

# Web Development – Mr. Turner

## Project – Picture Puzzle

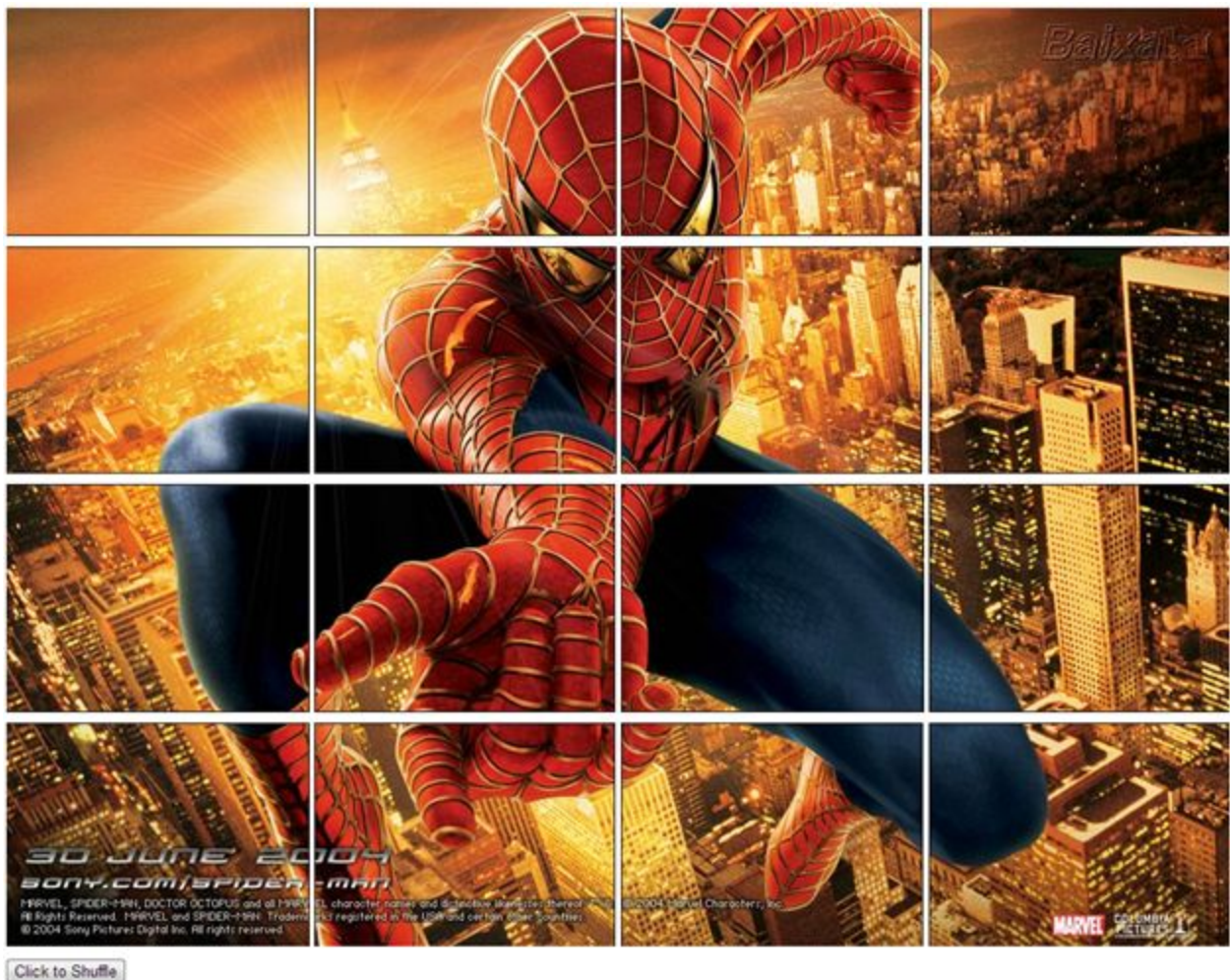
### Project Overview

Everyone likes doing a puzzle. Your page will allow users to mix up the pieces of a puzzle and then click them back into place.

### Display

The list below includes the essential elements of the page.

- Take any image and break it up into 16 identical rectangles. You will need to use a program like photoshop or paint to do this.
- Add a *Click to Shuffle* button.



## **Functionality**

When the page loads, the puzzle should be solved so that the user can have a look at it.

When the user clicks the shuffle button, move the pieces into random positions.

The puzzle operates on a click interface. When the user clicks on the first puzzle piece, that piece is considered selected. A selected piece should have some visual clue that it has been selected. You can use a border or opacity to make it look different from the others.

When the user clicks on the second puzzle piece, it should switch places with the selected puzzle piece. Afterwards there should be no selected piece and the display should return to normal.

The user continues switching pieces until the puzzle is solved. When all of the pieces are in their original positions, the page should display a congratulatory message.

## **Enhancements**

- Keep score. After the shuffle, how many clicks does it take for the user to solve the puzzle. Make sure to display the score as it happens on the page.
- Create a homepage for your site. The homepage will give the user the option of choosing from several different puzzles. Each of these options can link to a different page with a different puzzle. Many of the Javascript functions and all of the CSS can be stored in common files.

## **Necessary Programming Skills**

- Comprehension of the specifications sheet.
- Design Document
  - Figure out the information you need to keep track of.
    - This information will become your global variables.
  - Plan out the individual tasks your program must perform.
    - Think through the steps for each task.
    - Think through the information your task needs (where does it come from?).
    - These will become your functions.
  - Plan out the user interface.
    - You can start with the barest interface, but you should have an idea what you want the final product to look like.
- Managing your variables
  - What's global, local, and passed through as parameters (hint - this program can use all three)?
  - Are you making groups of variables into arrays?
  - Do you have a complete back end design (variables and functions that work the program)?
  - Does your back end inform your display?

- Sequencing
  - Does your program sort out the different tasks into their own functions?
  - Does your program sequence from the user interaction into the necessary functions?
  - Is there an efficiency to your code that flows from the design document?
- An intuitive user experience
  - Is your display appropriate to the program (what's viewable and what scrolling has to be done)?
  - Is your display adaptable to other resolutions?
  - Is the interface intuitive?