Devin Amdahl

Ali Jannesari

SE 319

1 Mar. 2022

Homework 3: NodeJS

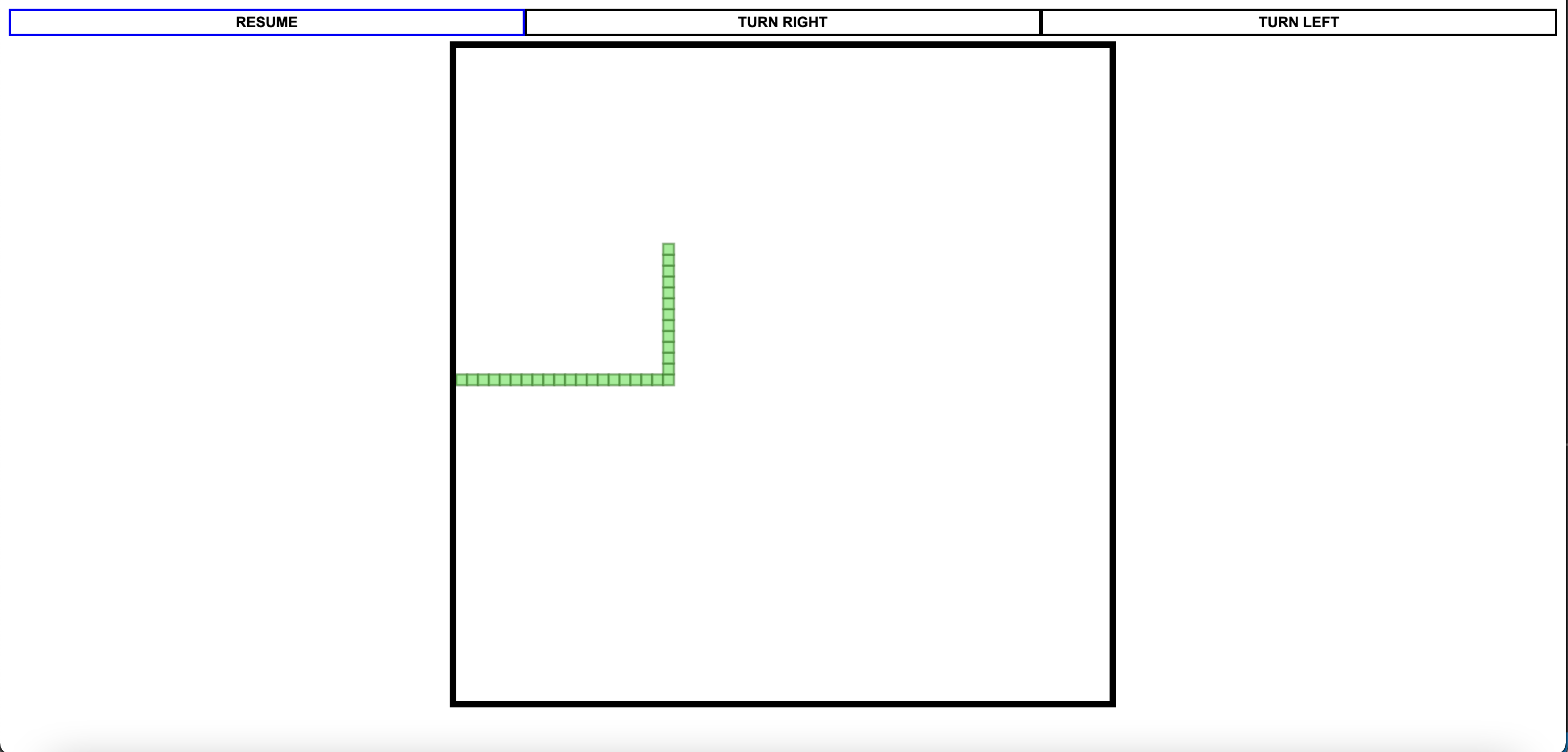
Firstly, I know I am submitting this assignment late, and I apologize for that. I have been dealing with some family events that have made it difficult to sit down and work on school. At the beginning of task one, I was feeling a little bit lost as to creating a dynamic game such as ‘Snake’ using HTML, CSS, and JavaScript. Thus, I began by reading through a few different tutorials about creating a ‘Snake’ game using these three languages. However, I only followed these tutorials until I felt as if I had a solid enough understanding of HTML 5’s canvas to then be able to print my snake in a way that resembled the old ‘Snake’ games.

