COMS 319: Construction of User Interfaces

Akmal Fahmi Mohamad Rafie

Homework 3 Report

For Task 1, snake.html is a simple html file with three buttons, a canvas, and a script tag to snake.js. The canvas width, height, and color can be adjusted in this file.

The snake.js file contains functions, and fields required to perform the required functions. Lines 1-2 finds the canvas element, and creates a drawing object, ctx. Then, we created a 2d array, map, in order to store invalid locations that the snake can move to. These include the edges of the rectangle and also the snake's body. We initialize the map with 0 to indicate that the location is traversable, and 1 to stop the snake.

We also created enumerators to store the snake's current direction of movement, and initialize the initial direction as right. The snake's initial position is set to x = 1, and y = middle of the canvas by height.

The startStop() function is linked to the first button, startStop in the html file, and can be toggled to start and stop the snake. Clicking the "Start" button starts the interval timer and runs the internal function runner(). runner() checks the current direction of the snake, and compares the current position of the snake with the map before calling up(), down(), left(), or right(). If the snake is at an invalid position, the runner() stops the snake, and disables the snake's future movements. Otherwise, the user can toggle the button to stop and start the snake.

The turnLeft() and turnRight() functions check the current direction of the snake and switch it using the enumerators discussed above.

The functions up(), down(), left(), right() create a single dot in the canvas using the current x and y coordinates according to the direction of the snake. Then they will increment x or y. These functions use beginPath(), moveTo(x, y), lineTo(), and stroke() functions obtained from the canvas object.

For Task 2, the readline-sync Node module is installed alongside the source file as demonstrated in the sample code. This allows the program to easily store terminal inputs into 4 variables. But first, we have to make sure that the inputs are integers, therefore we use parseInt() before storing them in the variables.

1. Calculating the factorial for the first number: A factorial is the sum of all decreasing numbers from the initial number to 1; therefore we first decrement the input and store the input in the result1 variable, and loop it until it reaches 1. In every loop, we multiply result1 by the decreasing number.

2. Calculate the sum of all the digits of the second number: There are various methods to accomplish this, and one of them is to convert the int to string, so that we can access it as arrays. Therefore, allowing us to use a for loop to get the value of each digit of the number. Then, we can easily add the sum of all the digits, but the digit must be converted back to int by using Number() method.

3. For the third number given as an input, show the reversed number as an output: Similarly, we convert the int to string, so that we can access the number as an array. Then, we use a for loop that starts from the back of the number to store every digit - beginning with an empty string.

4. For the fourth number given as an input, check whether that number is a Palindrome: The first step is to convert the int to string in order to access the input as a variable. Then, we use a for loop with 3 indexes: the first index i, starts from the beginning, the second index j, starts from the back, and the main index (index) starts from 0. In every loop, we check if the i and j match, and move towards the middle by incrementing i, and decrementing j. At the same time, we increment the main index until we reach the middle of the number, which will end the loop.

```
PS C:\Users\fahmi\Desktop\coms319\Homeworks\HW3\src\task2> 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
```