휴먼 컴퓨터 인터페이스

과제명: 블록 조립 계산기

제출일:2019.05.26

학과: 소프트웨어학부

학번: 2015726079

개요

인터페이스 구현 완성도:

생성 (create) – 완성

이동 (move) – 완성

조립 (assemble) – 완성

실행 (execute) – 완성

분해 (disassemble) – 완성

복제 (duplicate) – 완성

소멸 (destroy) – 완성

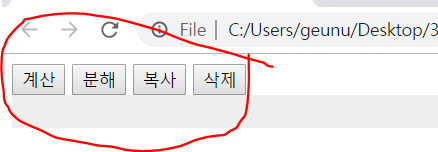
오픈소스 라이브러리 의존성:

Fabric.js만을 최대한 이용하였다.

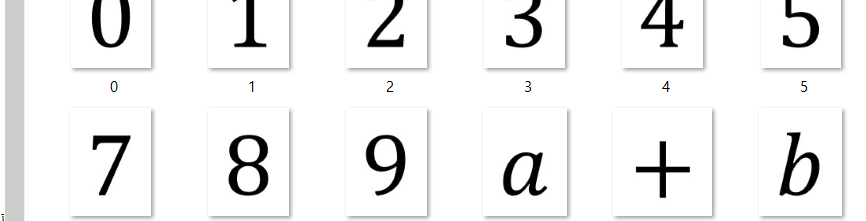
본문

-사용자 인터페이스의 구성 요소

계산, 분해, 복사, 삭제 버튼



계산기를 이루는 숫자블록들



-사용자 인터페이스의 사용방법

• 블록의 생성: 키보드를 입력하면 랜덤으로 생성.

• 블록의 이동: 블록을 누른채로 마우스를 이동

• 블록의 조합: 조합하고 싶은 블록 옆으로 이동 및 조합해 있는 세트의 블록 중 하나를 임의로 누르면 전체선택 및 조합

• 블록의 실행: 블록을 클릭 후 왼쪽 위의 계산하기 버튼을 클릭

• 블록의 소멸: 블록을 클릭 후 왼쪽위의 삭제버튼을 클릭

• 블록의 분해: 조합된 블록을 클릭 후 왼쪽 위의 분해버튼을 클릭

• 블록의 복사: 블록을 클릭 후 왼쪽위의 복사버튼을 클릭

특징적인 상호작용 방식들에 대한 세부 구현 방법

1.MathApp클래스:



MathApp클래스는 블록의 정보들에 대해가지고있다.

blocks[]: 사용자가 키를 입력한 블록 즉 화면에 나와있는 불록들의 정보에 대해 나와있다.

Selected\_block: 사용자가 마우스로 클릭한 블록에 대한정보를 담고 있다.

2. 사용자의 입력에 반응하는 이벤트 함수:



(1)사용자가 키를 입력하였을 경우

MathApp.handleKeypress(Key)가 작동을 한다.

