

coderdojowaterford / jquery_puzzle

Branch: master jquery_puzzle / README.md

Find file Copy path

audaxland the tutorial

09ad49d just now

1 contributor

189 lines (149 sloc) 5.44 KB

A jQuery puzzle

1) download and unzip the starter file from GitHub at

https://github.com/coderdojowaterford/jquery_puzzle

2) We'll be focusing on three files

index.html style.css index.html

Open these three files in a code editor such as notepad++ or sublime text and give a look at what is in it

open index.html in chrome

3) Javascript array

an array is a variable that contains multiple values

There are two syntaxes to create and use an array in javascript, those are the array syntax and the object syntax

The array syntax looks like `var my_array = ['value_one', 'second_value', 1, true]; my_array[2] = 'change'; alert(my_array[1]);`

The object syntax looks like `var my_array = { key1: 'value 1', key_other: 5, apple: 'green'}; my_array.key1 = 'new value'; alert(my_array.apple);`

Go in the script.js file and create in the `$(document).ready(function () { }` section an array that will contain the id's and position of the puzzle pieces like

```
// store the positions of the pieces in an array
var pieces = {
  0: {id: "piece_1_1", initial_x: 0, initial_y: 0, x: 0, y: 0 },
  1: {id: "piece_2_1", initial_x: 1, initial_y: 0, x: 1, y: 0 },
  2: {id: "piece_3_1", initial_x: 2, initial_y: 0, x: 2, y: 0 },
  3: {id: "piece_1_2", initial_x: 0, initial_y: 1, x: 0, y: 1 },
  4: {id: "piece_2_2", initial_x: 1, initial_y: 1, x: 1, y: 1 },
  5: {id: "piece_3_2", initial_x: 2, initial_y: 1, x: 2, y: 1 },
  6: {id: "piece_1_3", initial_x: 0, initial_y: 2, x: 0, y: 2 },
  7: {id: "piece_2_3", initial_x: 1, initial_y: 2, x: 1, y: 2 },
}
```

then create an array for the empty spot in the puzzle like

```
// store the position of the empty piece
var empty_piece = {x: 2, y: 2};
```

4) Javascript functions to move the pieces

A function is a piece of code that you can call directly elsewhere in your code, a function looks like

```
function my_function_name( variable1, variable2) {
```

```
    // do stuff here
    return 'a result';
}
```

Still, in the script.js file, create a function that moves a piece of the puzzle from one position to another like the following

```
// this is a function that places a piece in a chosen position
function place_piece( index, x, y ) {
    pieces[index].x = x;
    pieces[index].y = y;
    $('#'+ pieces[index].id).css('left', (x * 33.5) + '%');
    $('#'+ pieces[index].id).css('top', (y * 33.5) + '%');
}
```

now make another function that makes a move of the game using the previous function like

```
// try to move a piece if possible
function move_piece( index ) {
    var current_x = pieces[index].x;
    var current_y = pieces[index].y;
    // move this piece to the empty spot if possible
    if (Math.abs( current_x - empty_piece.x) + Math.abs( current_y - empty_piece.y) == 1 ){
        place_piece( index, empty_piece.x, empty_piece.y );
        empty_piece.x = current_x;
        empty_piece.y = current_y;
    }
}
```

5) jQuery click event on a piece of the puzzle

now use the .click() method of jQuery to associate an event on pieces of the puzzle like

```
// what to do when a piece is clicked
$('.puzzle-piece').click(function () {
    // find the corresponding piece
    for( var k = 0 ; k < 8 ; k++ ) {
        if ( this.id == pieces[k].id ){
            move_piece( k );
        }
    }
});
```

pay a little bit in the browser to see what it is doing and make sure you are understanding how it is working

6) add the play button

the play button is already in your index.html file but doesn't do anything, so create a jQuery click event on that button to make it mix the pieces of the puzzle and start the game, this looks like

```
// mix the pieces when playing
$('#play').click(function () {
    for ( var k = 0 ; k < 1000 ; k++ ) {
        var random_index = Math.floor(Math.random()*(7.99));
        move_piece( random_index );
    }
});
```

try it in the browser and make sure it works

7) Create a function that detects when you win

Now we need to make a function that can know if it has won, this looks like

```
//detect if the game is won
```

```
function detect_win() {
  var win = true;
  for( k = 0 ; k < 8 ; k++ ) {
    if ( (pieces[k].initial_x != pieces[k].x ) || (pieces[k].initial_y != pieces[k].y ) ) {
      win = false;
    }
  }
  if (win) {
    $('.fixed-piece').css('opacity', 1);
    $('#win').show();
    $('#play').show();
    empty_piece.x = 10;
    empty_piece.y = 10;
  }
}
```

Now we need to use that function, and to do so we need to change the event we have for when we click on a piece of the puzzle, this will now look like

```
// what to do when a piece is clicked
$('.puzzle-piece').click(function () {
  // find the corresponding piece
  for( var k = 0 ; k < 8 ; k++ ) {
    if ( this.id == pieces[k].id ){
      move_piece( k );
    }
  }
  detect_win();
});
```

8) correct the bugs

What we have done above has created unwanted behaviours, try to figure out what is going on.

once you think you understand, change the event that happens when we click on the play button to look like the following

```
// mix the pieces when playing
$('#play').click(function () {
  empty_piece.x = 2;
  empty_piece.y = 2;
  for ( var k = 0 ; k < 1000 ; k++ ) {
    var random_index = Math.floor(Math.random()*(7.99));
    move_piece( random_index );
  }
  $('#play').hide();
  $('#win').hide();
  $('.fixed-piece').css('opacity', 0.2);
});
```

and now check in the browser that it is all working better