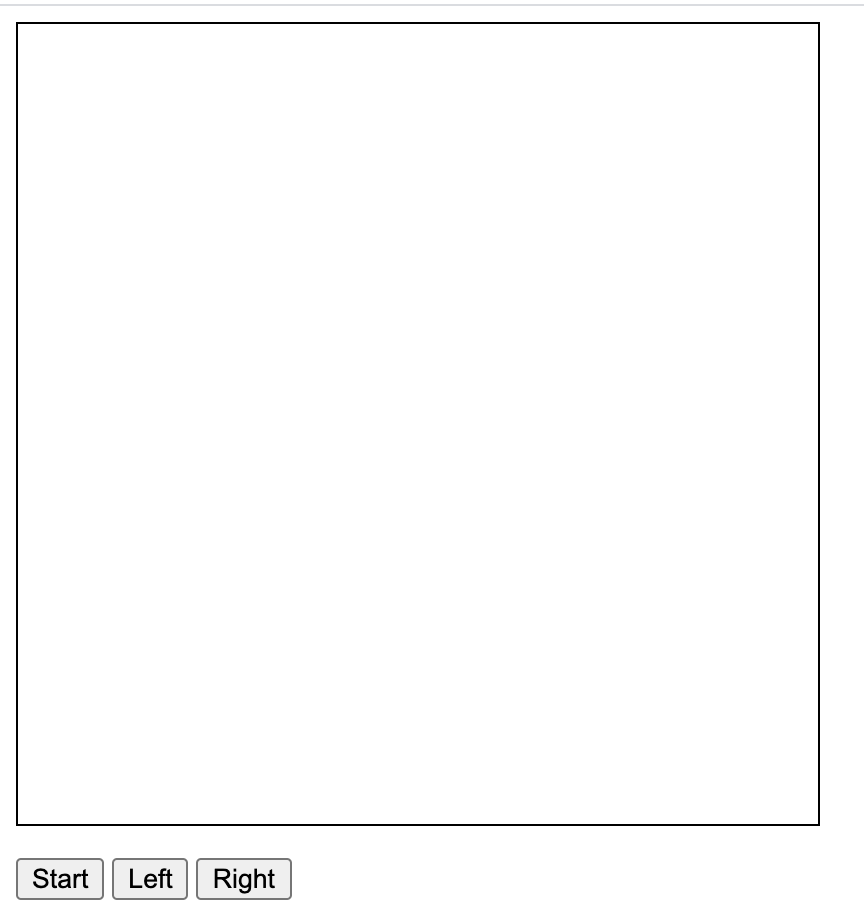
1. README
   1. Task 1: Press “Start” button to start the game. Press “Left” or “Right” button to change directions. Press “Stop” button to pause the game.
   2. Run the command: node HW3.js

Note: You might need to install “readline-sync” with the following command:

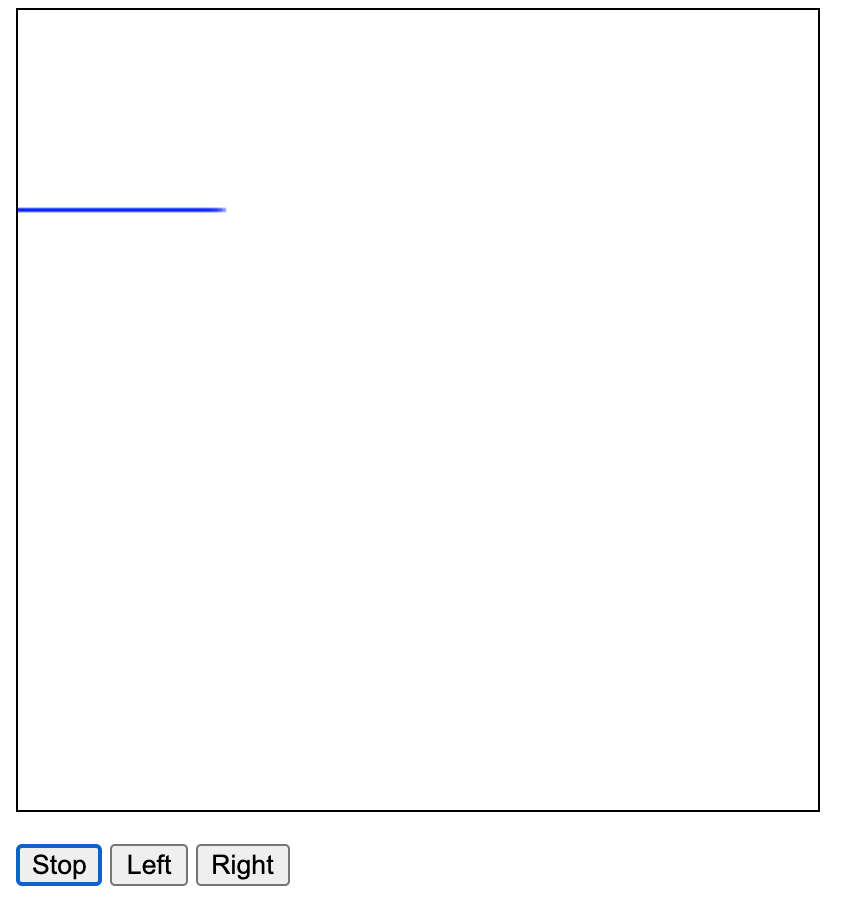
“ npm install --save readline-sync ”