(2)사용자가 마우스를 눌렀을 경우

MathApp.handleMouseDown(Key)가 작동을 한다

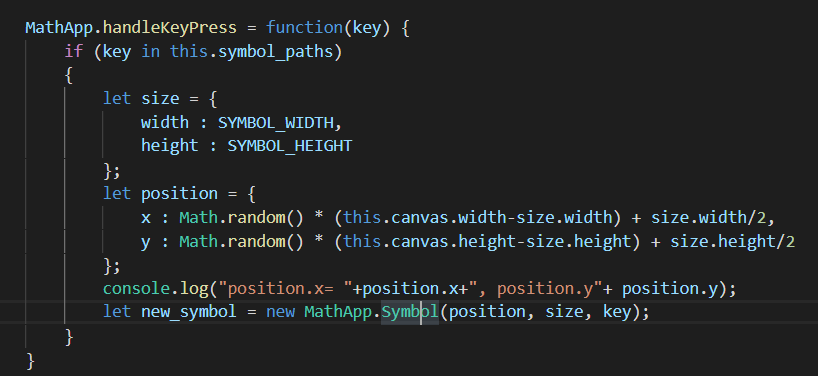
(3)사용자가 마우스를 눌렀다 땠을 경우

MathApp.handleMouseUp(Key)가 작동을 한다

(4)사용자가 마우스를 움직이는 경우

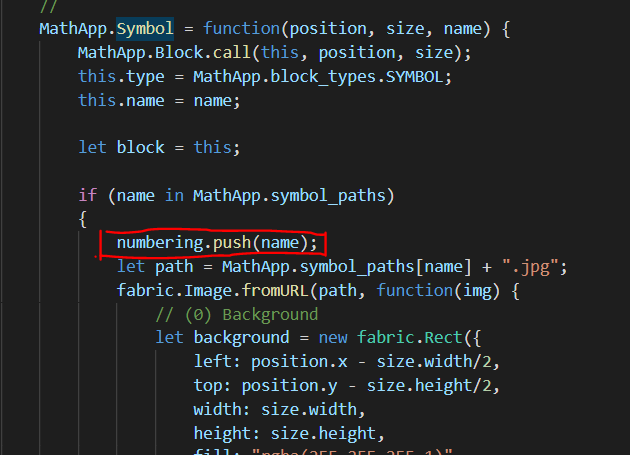
MathApp.handleMouseMove(Key)가 작동을 한다

MathApp.handleKeypress(Key)



입력된 키가 MathApp클래스가 소유하고 있는 키인지 확인한후 맞으면

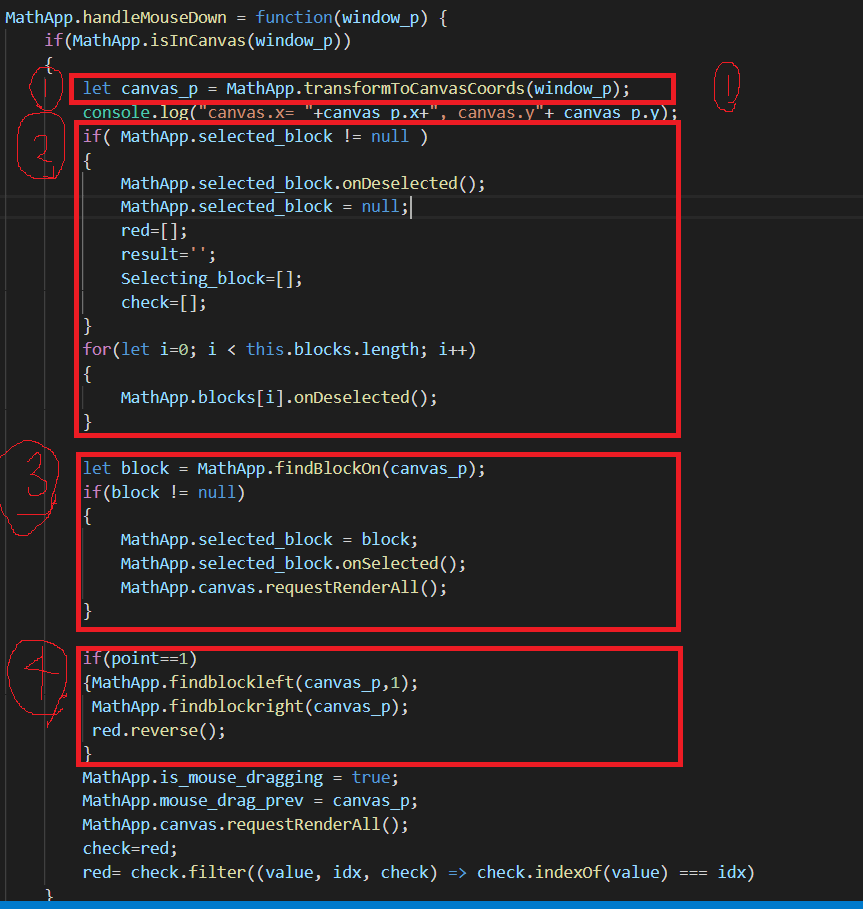
그키에 해당하는 블록의 위치와 크기를 배정해주고 MathApp.Symbol함수를 통해 생성해준다.



