East West University
Department of Computer Science and Engineering
Course: CSE479- Web Programming (Section 4)
Spring 2024

Lab Work -04

Objectives:
You will implement the N-Squared puzzle in JavaScript. This will be a group assignment.

Introduction

This assignment is about implementing JavaScript functions to solve the N-Squared Puzzle.

Puzzle Description

Originally called the **15-puzzle** (also called **Gem Puzzle**, **Boss Puzzle**, **Game of Fifteen**, **Mystic Square** and many other names), this game is a sliding puzzle that consists of a frame of numbered square tiles in random order with one tile missing. The puzzle also exists in other sizes, particularly the smaller 8-puzzle.

If the size is **3×3** tiles, the puzzle is called the **8-puzzle** or **9-puzzle**, and if **4×4** tiles, the puzzle is called the **15-puzzle** or **16-puzzle** named, respectively, for the number of tiles and the number of spaces. See https://en.wikipedia.org/wiki/15_puzzle for more information on the 15 puzzle.

Since the number of spaces for tiles in a puzzle is **N * N** (where N=3, 4, 5, 6, etc.) we are going to call it the **N-Squared** puzzle.

The object of the puzzle is to place the tiles in order (see diagram below) by making sliding moves that use the empty space.

Clicking on the closest tile in the **same row or same column** of the empty space moves that tile to the empty space. The new empty space becomes the previous position of the moved tile. **NOTE: tiles can only be moved vertically or horizontally, NOT diagonally.**

Implementation Details:

All the work for this assignment must be done in the **nSquaredPuzzle** project.

You can make the front-end, i.e., HTML and CSS code as you wish.

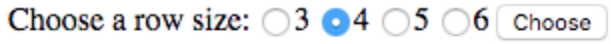
Use **JavaScript** to

- add the tiles for a new puzzle,
- respond to user interactions,
- move tiles legally, and
- notify the user when the puzzle is solved.

Implement as many JavaScript functions as needed to solve this problem.

Guidelines and appearance

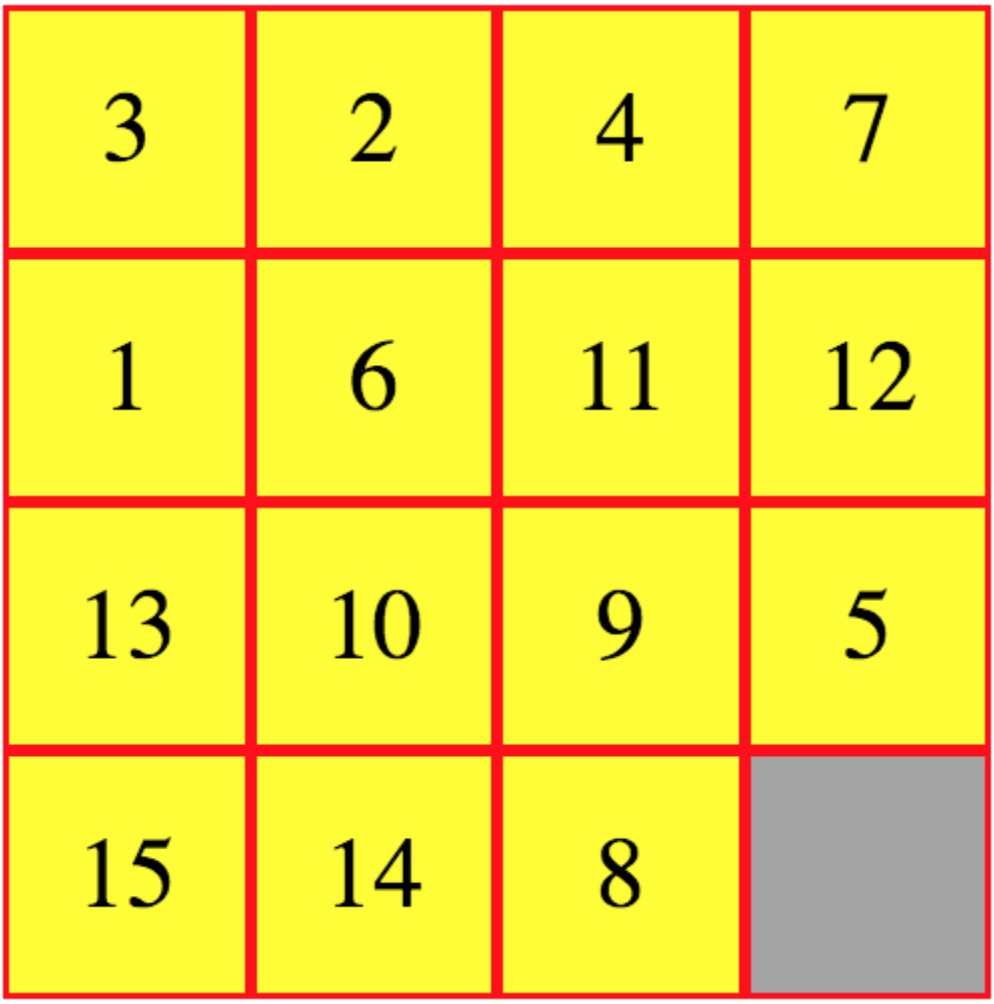
When you access the web page from which the puzzle is to be solved, the figure below depicts what you will see.



The default row size should be 4. Clicking in a different checkbox will change the row size to the indicated value. When the Choose button is clicked by the user, a squared frame of numbered squares with numbers and the empty space in random positions in the frame should be displayed. The figure below shows a sample of what the user should see.

Note: **you should start with a board in the win state**, **and shuffle it by calling your *moveTile()* function**.

All the best! 😊



The board size (a.k.a frame size), border width, & font size are all given in the JavaScript tile.

Each tile should be a square and its size should be determined by your script. You should create a span element for each tile and add it to the list of nodes that are children of the div element with **id == "board"**.

When the user clicks on a tile or the **Choose** button described above, a click event is triggered. You can register an event handler to respond to a click event as follows:

```
document.getElementById("sizeChoice").onclick = function() { … };
```

OR

```
const elClose = document.getElementById('close');   // Get the close button
elClose.addEventListener('click', dismissNote, false);// Click close-clear note
```

Where **dismissNote** is the function called in response to the click event triggered on the DOM element with **id == “close”**.

Code to exchange the locations of two elements in the DOM is provided in the JavaScript file.