1. Task 1:
   1. Start up screen:



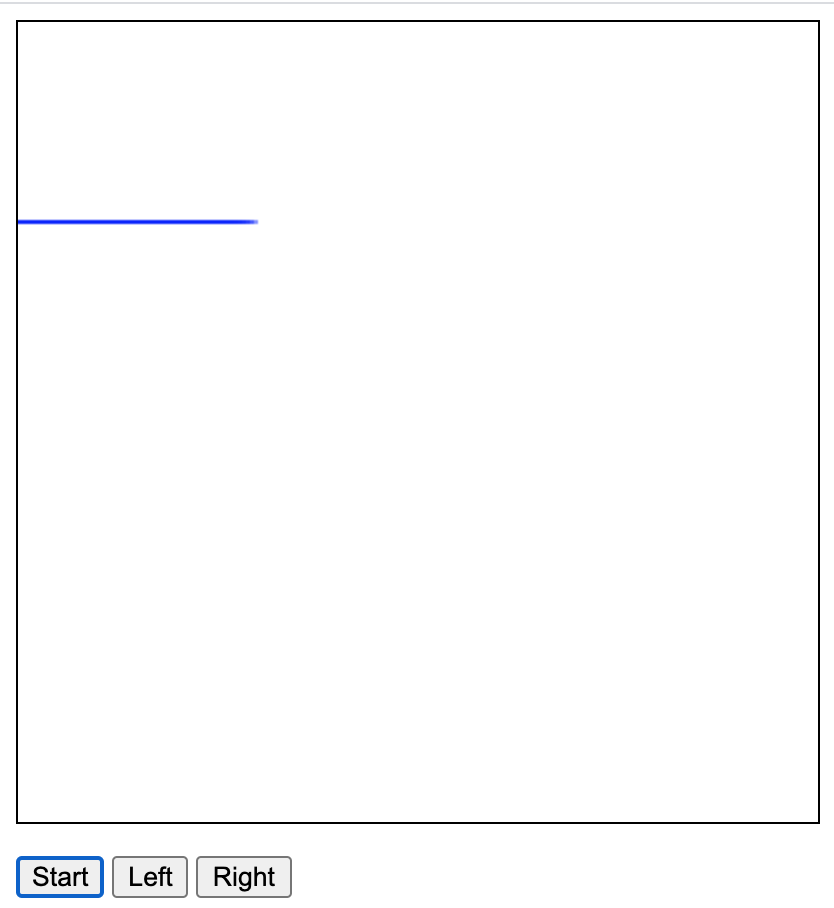
Built this functionality by create an html page with a canvas screen and 3 buttons as objects.

