산을 정확히 알립니까?	5	1	4	5

### [사용자 평가 결과 분석]

	내용	평균
1	이 계산기에 공학적 계산에 필요한 필수 기능이 충분히 있습니까?	5
2	이 계산기를 통해 수식이 올바르게 입력되었는지 확인할 수 있었습니까?	4.75
3	이 계산기의 각 버튼의 기능을 쉽게 알아볼 수 있었습니까?	4.75
4	잘못 작성된 수식을 수정하기에 편리했습니까?	4.5
5	이 계산기의 UI는 구성이 일관적입니까?	4.25
6	과거에 작성한 수식을 다시 사용하기 편리했습니까?	4.25
7	이 계산기의 UI의 디자인은 미적이고 단순합니까?	4.5
8	이 계산기는 잘못된 계산을 정확히 알립니까?	3.75

### UI 가설 수립

- H1. 구글 계산기의 버튼 정도만을 제공해도 사용자는 버튼이 충분하다고 느낄 것이다.
- H2. 제공한 버튼의 개수가 적당하다고 느낄 것이다.
- H3. 에러의 유형을 나타내면, 사용자가 에러를 처리하기 쉬울 것이다.

### [사용자 의견]

- 1. 폰트의 크기나 굵기가 작고 얇아서 보기 불편했다.
- 2. 오류 같은 경우 특별한 표시 ( 색상 변경, 굵기 변경)이 없어서 식별하는 데 시간이 걸렸다.
- 3. Geo gebra보다 창이 크고 제곱이나 분수 입력시 방향키 입력을 하지 않아도 된다는 점이 편했다.
- 4. 오류 메시지는 나오지만, 에러 종류를 봐도 어떤 에러인지 잘 모르겠다.

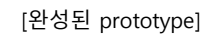
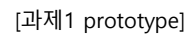
### [결론]

디자인의 경우, 8번 문항을 제외하고는 평균 4.25점 이상의 수치가 나왔다. 이 경우, 디자인과 기능적 만족도가 상당히 높은 것으로 평가되어, 필자가 다수의 사용자가 이용하는 구글 계산기와 버튼의 개수, 비슷한 UI를 제공할 경우, 만족스러워 한다는 가설을 입증했다. 그러나, 에러의 유형의 경우, 에러의 유형을 좀 더 자세히 설명하고, 특별한 표시를 하는 것이 필요하다는 결론을 도출했다.



## 프로토타입 간의 비교

[좌-과제1 prototype, 우-완성된 prototype]