From there, I began solving the remaining required functionality on my own, which was surprisingly enjoyable. I began by creating the outline of the page using CSS and HTML. This consisted of the three buttons and the blank canvas. From there, I began by adding the switching of the first button from ‘Start’ to ‘Stop’. I then handled the drawing of the snake, incorporating a restart and resume button, and finished by handling the constraints that can end the game. While handling the borders was easy, it took me a while to come up with a working solution for handling when the snake would cross over its own body. While a ‘Snake’ game, especially one like the one that we were tasked with creating, is rather simple, I am still a bit new to JavaScript so attempting to solve the issues intuitively was interesting. I am pleased with my solution.

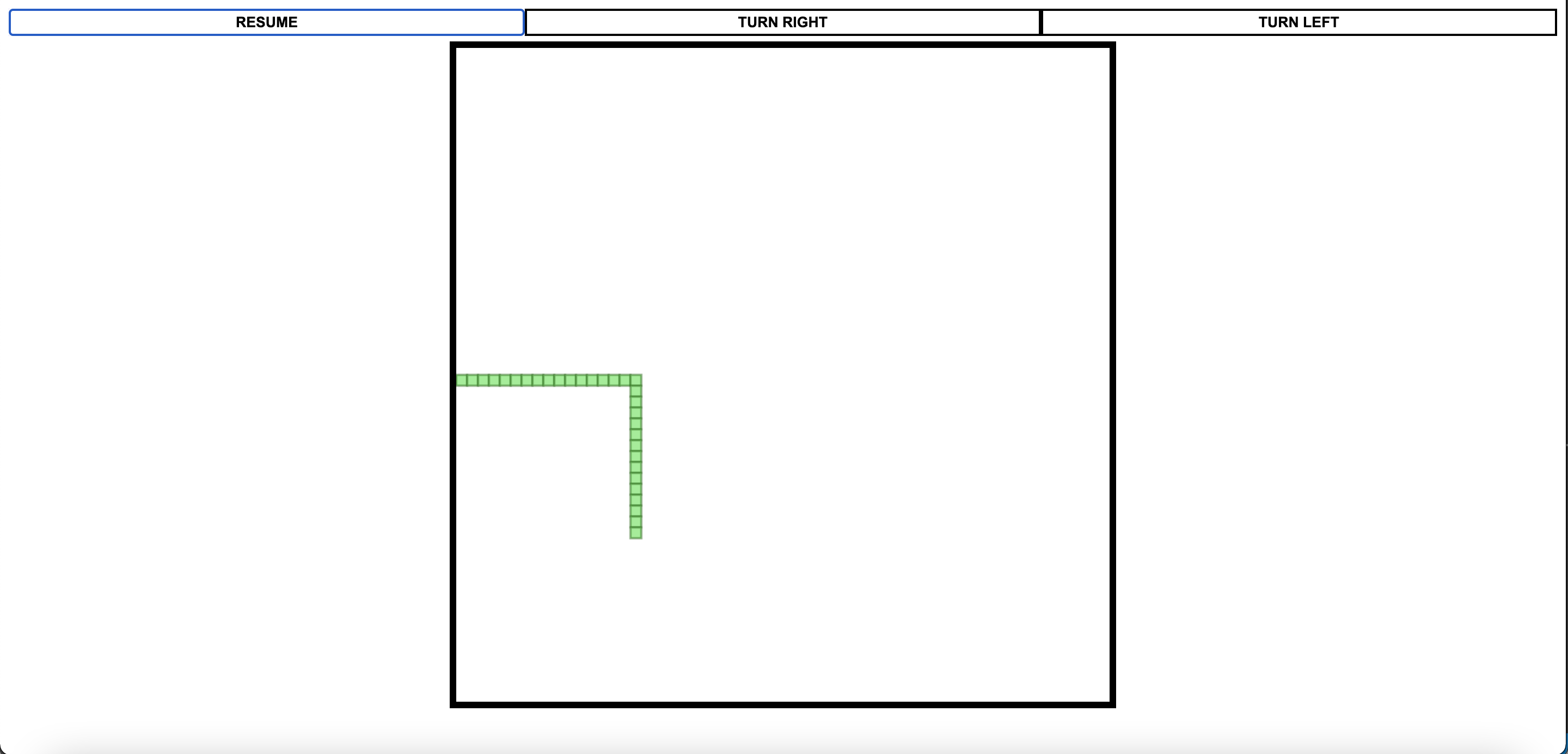
As for task two, I found the functionality that was asked of us to complete to be very easy. The requirements were very similar to those of our fourth lab. I do not have much to say about my solution approach for this task as it was rather straightforward. I began by accepting the user’s input, and then worked on implementing the four functions required to calculate the required outputs. All of these functions were rather straightforward, and while my ‘isPalindrome()’ function could be more efficient, I simply wanted to save the time and utilize the ‘reverseInt()’ function that I had written for the third required operation.