Symbol함수를 통해 블록의 이미지를 경로를 가져온후 생성하여주고

numbering array는 만들어지는 블록의 순서에 따른 값을 저장해준다.

MathApp.handleMouseDown(Key)



1. 사용자가 마우스를 클릭하는 순간의 좌표지점을 가져온다.
2. 사용자가 이전에 선택한 블록의 정보를 초기화하고 다시 선택할 수 있는 상태로 만든다.
3. 사용자가 클릭한 지점에 있는 블록을 찾고 onselect한다.

Onselect을 하는 순간 블록이 빨간 테두리를 뛰게 한다.

1. 사용자가 분해버튼을 클릭하지 않는다면 (분해 클릭시 point=0)) 선택 블록의 바로 옆에 인접해 있는 버튼 즉 조합된 버튼을 찾는다.

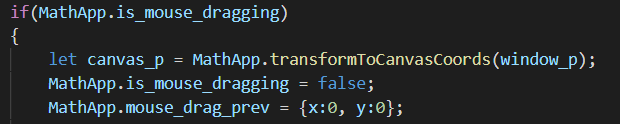
이때 red[] 배열은 현재 선택 조합된 블록의 인덱스를 담고 있으며

MathApp.findblockleft()는 선택된 블록의 왼쪽 조합블록을 찾는 역할이며 조합블록이 있을 경우 red[]에 해당 블록의 인덱스를 담으며 Onselect를 한다.

MathApp.findblockright()는 선택된 블록의 오른쪽 조합버튼을 찾는 역할을 하며 조합버튼이 있을 경우 red[]에 해당 블록의 인덱스를 담으며 Onselect를 한다.

1. 모든 작업이 끝난 후 마우스를 드래그 할 수 있게 하고 화면에 업데이트 되도록 하며 red에 중복되어 있는 값들을 제거한다.

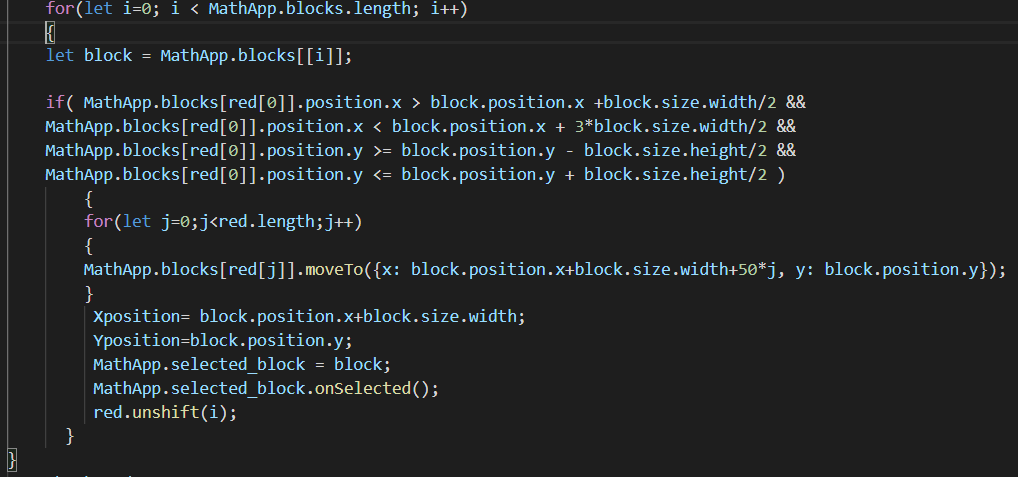
MathApp.handleMouseUp(Key)



mouse\_draaging이 true일 때 작용(Down)이 완전히 끝난 후 작용가능

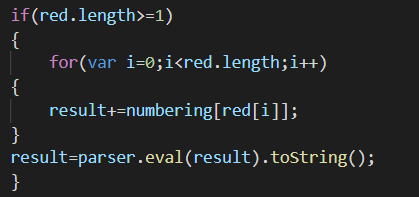
down가 마찬가지로 사용자가 마우스를 up하는 순간 좌표를 가져옴.