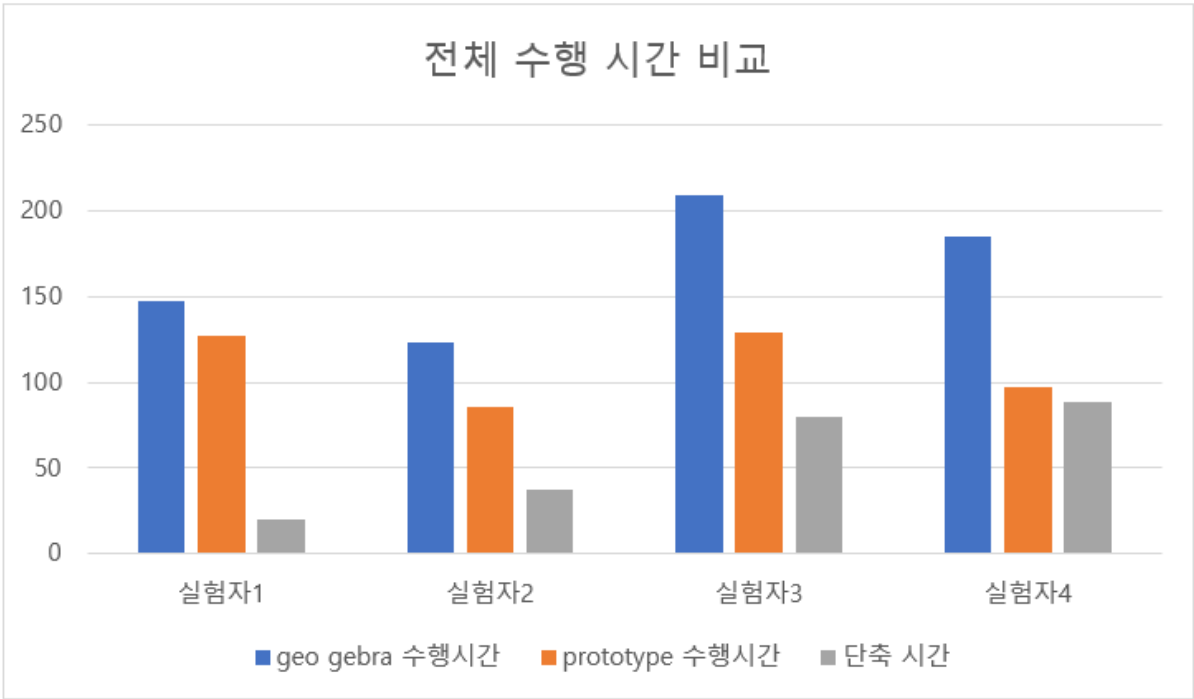


H4 : degree와 radian 모드를 제공할 경우, 더 계산을 쉽게 느낄 것이다. -> 사용자 평가 설문 3번의 만족도가 4.75점이 나옴

## Geo gebra와 prototype 비교

	실험자1		실험자2		실험자3		실험자4	
과업1	54	57	68	44	140	60	122	45
과업2	34	26	18	15	34	10	14	22
과업3	59	44	37	27	35	59	49	30

[좌-geo gebra, 우-prototype]



### [가설 및 분석]

H1 : 지오지브라의 입력창보다 클 경우 긴 수식을 입력하는 데 시간이 적게 걸릴 것이다. -> 평균 과업1 수행 시간이 지오지브라의 경우 96초, 완성된 prototype의 경우 51.5초로 평균 44.5초가 절약된 사실을 알 수 있다.

H2 : 지오지브라와 달리 분수와 제곱의 경우 방향키를 입력하지 않아도 되도록 설계했다. -> 과업 3의 경우 지오지브라의 평균 수행 시간이 45초, 프로토타입의 평균 수행시간이 40초로 비슷하지만, 방향키를 입력하지 않게 설계하는 것이 사용자의 수행시간을 5초 정도 줄여줄 수 있음을 발견했다.

H3 : degree와 radian 모드를 설정해주면, 더 계산 수행시간이 빠를 것이다. -> 지오지브라의 default 값은 degree로 설정되어 있는데, 이것을 degree와 radian 모드로 나누어 사용하게 해주는 것과 실험 결과에서는 geo gebra가 빠른 경우와 프로토타입이 빠른 경우 각각 50% 확률로 별로 다르지 않음을 발견했다.

## Ⅰ 개선방향

### [디자인]

1. 폰트의 크기나 굵기가 작고 얇아서 보기 불편했다.
2. 오류 같은 경우 특별한 표시 ( 색상 변경, 굵기 변경)이 없어서 식별하는 데 시간이 걸렸다.
3. 오류 메시지는 나오지만, 에러 종류를 봐도 어떤 에러인지 잘 모르겠다.

### [기능]

1. 그래프 기능이 없어서, 아쉬웠다는 사용자 평가가 있었다.
2. Intellisense를 제공하는 키워드가 적었다.
3. 각도 변환 함수의 에러 수정 필요 ( 스페이스 바 입력 시 에러 문제)

인간의 주의 능력은 제한되어 있어, 정보 과부하가 발생할 경우, 정보처리가 느려지므로 사용자가 원하는 상황에서만 그래프 기능이 출력되도록 하였다.

출처 : 가상현실 개념사전 concept15

폰트의 크기  
확대 및  
굵기 조정

오류 분석을  
통한 쉬운  
한글 설명  
제공

그래프 기능  
추가 및  
사용 원할 때  
출력되도록  
하기

Intellisense  
키워드 추가

각도 변환  
함수  
오류 수정

# 증빙 자료



## 설문참가자용 설명서 및 동의서

설문 주제 : 사용자 편의성을 고려한 공학용 계산기 프로그램

설문 담당자: 박선영 (광운대학교 소프트웨어학부, 학부생)

1.설문 목적 : '휴먼컴퓨터인터페이스' 과목의 학기 프로젝트로 구현한 '공학용 계산기 프로그램'을 사용자가 직접 사용하며 내린 사용성 평가를 분석하여, 사용성 향상을 하기 위함입니다.

