1. Fix the plus operation. (done)
   1. Only add block-2 to the right side if it used to be on the left side (done)
   2. Only flip the sign on the 3 if it moves side.
2. Make the operator dim when the 2 is dragged.
   1. I had to pass the dimmed property so many places. But I finally made it so that the plus sign dims. But for now it dims all of the time.

I need to go back and learn some JavaScript Syntax. It turns out Javascript is a lot like C. Typescript adds a lot of the features that confused me.

I probably need to make this look more like a linked list, but I need to think through the architecture some more. Chatgpt says not to keep track of prev, but to just calculate it each time. It assigns an index to each element and then uses that to find the previous element. It uses the useMemo function which is short for useMemory meaning only recalculate if things change. Actually, there is an even easier way to code this with index=findIndex() leftBlocks[index-1]. That is what I will use for now.

This works when it is on the left. Now I need to extend it to work on the right.

Now that works on the left and right

1. Update the history after dragging. I am working on this one.

This one works now.

1. Only make the blocks draggable that can be dragged at this step.

I think this is done.

1. Make operators clickable if they can be clicked.
   1. This is hard wired.
2. Add divide with simple fraction (e.g. ½)
   1. Create a fraction (A simple fraction like ½ works now. I would like to make fractions with blocks though.
3. Add divide with blocks (block-1 / block-2)
4. Add parentheses
5. Generalize the code so it can solve other equations too.
6. Add attributes to draggable blocks
   1. Side = left or right
   2. Paren Level = 0 to 100 (actually unbounded)
   3. Can I treat fractions as parens?

What are the steps of solving a simple linear algebra equation of one variable?

1. Combine like terms inside parantheses
   1. Check each operator. If it is + or -, look on either side if both sides are numbers or variables make it clickable. On click combine.
2. Get rid of any parantheses
   1. Distribute anything outside the parenthese
3. Gather like terms
   1. You can drag within paranthese
   2. You can drag if you are outside parantheses
4. Once there is a single variable term and it is alone, split it into number \* variable.
5. Then let the user divide by the number

These aren’t states because more than one can be true. It is more about looking at each term.

Do I want a function that sets draggable and clickable for each block.

Block types = variable, number, operator, equals, and pararenthesis.

Equals can’t be in leftside or right side array.

Variable can be x or 3x.

3x+2x + 5 = allow dragging of 3x and 2x to other side.

3x\*4+3 = force multiple first

3(x+2) = allow drag, but not drop because of paren

Variable Rule 1: Allow drag unless it has a multiply operator on either side

Variable Rule 2: Set paren level, only allow drop within same paren.

Number rule 1: Allow drag unless it has a multipley operator if there is a plus or minus operator on side

Number rule 2: Set paren level, only allow drop within same paren.

Operator rule 1: operators are never dragable

Operator rule 2: operator between two numbers or two variables can be clicked.

Paren rule 1: paren are never draggable

Paren rul 2: Clicking paren distributes multiplier