**Screenshots of every required output:**

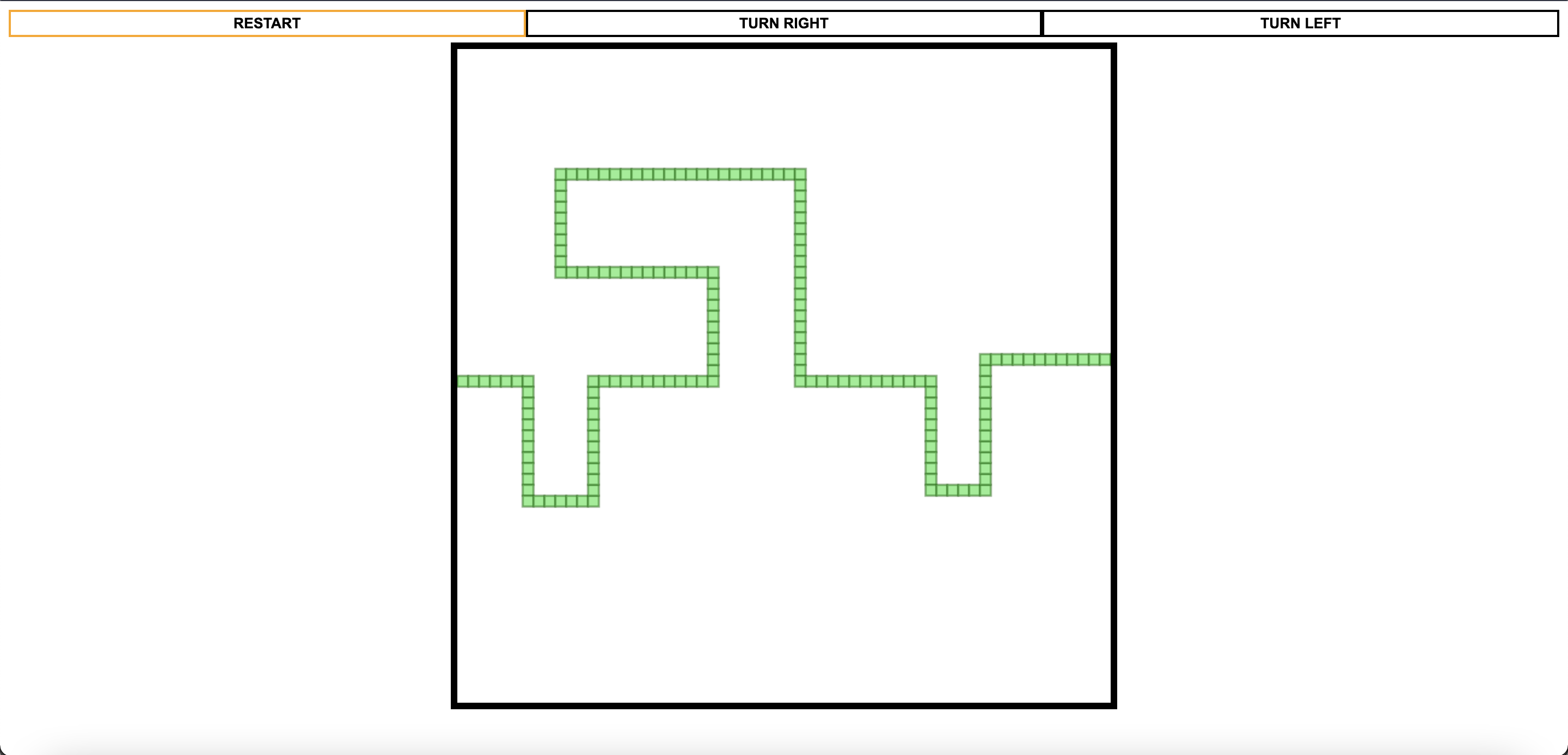
* Task 1:
  + The snake being able to bend left.