2.연구 참여 진행 과정 : 참여의사가 있다면 설명서를 읽고 동의한 후 공학용 계산기 프로그램으로 3개의 문제를 푼 후, 설문지를 작성하게 됩니다. 설문지는 공학용 계산기 프로그램의 편의성, 사용성, 심미성 등 총 9개 평가 문항으로 구성되어 있으며 약 15분의 시간이 소요됩니다.

3. 참여 도중 중단 여부 : 참여자본은 참여 도중 언제든지 설문을 그만 둘 수 있으며, 어떠한 불이익도 없습니다. 귀하가 설문조사 중 중지 의사를 밝히면 즉시 설문을 중단하고 절희의사를 밝힐 경우 수집된 자료는 폐기됩니다.

4.개인 정보의 비밀 보장 : 이 연구에서 얻어진 개인 정보와 문항에 대한 모든 답변은 사용성 분석 이외에 사용되지 않으며, 철저히 비밀을 보장합니다.

5. 담당자 연락처 및 개인정보 관리 책임자 : 연구에서 발생한 문제, 질문이 있을 경우 다음의 연구 담당자에게 연락하십시오.

소속 : 광운대학교 소프트웨어학부

이름 : 박선영

연락처 : 010-3433-9451, [tjcdud9151@gmail.com](mailto:tjcdud9151@gmail.com)

## 동의서

- 나는 이 설명서를 읽었으며 담당자와 이에 대하여 의논하였습니다.
- 나는 연구 참여 진행 과정에 관하여 들었으며, 나의 질문에 만족할 만한 답변을 얻었습니다.
- 나는 이 연구에 참여하는 것에 대하여 자발적으로 동의합니다.
- 나는 언제라도 이 설문의 참여를 철회할 수 있고, 이 결정이 나에게 어떠한 불이익도 없다는 것을 확인하였습니다.
- 나는 나의 개인 정보와 내가 작성한 문항에 대한 모든 답변이 철저히 비밀로 유지됨을 확인했습니다.

설문참여자

성명 : 서명 :

참여 날짜 : 2019년 월 일

설문담당자

성명 : 서명 :

참여 날짜 : 2019년 월 일

## 공학용계산기 설문조사

문제1.	$((237-261) \times 37/5 + 3813) - 32 + (36 \times 23 - 326) + (7736/3 + 67)$ .	.	.
문제2.	If $x=1.33^\circ$ , $\tan x + \sin x = ?$ .	.	.
문제3.	$F(x)=x^4+3 \times x^3+21 \times x^2+32$ f(2)=?.	.	.

사용자 편의성의 정성적 평가를 위한 설문						
번호	문항 내용	① 전혀 그렇지 않다	② 그렇지 않다	③ 보통 이다	④ 약간 그렇다	⑤ 매우 그렇다
1	이 계산기에 공학적 계산에 필요한 함수 기능이 충분히 있습니까?	①	②	③	④	⑤
2	이 계산기를 통해 수식이 올바르게 입력되었는지 확인할 수 있었습니까?	①	②	③	④	⑤
3	이 계산기의 각 버튼의 기능을 쉽게 알 수 있었습니까?	①	②	③	④	⑤
4	잘못 작성한 수식을 수정하기에 편리했습니까?	①	②	③	④	⑤
5	이 계산기의 내는 구성이 알맞습니까?	①	②	③	④	⑤
6	과거에 작성했던 수식을 다시 사용하기에 편리했습니까?	①	②	③	④	⑤
7	이 계산기의 내의 디자인은 매력적이고 단순합니까?	①	②	③	④	⑤
8	이 계산기는 잘못된 계산을 정확히 알립니까?	①	②	③	④	⑤

기타 건의 및 요청사항이 있으면 아래의 여백에 적어주세요.

