Nicolas De la Cruz
Homework 3
COM S 319

Homework 3 README

Task 1:

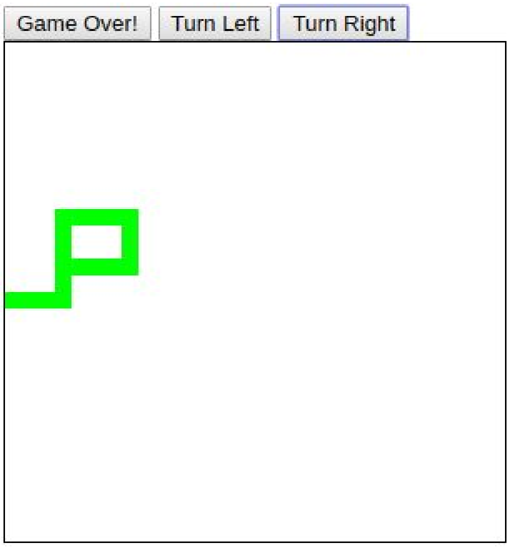
Compiling and Running instructions:

- Task 1 does not need compile
- To run the program you need to open snake.html file by double clicking the file in your file explorer then a new browsing window will open with the program.
- Hit the start button to start playing

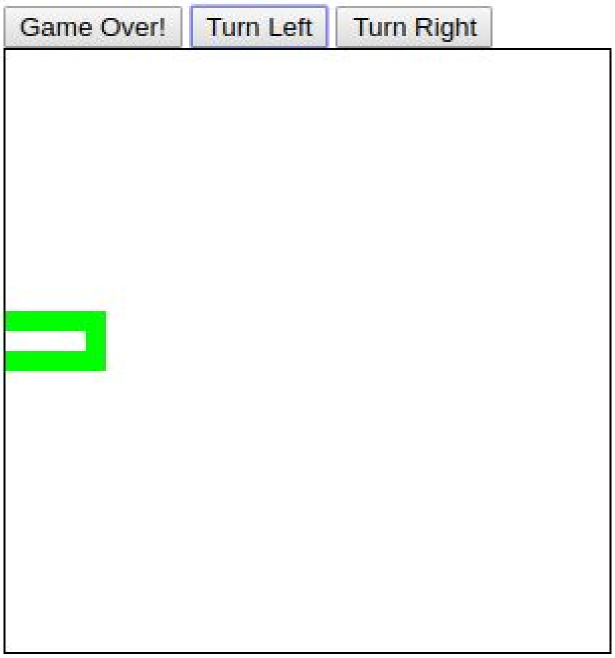
Solution Approach:

My solution approach for this task is as the following:

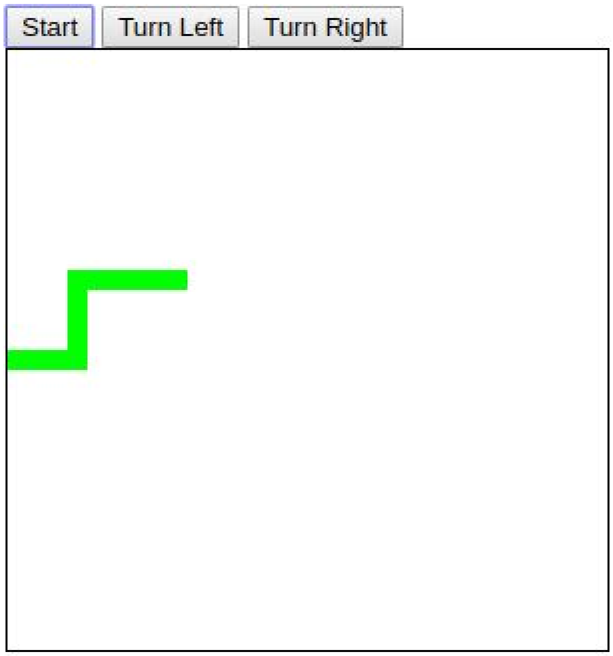
- I used a timer as given in the homework 3 spec sheet
- I have an array of coordinates to hold the snake's body
- I programmed the "start" button to enable the timer and set the direction to "right"
- Inside the timer I called two functions that handled the following:
    - Fill the snake, so to draw the rectangles onto the canvas for each of the snake body elements
    - Move the snake, to increment the next move for the snake in the current direction
        - The movement of the snake incremented by a fixed amount in the current direction and calls for boundary checks as well as if the snake is going to overlap its body
- The left and right movement buttons are programmed so that it checks the current direction and based on the turn (left or right) to update the direction in regards to the head of the snake
- If the snake overlaps it's body or moves out of bounds the game is over
- Another functionality implemented is that if the user starts the game then presses stop the program disables the timer until start is pressed again