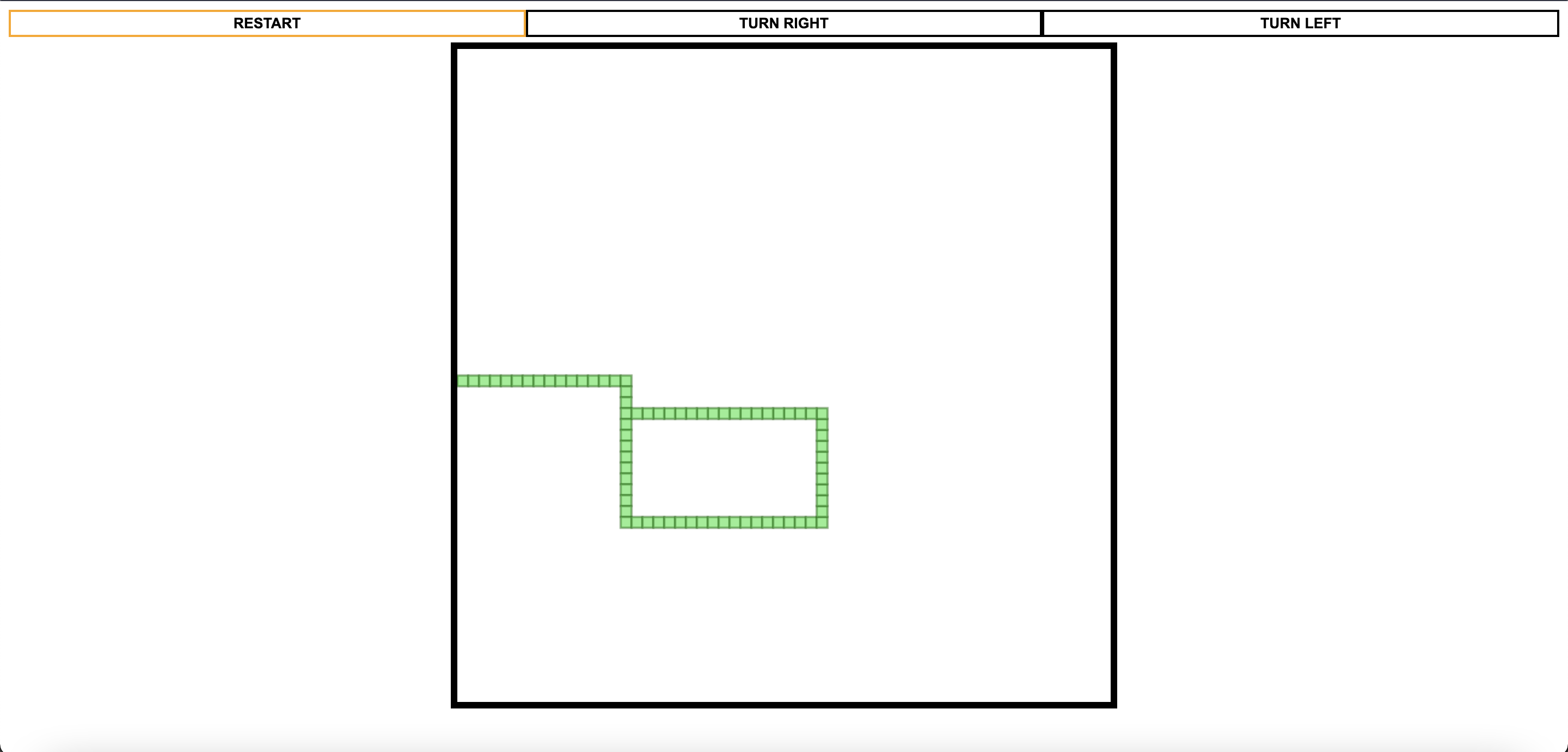
* + The snake being able to bend right.