함수가 실행된 후 다시false상태로 변경



사용자가 선택한 조합의 블록들중 가장 왼쪽에 있는 블록을 대상으로 마우스up한 지점 왼쪽에 인접한 블록이 있는지 조사

있다면 조합의 블록을 해당블록 왼쪽으로 이동시킴 그리고 해당블록을 select하며 onselected 한 후 red의 처음 인덱스에 넣는다.

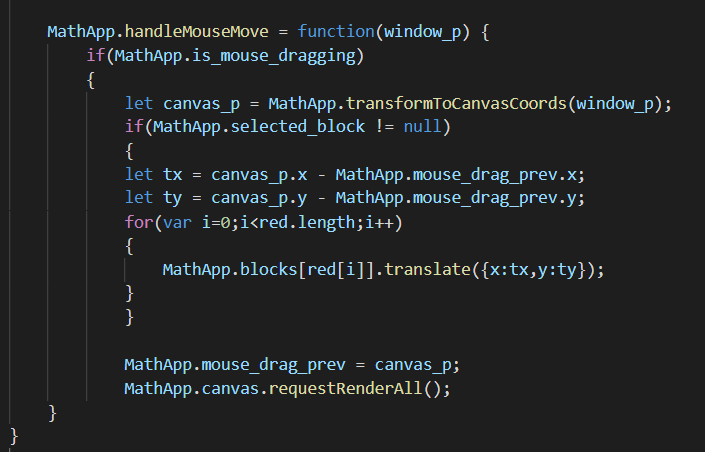


red[] 인덱스에 속한 값들을 result에 담아서 그것을 math함수들을 통해 계산한 결과들을 다시 result에 저장한다.



그후 다시 작업한결과에 관한 것을 화면에 업데이트한다.

MathApp.handleMouseMove(Key)



다음과 같이 mouse\_dragging이 true일 때 즉 사용자가 마우스로 블록을 누르고 있는 상태에서 적용된다.

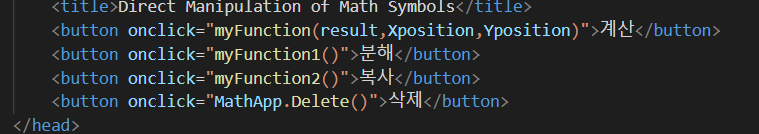
이때 움직이고 있는 좌표에 대한 정보를 가져온다.

그리고 반복문을 통해서 red[]에 포함 되어있는 블록들을 함께 움직이도록 한다.

그리고 화면 정보를 업데이트 한다.

