

# **Information Infrastructure II**

**INFO I211 – Spring 2014 – Sections 18530 & 22719**

***Lecture 23 – 2014.04.16 & 2014.04.17***

**Instructor:**

**Mitja Hmeljak,**

**<http://mypage.iu.edu/~mitja>**

**[mitja@indiana.edu](mailto:mitja@indiana.edu)**

# a Javascript + HTML game: "runner" against "chaser"

game objective:

move the "chaser" until you catch the "runner"

Clicks: 10



# implementing a playing field: its *view*

the game playing field will look like this:

- a table with H (height) rows and W (width) columns
- how to specify the size of each table row/column in HTML :

```
<table border="1" id="playfield">  
  <tr height="30">  
    <td width="30"></td>
```

# implementing a playing field: its *model*

- the content of a table can be stored in a 2D array in Javascript.
- Does Javascript have 2D arrays?

- not quite... but we can **build**  
an *array of arrays*

```
function MyArray2D (pWidth, pHeight) {  
  var lArray = new Array(pWidth);  
  for (var j=0; j < pWidth; j++) {  
    lArray[j] = new Array(pHeight);  
  }  
  return lArray;  
}
```

- then instantiate  
our "2D" array  
with a function call

```
var myArray = MyArray2D(2, 2)
```

## implementing a playing field: its *model*

- the content of an HTML table stored in a 2D array
- can then be initialized like this:

```
for (var j = 0; j < 7; j = j + 1) {  
    for (var i = 0; i < 5; i++) {  
        myArray[i][j] = 0;  
    }  
}
```

# playing field: what are its dimensions?

- find out an HTML table's dimensions from Javascript:
  - theTable.rows = all rows in an HTML table
  - theTable.rows.length = how many rows?
  - theTable.rows[i] = the  $i^{\text{th}}$  row in an HTML table
  - theTable.rows[i].cells = all cells in a row
  - theTable.rows[i].cells.length = how many cells in this row?
  - theTable.rows[i].cells[j] = the  $j^{\text{th}}$  cell in the  $i^{\text{th}}$  row

```
var lTable = document.getElementById("playfield");

var lTableHeight = lTable.rows.length;
var lTableWidth = 0;

for (var j = 0; j < lTableHeight; j = j + 1) {
    lTableWidth = lTable.rows[j].cells.length;
}
```

# group task 1

- download  
I211-lecture-23-task-1-starting-text.html  
from Oncourse → Resources → SampleCode
- modify the `run()` function so that it displays  
the number of **clicks** from *any of the 5 buttons*  
as text in the HTML page
- the initial "*Clicks: 0.*" text needs to be updated  
*every time* one of the 5 buttons is clicked, to show:  
"*Clicks: 1.*", "*Clicks: 2.*", "*Clicks: 3.*", etc.

# group task 2

- (continue working on the file  
I211-lecture-23-task-1-starting-text.html  
from Oncourse→Resources→SampleCode)

1. modify the `<table>` element  
to always display 30x30-pixel cells
2. modify the `run()` function to display  
in each cell of the table the value  
 $(i*j)+k$  , where:
  - $k$  = number of clicks on *any of buttons*
  - $i$  and  $j$  = *coordinates* of the table cell

*hint: use a Javascript array to keep track of each cell's value*

The image shows three overlapping 5x5 grids representing a 30x30 table at different click counts (k). The top grid (k=0) has values from 1 to 25. The middle grid (k=1) has values from 9 to 29. The bottom grid (k=2) has values from 17 to 33. The grids are offset to show the progression of values.

1	1	1	1	1
1	2	3	4	5
				9
9	9	9	9	9
9	10	11	12	13
9	11	13	15	17
9	12	15	18	21
9	13	17	21	25
9	14	19	24	29
9	15	21	27	33