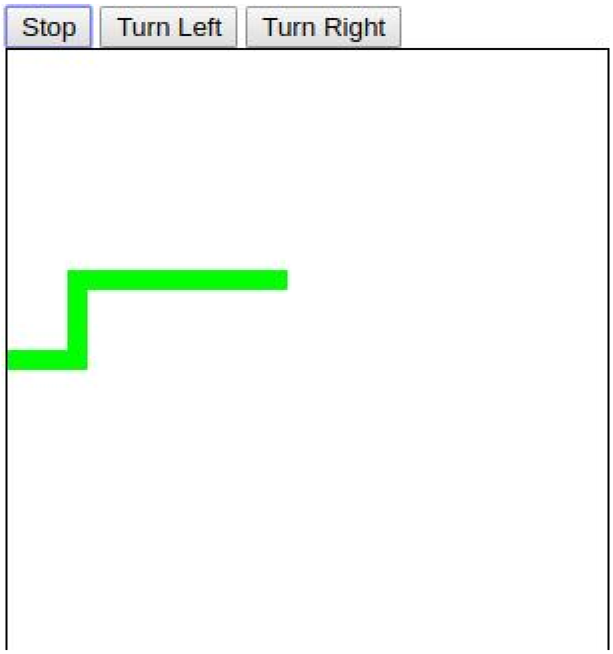


User overlaps snakes body



User moves snake out of bounds





User pauses game, notice start button text is set to “start” and in the second image the button is set to “stop”

Task 2:

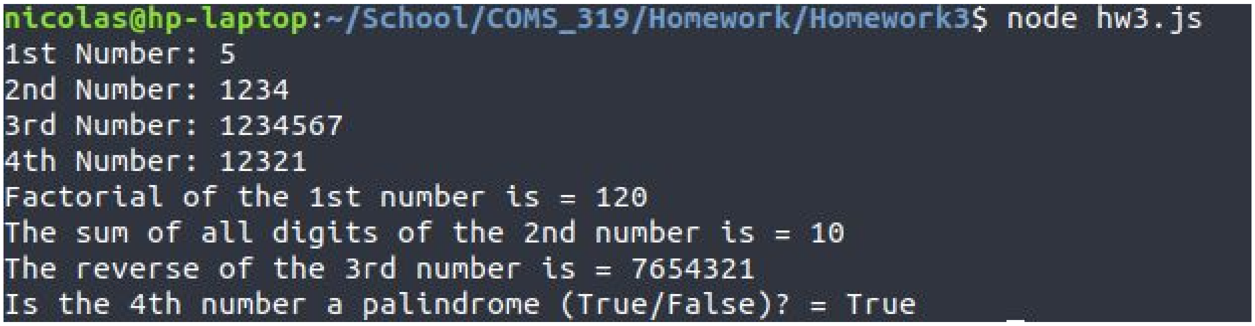
Compiling and Running instructions:

- Install ‘readline-sync’ like here
- Open terminal and change directory to the directory with the downloaded project
- Step into the project
- Enter the command in the terminal: “node hw3.js”
- Complete the prompt by entering four numbers, one at a time

Solution Approach:

My solution approach for this task is as the following:

- I used a similar approach to what was given in example.js
- I require user to have readline-sync
- I saved the four numbers asked in prompt using rs.question
- Then I made four more variables to store the result for each process done to each number
- To get the result for each number I created a function explained:
  - fact: is a recursive function to get a factorial of a given number
  - sumOfAllDigits: is a function that takes a number and converts the number to string then iterate through each digit of the string and get a cumulative sum of each of the digits
  - reverseOfNumber: is a function that takes a number and converts the number into a string then using the built in split, reverse and join functions we reverse the string as a result
  - palindromeCheck: is a function to check if the given number is a palindrome or not. The way this is done is by converting the given number to a string and using a different variable we get the reverse of the given number using the reverseOfNumber function. Lastly we compare if both string and reverse of the string are equal, if so then return true, else return false;
- Lastly we just do console log with the results of each number



```
nicolas@hp-laptop:~/School/COMS_319/Homework/Homework3$ node hw3.js
1st Number: 5
2nd Number: 1234
3rd Number: 1234567
4th Number: 12321
Factorial of the 1st number is = 120
The sum of all digits of the 2nd number is = 10
The reverse of the 3rd number is = 7654321
Is the 4th number a palindrome (True/False)? = True
```