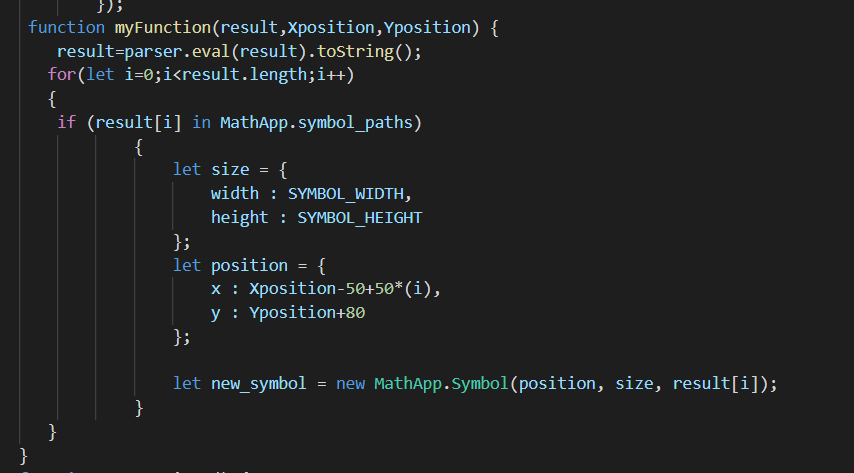
3.기능구현 버튼 메서드





1. 계산버튼
2. 분해버튼
3. 복사버튼
4. 삭제버튼

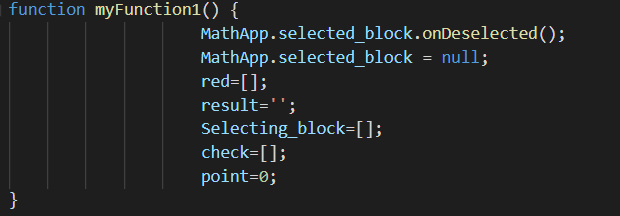
계산



MouseUp 메서드에서 선택된 블록조합들의 값들을 문자열로 result에 저장하고있는데 이것들을 계산한후 다시 문자열에 저장한다.

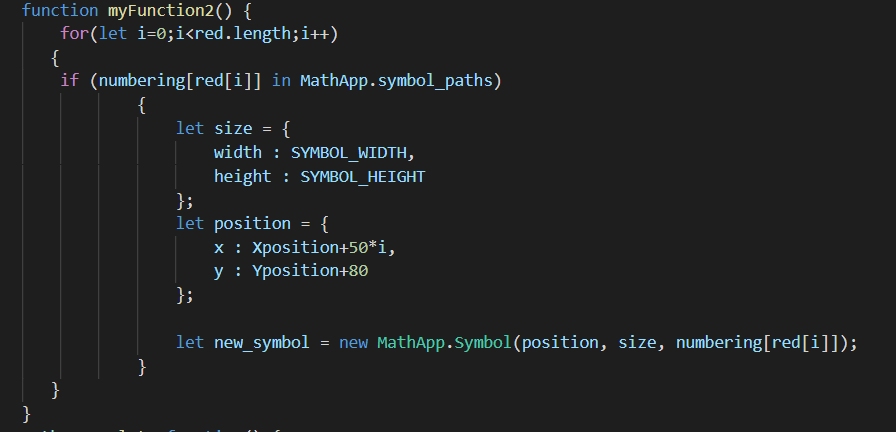
그후 반복문안에 문자열을 문자로 쪼갠후 크기와 위치를 배정 한후 해당하는 문자의 블록들을 만들어준다. 이때 만든 문자들은 모두 순서대로 인접하게 만든다.

분해



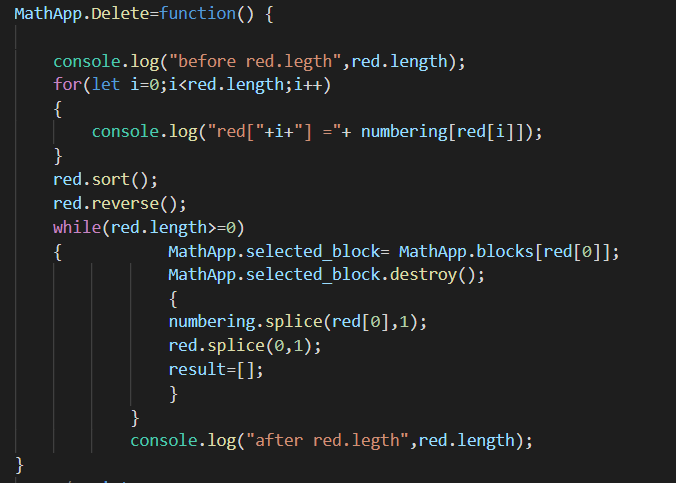
선택된 블록과 조합블록의 인덱스, 수학계산결과 등등 정보를 초기화하며 Mouse\_down() 에서는 point=1일 경우만 선택한 블록의 조합한 블록을 함께 선택하고 움직일 수 있도록 설정하였는데 point=0으로 바꿔줌으로써 선택블록 한 개만 선택되도록 한다. MouseUp()하는 순간 다시 point=1으로 바뀐다.