# | 설문 조사 내용

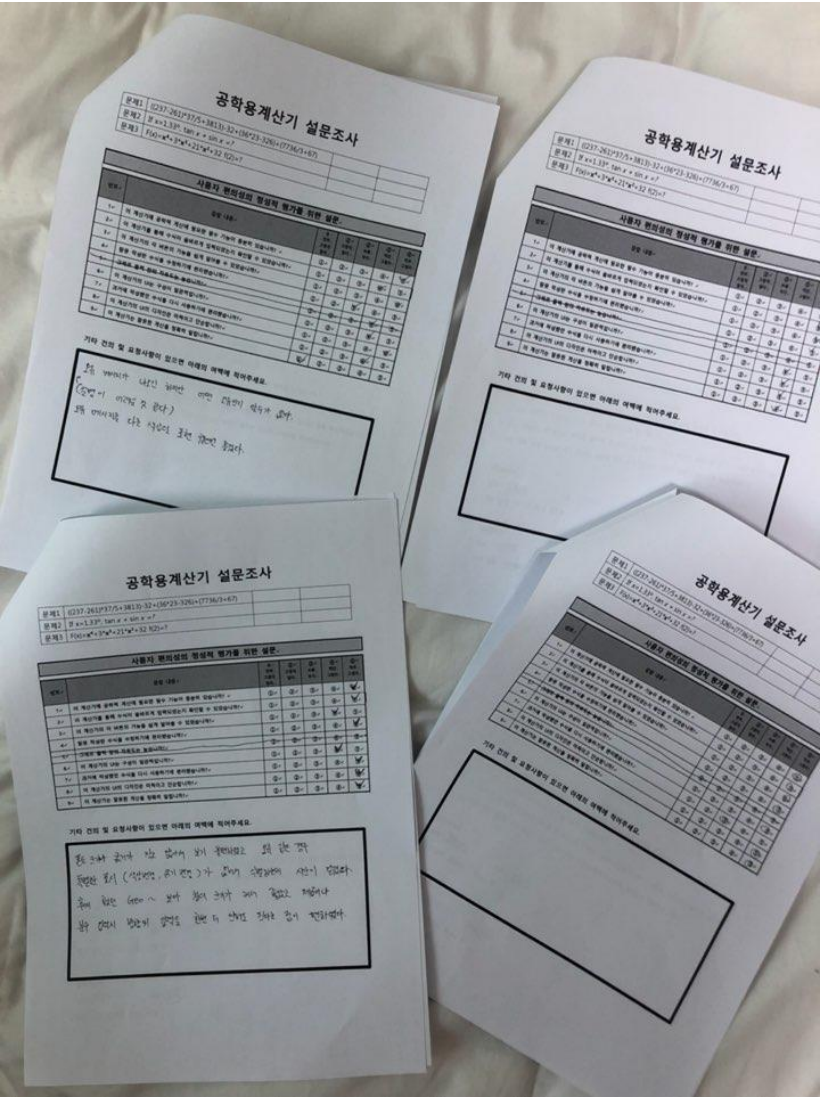
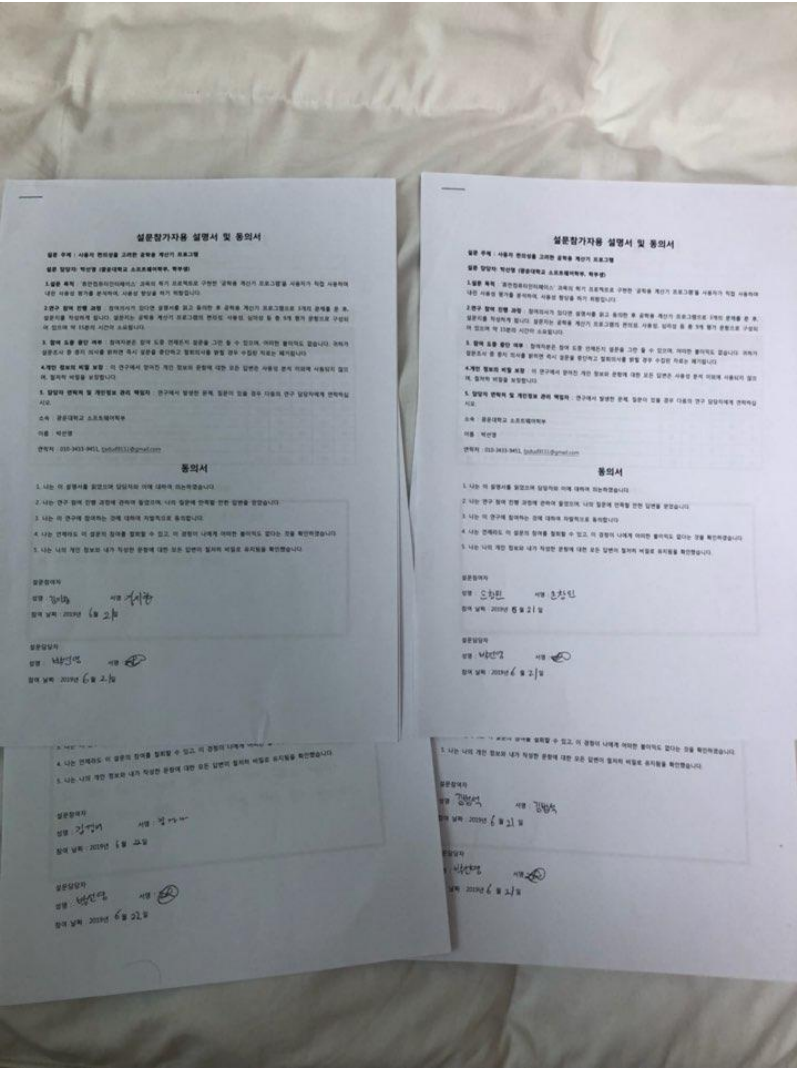


