

# **Information Infrastructure II**

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

***Lecture 22 – 2014.04.07 & 2014.04.08***

**Instructor:**

**Mitja Hmeljak,**

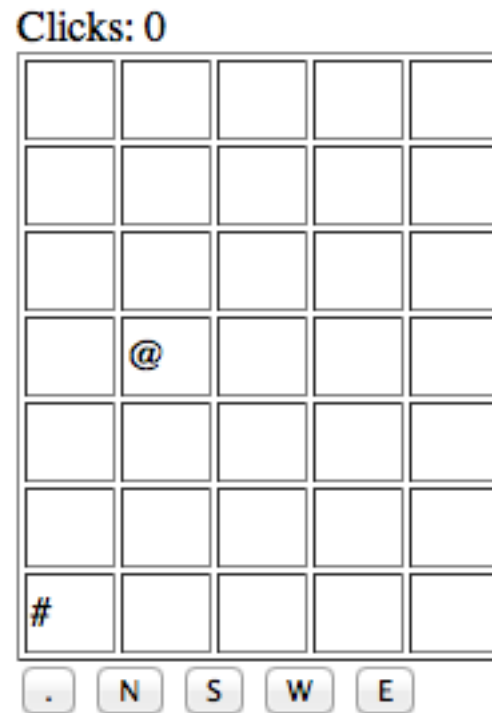
**<http://mypage.iu.edu/~mitja>**

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

# Combining Events, Input & innerHTML...

... to build a JavaScript+HTML game:

"runner" (@)  
against  
"chaser" (#)

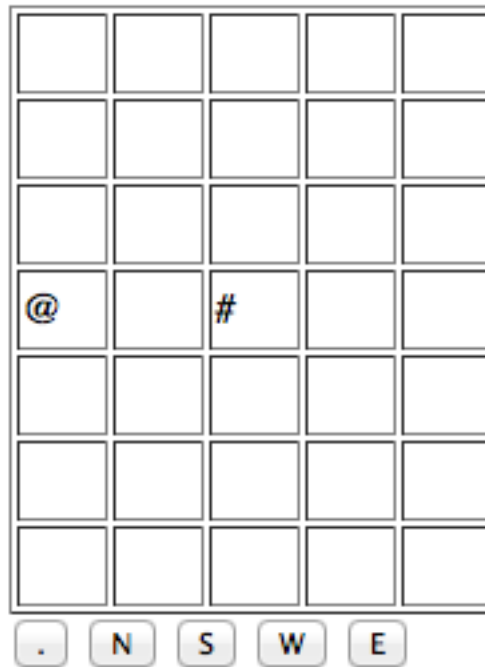


# a Javascript + HTML game

game controls – user input – 5 buttons:

- move the "chaser":
  - North (N)
  - South (S)
  - West (W)
  - East (E)
- don't move (.)

Clicks: 7



# game: "runner" against "chaser"

game objective:

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

Clicks: 10



# a Javascript + HTML game: what's needed?

the game allows for two characters:

- "runner" (@) – moves automatically
- "chaser" (#) – user-controlled

the game needs a playing field:

- a table with H (height) rows and W (width) columns

the game provides user controls:

- buttons

# game characters: moving the "runner"

moving the two characters:

- "runner" (@)

movement:

- random movement... how?

```
function MyRandomDirection() {  
    // a float between 0.0 and 1.0  
    lRandomFloat = Math.random();  
    if (lRandomFloat < 0.25) {  
        return "N";  
    } else if  
        ....  
}
```

- use the Javascript function Math.random()
  - it returns a random floating-point value between 0.0 and 1.0

# game characters: moving the "chaser"

moving the two characters:

- "chaser" (#)

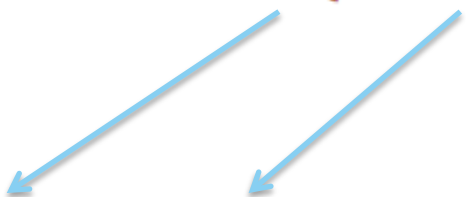
user-controlled movement:

- input buttons... how? On the HTML side:

```
<input type="button" value="N" id="N" onClick="Run(this.id);">
```

- on the Javascript side:

```
function Run(pClickedID) {  
    // move chaser:  
    if (pClickedID=="N") {
```



# 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

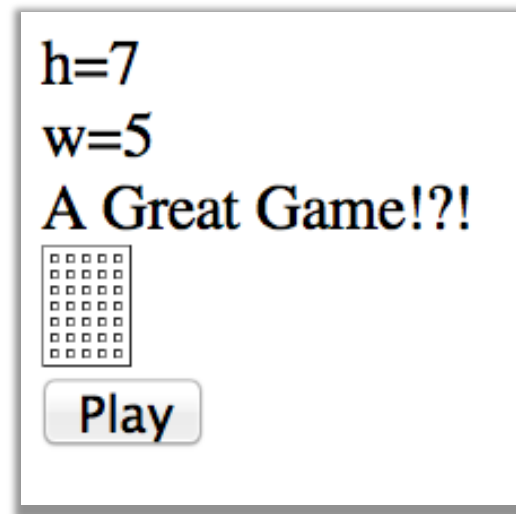
```
var lTable = document.getElementById("playfield");

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  
I2I1-lecture-22-task-1-starting-text.html  
from Oncourse → Resources → SampleCode
- modify the `run()` function so that it displays  
the number of rows and columns in the table
  - as text in the HTML page
  - instead of "H" and "W"
  - when you click "Play"



# group task 1: solution

```
function run() {  
  
    // 1) get the number of rows in the HTML table and "print it out":  
    // find the "playfield" table element in the HTML document:  
    var lTable = document.getElementById("playfield");  
    // then obtain the length of the list of rows in the table element:  
    var lTableHeight = lTable.rows.length;  
  
    // to display the obtained value, you can ...  
    // ...create a variable in your Javascript function...  
    var myLabelH = document.getElementById("hLabel")  
    // ...to refer to the innerHTML of the element in the HTML document...  
    // ...and update the innerHTML's value:  
    myLabelH.innerHTML = "h="+lTableHeight;  
  
    // 2) get the number of cells in one of the HTML table's rows:  
    var lTableWidth = lTable.rows[2].cells.length;  
    // to display the obtained value, you can also...  
    // ...directly update the innerHTML value of an HTML element:  
    document.getElementById("wLabel").innerHTML = "w="+lTableWidth;  
}
```