Change Log – Simple Calculator

1. Lessons
   1. “Enter” key will onClick focused elements. If “enter” key is causing unwanted and mysterious effects, check if the effect matches that of a focused element. A straightforward solution is to remove focus using document.activeElement.blur() upon keydown of “enter” key.
   2. Switch statements do not take “or ||” arguments.
   3. Use event listeners with componentDidMount and componentWillUnmount for basic keyboard events.
2. Rev A
   1. Imported incomplete RevA of Math Calculator;
   2. Commented out:
      1. Open and close parentheses.
3. Rev B
   1. Deleted all RevA commented out content;
   2. Currently:
      1. Clicking a number appends that number to a string;
         1. If the displayed number is 0, then clicking a number replaces that 0.
      2. Clicking AC resets all states to 0.
   3. [Actions][Resolved]
      1. [Resolved in RevC] Multiple decimals;
      2. [Resolved in RevB] Different operators used when equals is clicked;
         1. Solved with switch.
      3. Root operator:
         1. [Solved in RevB] If number is clicked first, show result; [solved in RevB]
         2. [Cancelled] If root is clicked first, show “sqrt()”;
            1. Then when number is clicked, display *inside* “sqrt()”.
      4. [Resolved in RevC] Backspace;
      5. [Cancelled] History;
   4. Progress:
      1. The add button is working;
      2. Equals button is working.
4. Rev C
   1. Deleted RevB commented content except reserved sanity checks.
   2. Progress:
      1. Multiple decimals solved with a decimal counter. If decimalCounter == 0, then append a decimal. Otherwise, nothing happens.
      2. Backspace
         1. If displayedValue.length > 1, then displayedValue.slice(0, -1) for each click;
         2. If displayedValue.length == 1 && displayedValue != 0, then setState displayedValue: “0”.
   3. [Actions][Resolved]
      1. [Forwarded] Chaining operations;
      2. [Resolved] Clicking equal resets? Give options.
         1. Decision made to reset counters, but not stored values.
      3. [Shelved] Cleaning up the counter (low priority);
5. Rev D
   1. [Actions][Resolved]
      1. [Resolved] Root event needs fixing;
      2. [Resolved] Chaining operators;
         1. Clicking an operator needs to finish previous operator’s job;
         2. Resolved by cascading values down states.
      3. [Forwarded] Backspace hitting a decimal.
      4. [Resolved] Clicking an operator needs to execute the *previous* operation. Working idea:
         1. Store values in state;
         2. Within operator, call equals() to calculate stored values.
6. Rev E
   1. Progress
      1. Solved chaining operators. Needed to store values in state and evaluate stored values upon executing proceeding operator.
      2. Root:
         1. Skipped calling equals(),
         2. Calculated directly within root().
   2. [Actions][Resolved]
      1. [Resolved] Equals button needs to call a different event than equals();
         1. Why is equalsButton() “not a function?”
         2. Forgot to bind equalsButton() in the constructor.
7. Rev F
   1. Progress
      1. Keyboard Events
         1. Figured out how to pass keyboard inputs to displayedValue, but maybe still better to bind keyboard inputs to <button>s;
         2. Figured out how to connect keyboard presses to events using componentDidMount, componentWillUnmount, and document.add/removeEventListener.
   2. [Actions][Resolved]
      1. [Resolved] Clean up first:
         1. [Resolved] Streamline counter iteration;
            1. Streamlined for numbers, kept operators as is cz mafan.
         2. [Resolved] Streamline onClick event handlers;
      2. [Resolved] Bind number buttons to keyboard;
         1. Try event listeners and component did mount and such;
         2. Try onKeyPress in the <button>s
         3. Solution: eventListeners to sense keypress, and bound keypress to onClick buttons.
      3. [Resolved] Bind operator buttons to keyboard;
         1. Need to find the power keyCode;
         2. Need to figure out how to incorprate shift + keyCode;
      4. [Resolved] Decimal
         1. Add decimal button keyboard bind
         2. Limit one decimal
      5. [Resolved] Equals
         1. [Resolved] Link numpad enter button;
            1. Lol it’s suddenly working for some reason.
         2. [Resolved] Fix keyboard equals button;
            1. [Resolved] operatorCounter not resetting;
            2. [Note] Shift events use same keyCode as non-shift events, only the keyboard position matters, so needed to create if-else statement to capture when the shift key is pressed.
      6. [Cancelled] Power button needs to reset displayValue;
      7. [Resolved] Operators
         1. [Resolved] Repeatedly clicking an operator should have no effect;
         2. [Resolved] Switching between operators doesn’t quite work yet;
            1. Wasn’t the operator’s problem.
      8. [Resolved] Enter key resets everything for some reason.
         1. [Solution] <Buttons> clicked via mouse maintains focus. Hitting “enter”, in addition to executing code, also onClicks the focused element. So if focus is on the AC <button>, then hitting “enter” will trigger that <button> (if focus is on a number <button>, then hitting “enter” will activate that number <button>). Solution was to remove focus from all active elements when equals event is triggered since “enter” is set to trigger equals event.
      9. [Resolved] If displayValue = 0, then need to overwrite.
      10. [Resolved] Finish backspace button.
8. Rev G
   1. Main issues resolved, now to work on styling and streamlining;
   2. Progress:
      1. Cleaned up commented out code;
   3. [Actions]
      1. [Resolved] Alternating operators should have no effect on displayedValue;
      2. [Cancelled] Streamline process of updating displayedValue;
         1. Each button calls their event handler, but their event handler updates a state;
         2. Then a switch statement, using the state, appends the numbers to displayedValue.
      3. [Carried] Delete unused events and such;
      4. [Cancelled] Decimals;
         1. [Resolved] Buttons to move decimals;
         2. [Cancelled] Start with no decimals, 3 decimals on enter, and user can manipulate decimals at beginning if they want;
         3. [Cancelled] Try making the if-else statement with calculation within equalsbutton();
      5. [Carried] Re-style;
         1. No history button;
         2. Fix spacing for “0”;
      6. [Carried] Understand componentDidMount() & componentWillUnmount()
      7. [Carried] Add comments everywhere;
9. Rev H
   1. Changing decimal place buttons to x3, xy, and invert.
   2. [Actions]
      1. [Resolved] Clicking square or root after an arithmatic operation should apply square or root on the result;
      2. [Resolved] If backspace deletes a decimal, reset decimal counter,
      3. [Resolved] Undo button
         1. Use an array
      4. [Resolved] Remove “cube” button;
      5. [Resolved] Add “Ans” button;
      6. [Resolved] Shift buttons around;
      7. [Resolved] Bind new buttons to keyboard events;
10. Rev I
    1. Javascript finished (almost)!
    2. Removed “cube” button;
    3. Changed “square” operation to instantaneous.
    4. [Actions]
       1. [Resolved] Add user-guide for keyboard shortcuts.
       2. Styling:
          1. Sizing of buttons when height changes;
          2. Responsive sizing;
          3. Button highlighting during click and keyboard events.
       3. Streamline:
          1. Organization;
          2. Calls and references;
          3. Comments.
       4. General
          1. Understand componentDidMount and componentWillUnmount
    5. Future considerations
       1. decimalCounter, instead of manually iterated by each event, can be better deployed by actually counting decimals in displayedValue.
       2. Operators, instead of having individual events, can be combined into one event with a switch expression directing the specific operations.