복사



반복문을 통해서 선택된 조합의 블록들을 그대로 새로 생성한다.

삭제



선택된 블록들을 삭제하며 이때 삭제 때마다 삭제한 블록에 속하는 red(해당인덱스)와

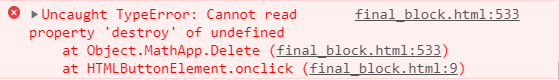
numbering(해당 블록의 값)들을 삭제한다.

구현 동영상링크: <https://youtu.be/azu_nn1TpJk>

\*논의 및 개선점

1. 버그 나는점(실패점)

(1) rendering오류(삭제버튼누를시)



삭제함수를 실행할 시에 가끔씩 이러한 오류가 뜨는 것을 볼 수 있는데 이는 삭제 도중에 한번에 조합된 블록이 삭제가 한번에 이뤄지지않아서 여러 번 삭제 버튼을 눌러야한다. 이를 해결하기 위해서 조사하였는데 이 에러가 뜨는 이유는 많은데 가장 일반적인 경우는 UI component를 렌더링 하는 동안 상태 초기화가 제대로 이루어지지 않아서 그렇다고 한다. 두번째 이유는 html이 로드 되기전에 js가 실행될 경우 이러한 오류가 날 수 있다고 한다. 동기적으로 함수가 작동하기 위해서 여러가지 방법을 취해보았지만 해결하지 못하였다.

따라서 해결하기 위해서 함수를 동기적으로 다시 구현하고 적절한 코드 배치가 필요할것같다.

(2) 마우스의 좌표가 때때로 좌표가 변한다.

이벤트가 발생할 때 동기가 적절하게 이뤄지지 않아서 그런지 y좌표가 약간 위로 설정되거나 포인트가 적절하게 맞지 않을 때 가 있다. 그래서 적절하게 drag포인트를 (0,0)으로 움직이게 할 필요가 있는데 코드안에 설정을 하니깐 적절하게 작용을 하지 못한점을 보았을 때 렌더링과정이나 cpu의 처리과정에서 시간차가 발생해서 그런것같다. 이를 해결하기 위해서 직접 마우스 드래그를 (0,0)으로 한번 클릭 시 나아지는 것 같고 조금 천천히 블록을 실행시 해결되는 것 같다.

1. 성공적인 측면

교수님이 올려주신 동영상의 프로그램과 최대한 비슷하게 구현할려고 노력하였으며 버그가 나는점을 제외하고는 생성,삭제,복사,계산,조합,이동과 같은 기능들을 완벽하게 구현하였으며 fabric라이브러리에대해 완벽하게 이해하고 실현하였다.

1. 과제결과에 대한 전반적인 자체평가

사용자가 사용하기 편한, 그리고 보기 좋은 그러한 인터페이스를 만든 것에 대한 생각을 다시 한번 해 보았고 더 나아가서 자바스크립트 프론트 개발자로써 나아감에 있어서 프로그램을 짜는데 생기는 시행착오와 고뇌, 계산, 수정 이러한 행동들을 반복하면서 저번 과제 결과물 보다는 조금더 나은 완성도가 나온 것에 대한 만족도를 느끼며 조금더 나은 개발자가 된것에 너무 뿌듯하다. 그럼에도 불구하고 recursion을 자주 활용해 복잡도가 높았으며 버그수정을 완벽히 못하는등 완벽한 프로그램을 만들지 못한 것에 대해 다시한번 생각해보며 수정을 반복해야 할 것 같다.

구현 동영상링크: <https://youtu.be/azu_nn1TpJk>