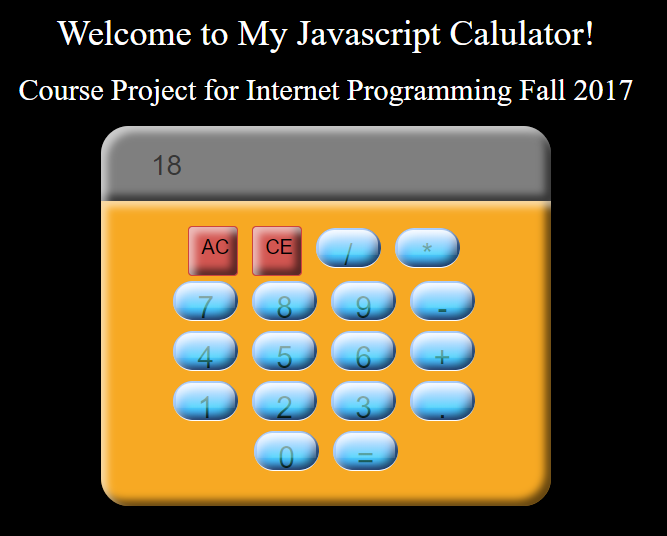
Joseph Sauceda

**Course Project Report**

The project in this folder is a calculator. It contains the Javascript and CSS in separate files linked into the HTML code. Below is an explanation of the design and implementation of the project.

**Design:**

I chose to use orange and black to resemble the UTRGV colors for the interface. In the HTML file, I used Bootstrap classes to make use their streamlined button placement using classes like “btn –btn-primary” for white buttons and “btn btn-danger” for red buttons. That default look isn’t seen for the most part because it was changed in favor of a glossy button style with animated size transitions in the end.



**Implementation:**

JQuery’s has an event system that can attach a listener to an HTML element, and that is what I used to have functions called when the element was clicked. At the start of the script, I have an array that I use to store input which is then used to store both operators (which are checked for duplicity) and numbers that are pushing into the array after being compared against both an array of numbers between 0 through 9 and an array of operators. The “=” button calls a different function to evaluate the expression using JavaScript’s eval() function. The AC and CE button clear the input array with an empty string and pop off of the top of the input array(for backspacing) respectively.

**Documentation:**

**Inputs[]:** This is where the user input is stored as an array.

**totalString**: This variable holds the joined expression that will be assigned to the HTML content of the anchor steps element.

**Operators1**: This array holds the operators as strings.

**Operators2**: This array holds the dot operator as a string.

**Nums[]**:This array holds the valid number inputs as strings.

**getValue(input):** Takes the id of the button that was clicked to compare it against the valid operators, checks the dot operator for duplicity and verifies that there are not consecutive operators. It also checks to see if the input is one of the operands before pushing the input into the inputs array.

**update():** Joins the inputs string using JavaScript’s join function and assigns it to the HTML content of the steps element.

**getTotal():** Joins the inputs string using JavaScript’s join function and assigns an evaluation of the joined string using JavaScript’s eval function and assignes the string to the HTML content of the steps element.

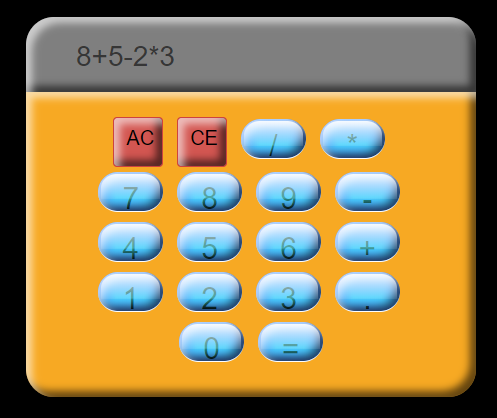
**$("a").on("click",function():** This is the event handler function that listens for a click. When this function is called:

If the button is the AC or AllClear button, reassign the inputs string with “”. Call update().

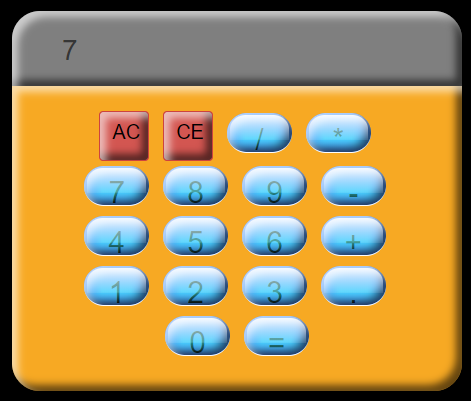
If the button is the CE or ClearEntry button, pop off of the top of the inputs array. Call update();

If the button clicked is the equal button, call the getTotal() function.

If the button clicked is any of the operators, call the getValue() function and check to see if there aren’t any double decimals or other operators.



The input array is always evaluated as a whole whenever the equal sign is clicked.



Clicking “\*” followed by “3” after this would result in 8+5-2\*3\*3 which would be -5 as opposed to 21.