* 1. Screen when simulation starts: Change in the start button, and start of the snake



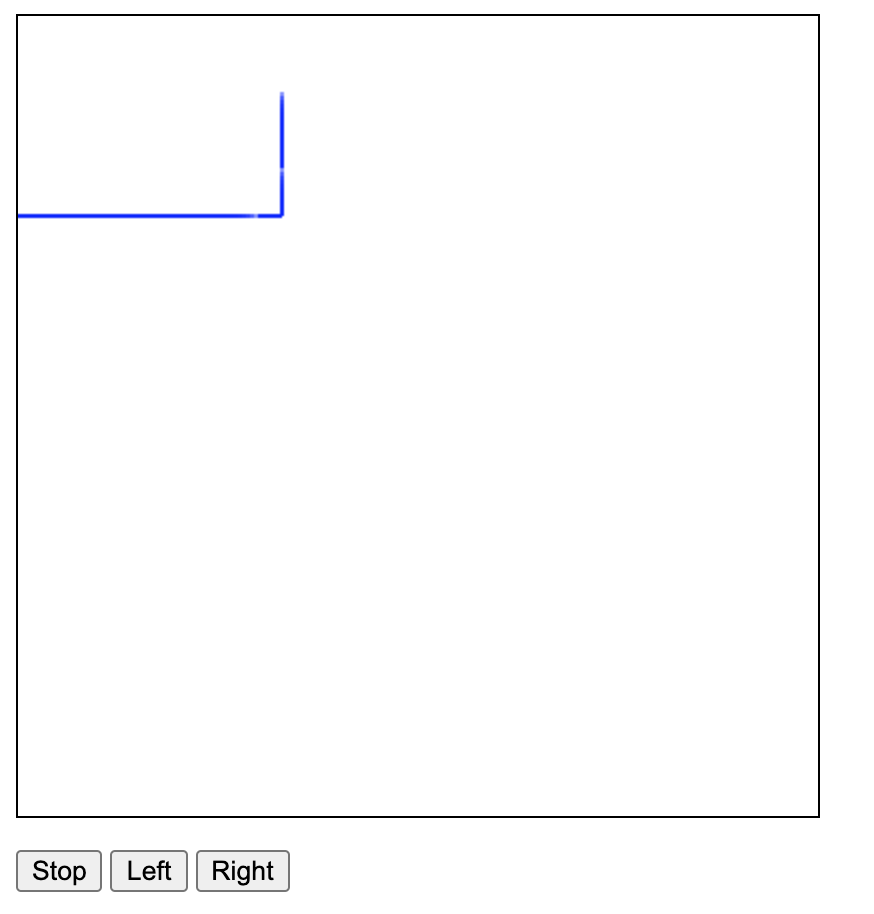
Constantly wrote on the canvas using the setInterval() module, and drawing small lines to form a snake when the start button was pressed.

* 1. Stopping of the screen: button changed back to start, and snake’s progress is paused



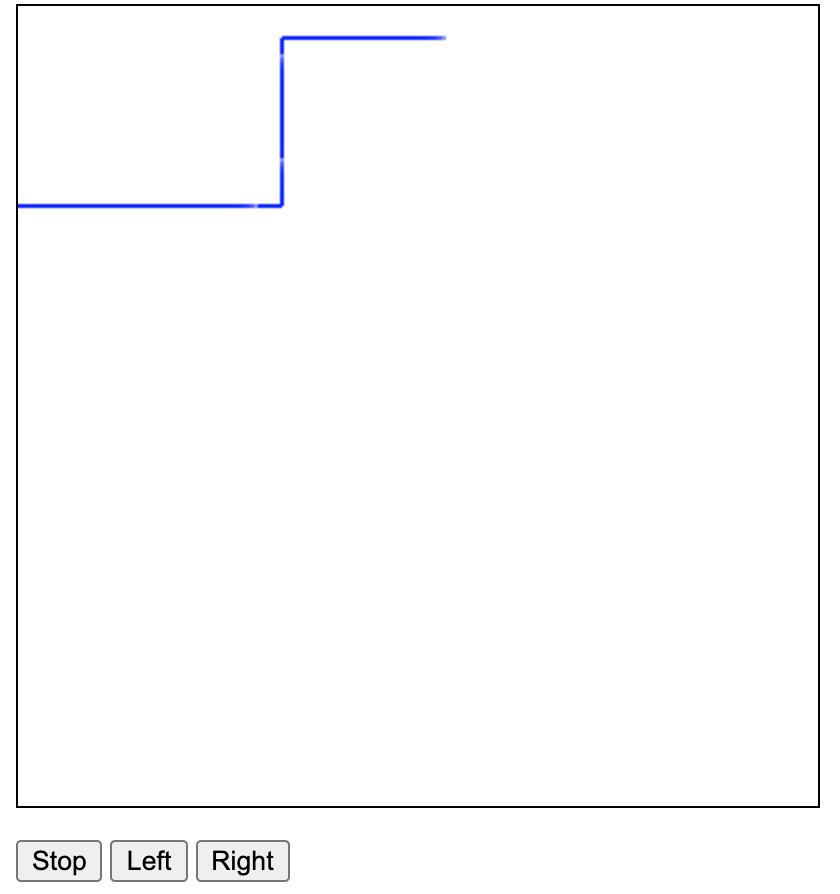
Called the clearInterval() module when the user pressed stop, while keeping track of the location of the head of the snake on the canvas.