문제1↵	$((237-\underline{261}) \cdot 37/5 + 3813) - 32 + (36 \cdot 23 - 326) + (7736/3 + 67)$ ↵	↵	↵	↵
문제2↵	If $x=1.33^\circ$ , $\tan x + \sin x = ?$ ↵	↵	↵	↵
문제3↵	$F(x)=x^4+3 \cdot x^3+21 \cdot x^2+32$ $\underline{f(2)}=?$ ↵	↵	↵	↵

↵

사용자 편의성의 정성적 평가를 위한 설문↵						
번호↵	문항 내용↵	①↵ 전혀 그렇지 않다↵	②↵ 그렇지 않다↵	③↵ 보통 이다↵	④↵ 약간 그렇다↵	⑤↵ 매우 그렇다↵
1↵	이 계산기에 공학적 계산에 필요한 필수 기능이 충분히 있습니까?↵	①↵	②↵	③↵	④↵	⑤↵
2↵	이 계산기를 통해 수식이 올바르게 입력되었는지 확인할 수 있었습니까?↵	①↵	②↵	③↵	④↵	⑤↵
3↵	이 계산기의 각 버튼의 기능을 쉽게 알아볼 수 있었습니까?↵	①↵	②↵	③↵	④↵	⑤↵
4↵	잘못 작성한 수식을 수정하기에 편리했습니까?↵	①↵	②↵	③↵	④↵	⑤↵
5↵	이 계산기의 UI는 구성이 일관적입니까?↵	①↵	②↵	③↵	④↵	⑤↵
6↵	과거에 작성했던 수식을 다시 사용하기에 편리했습니까?↵	①↵	②↵	③↵	④↵	⑤↵
7↵	이 계산기의 UI의 디자인은 미적이고 단순합니까?↵	①↵	②↵	③↵	④↵	⑤↵
8↵	이 계산기는 잘못된 계산을 정확히 알립니까?↵	①↵	②↵	③↵	④↵	⑤↵

# | 사본



## 참고 문헌



1. 공학용 계산기 인터페이스 디자인의 사용성에 관한 연구, 김재현(국민대학교)
2. 몰입형 인터페이스 / 네이버 지식백과  
<https://terms.naver.com/entry.nhn?docId=2066404&cid=50305&categoryId=50305>
3. Intellisense / visual studio의 Intellisense  
<https://docs.microsoft.com/ko-kr/visualstudio/ide/using-intellisense?view=vs-2019>
4. GUI 디자인 가이드/이진호/언그라픽스, 2002
5. 가상현실 개념사전/ 정동훈 / 21세기 북스

## youtube

<https://youtu.be/DSkJnjGjP18>