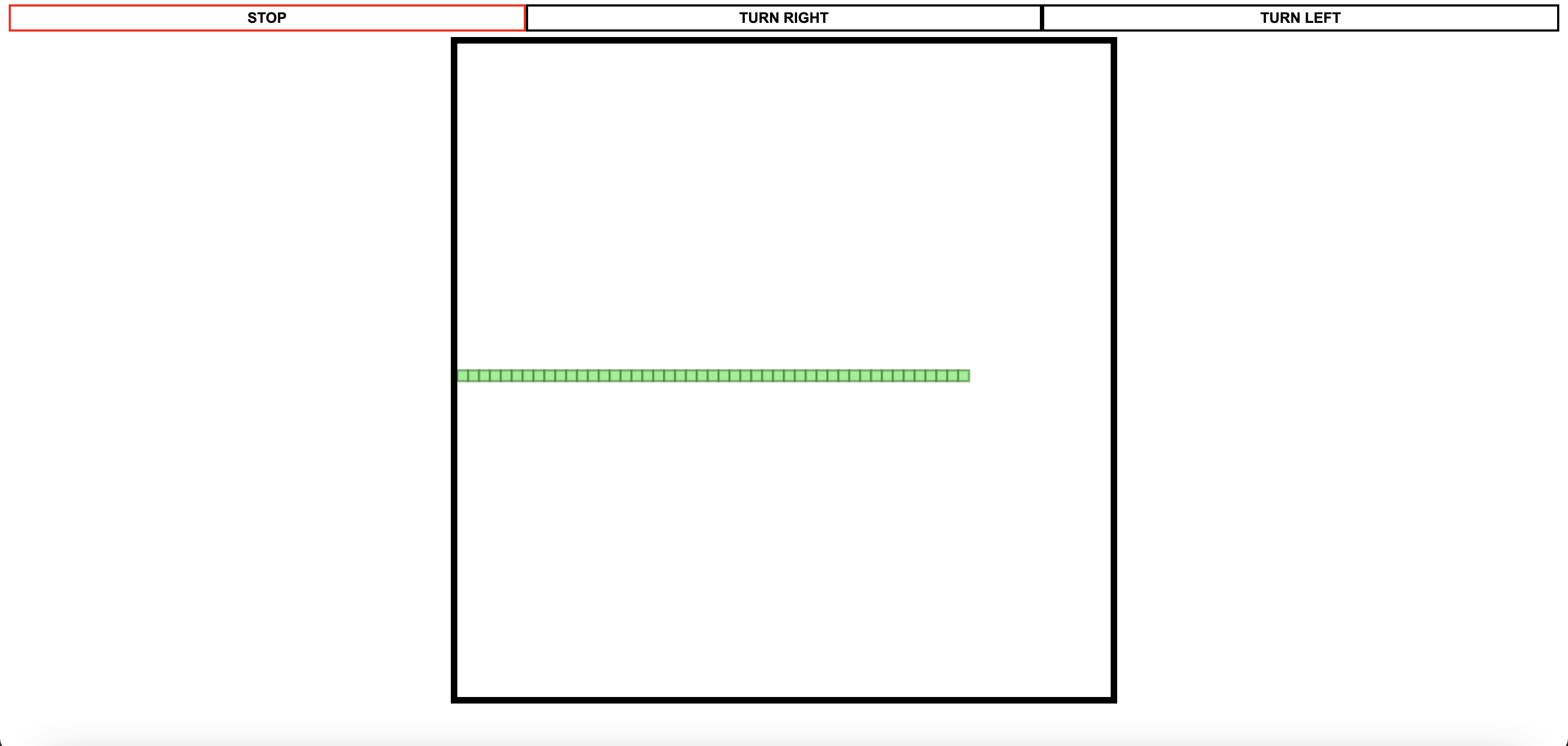
* + The game (and snake) will end when the snake touches the border of the canvas.