* 1. Turning the snake to left direction: left button pressed and the snake moves up



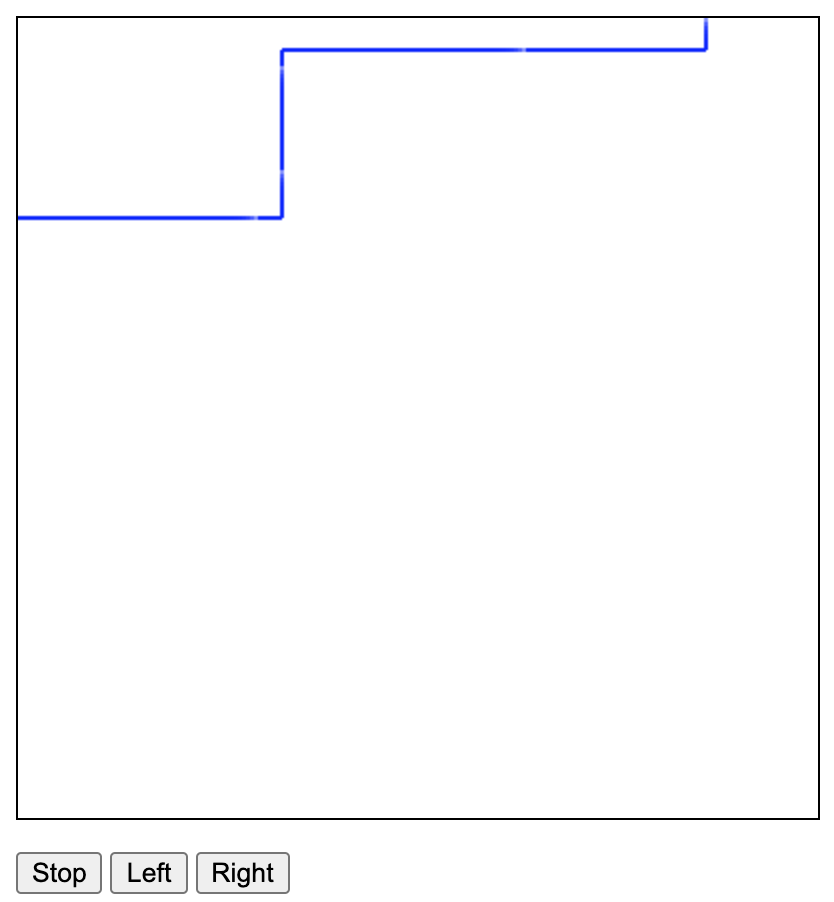
When the left button is pressed, the change in x value and the change in y value is updated to turn the snake to the left direction.

