Test documentation for Calculator application.

Basic tests for calculator function include:

Ensuring the addition/subtraction/multiplication/division function works logically.

Clear function clears the current numbers being input into the application.

Added in memory store and output display function.

Ensure user input matches the output display and that the output display shows the history of previous input.

Ensure the “MS” button stores the selected input under the “memory store” tab, which can then be used for future calculations.

Make sure both the buttons on memory clear under memory store and history clear under output display clears the data in display table for both.

Points for improvement:

“MS” button turns completely black after using it once, have yet to fix the bug.

Did not have much time to alter the CSS to make it look more pleasing.

Could have made the display bigger, would be better for people that can’t see well(?)

Test documentation for Snake game.

Only had time to make a few basic adjustments to the game so there isn’t much change to the original game.

Basic tests for the snake game include:

Making sure the UP DOWN LEFT RIGHT arrow keys are working.

Current state of game ends if the snake crashes into the wall OR if it eats a part of its body. Check that both options end the game rather than continuing.

Check that the score updates accordingly. (current version of the game increases score by 10 points)

Points for improvement:

Added in the boxes for the functions (Start, Stop, Pause, Speed up, Slow down) but yet to input the code for the respective buttons to work.

There are many possibilities to improving the snake game as shown by other members of the team, for example to include a new “food” function that displays intermittently which can grant more points.

An idea I did not have the time to implement would be to have different difficulties. As the difficulty increases, so would the number of obstacles increase in the current canvas. The obstacles function similar to the wall function, if you hit the obstacles, the game ends.

Another idea would be to remove certain parts of the initial wall so that when the snake passes through it, the snake appears on the other side of the wall rather than ending the game. What I mean is if the snake goes through the right side of the canvas, it appears back on the left side and continues the game, if it goes through the top side of the canvas, it reappears back from the bottom. Vice versa. Like a portal function in the latest version of snake games.