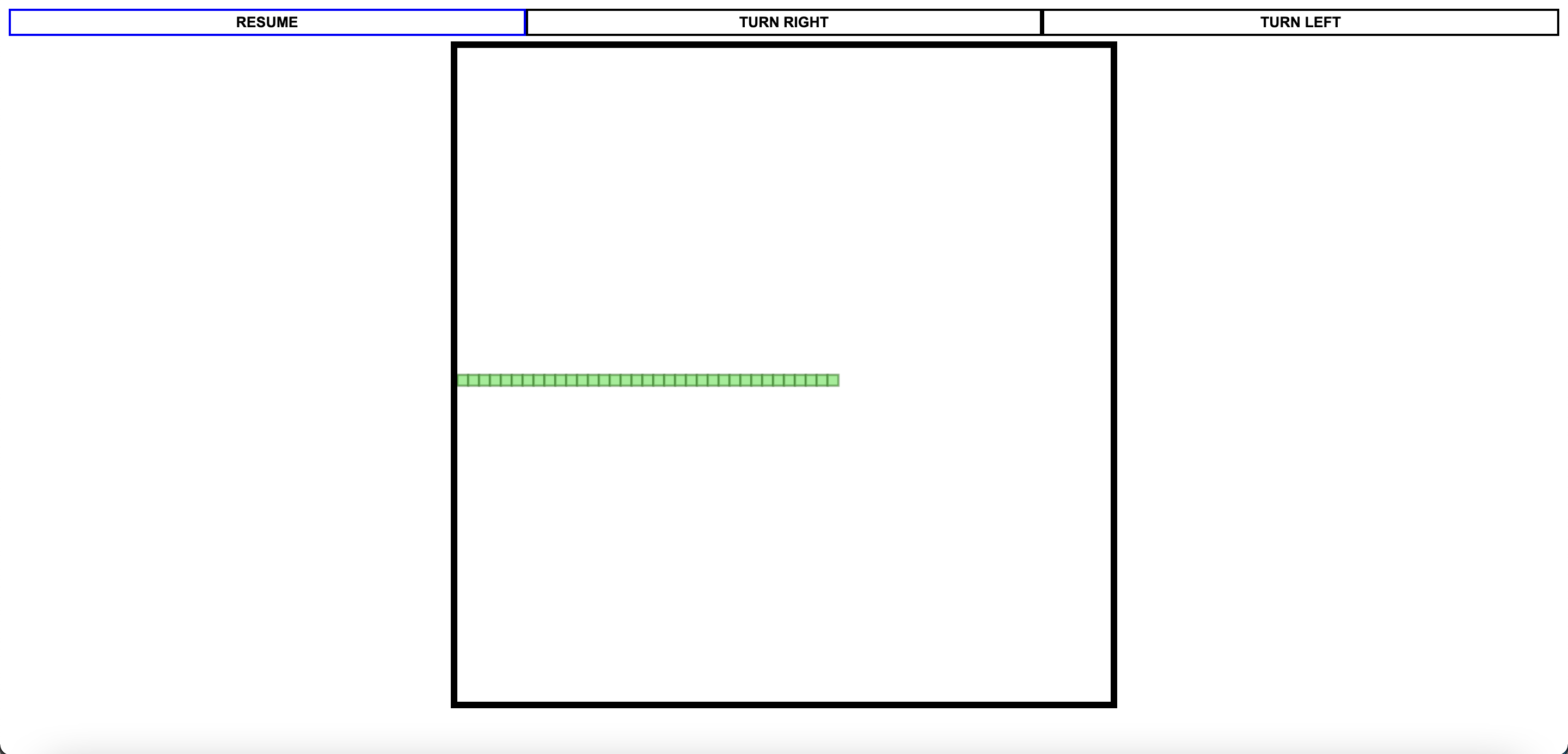
* + The game (and snake) will end when the snake’s head touches any part of its body.



* + The ‘START’ button will change to ‘STOP’ once pressed.



* + The game will pause when the user clicks the ‘STOP’ button.



* Task 2:
  + All the required output can be shown in a single screenshot.