* 1. Turning of the snake right: right button pressed and the snake goes in the horizontal direction



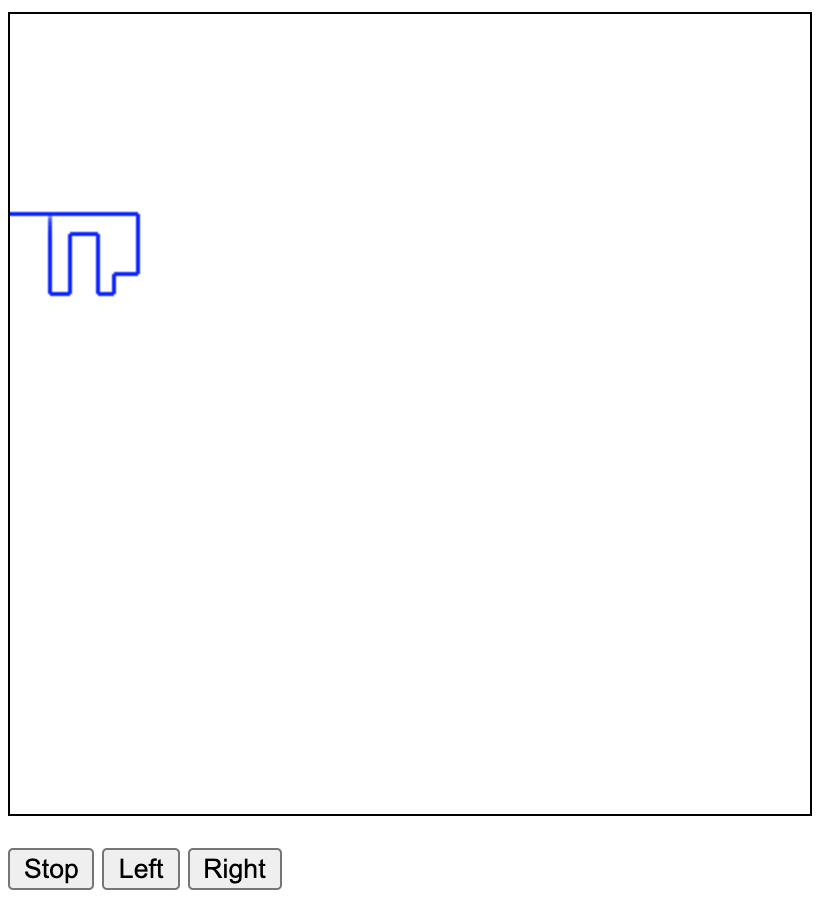
When the right button is pressed, the change in x value and the change in y value is updated to turn the snake to the right direction.

* 1. Reaching an dead end: Snake stops moving



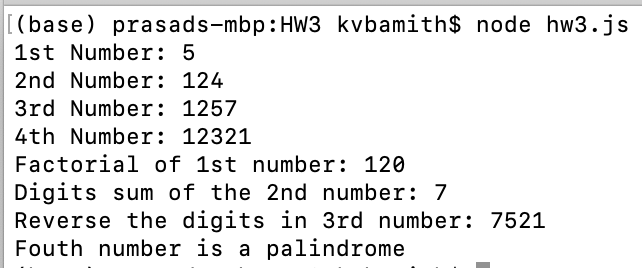
When the snake’s head value equals the size value of the canvas, then stop the running of the game by calling clearInterval()

* 1. Snake’s head touching the body: stops the game

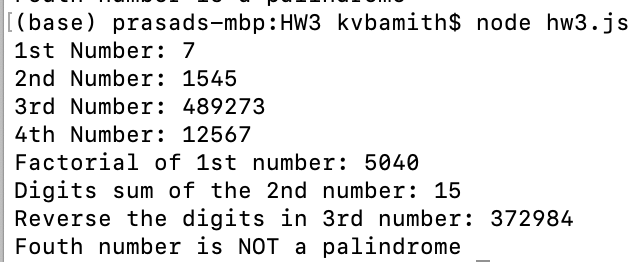


Constantly checked the color of the upcoming pixel in the canvas. If it was blue, then we know we already reached there, so we stop the game by calling clearInterval().

1. Task 2:
   1. When the fourth number is a palindrome:



* 1. When fourth number is not a palindrome:



* Got the factorial of the first number by going through a for loop and multiplying values from 1 to n, where n is the 1st number, which is n!.
* Got the sum of the digits by going through a while loop and separating each digit by taking the modules of the number by 10, and adding the digits up.
* Reversed the digits, by getting each digit like the last method, and adding each digit to a string in the opposite direction.
* Found if a number was a palindrome by reversing the digits like the last problem, and checking if it is the same as the number given to us.