**HW6: RE\_PHP *–* Introducing Dynamic Features into our RE Page Design**

**Due: Feb 23**

At this point we should have a good idea of the basic framework of our main RE page, and have made some initial decisions regarding implementation of some of the data structures underlying the page-display. We should have looked at the structure of the HTML underlying the main page and found some opportunities for auto-generating portions of it in PHP using loops.

In today's lab we would like to accomplish several things:

1. Represent the state of our boards (i.e where the tiles, shrimp and holes are) in a data structure *bord* that can be scanned in order to generate and place web elements using SPAN tags
2. Write another function *ShowBoards()* that scans through *bord* and generates all needed lines of HTML according to the values within it

* Use this function to replace the HTML implementing starting-board-state in your prettified HTML file. SAVE this file AS a PHP file (e.g. hw6.php)
* Example: see lines 628-695 in the source of Prof. Neville's <http://cs.bemidjistate.edu/fneville/RE/rehack.htm> -- replace (most of) these with your function!
* A screenshot of the beginning of this code is shown in Figure 1 below

1. Write a function *InitBoards* that initializes *bord* to the start-game state

* Find a way to have this function called when someone clicks on the [Start your turn over] action, so the board resets to its start-state

1. Further demonstrate this data structure by correctly displaying valid location-links for the placement of a new, red shrimp

* Implement these links such that upon clicking on a single one, a new shrimp symbol is added to *bord*, before displaying the boards showing their new state with the new shrimp placed.
* Example: <http://cs.bemidjistate.edu/fneville/RE/rehack.php> (click Action 4)

1. Finally, write a report in GoogleDocs discussing your implementation of the four points above, showing the text of your functions and describe their placement within your php file. Add the entirety of the PHP file's text as an appendix to your report, for reference.
   1. Provide an explanation for each of your four solutions, using code-snippets as necessary
   2. Formatting code: your functions and your Appendix should be in Courier font, to distinguish your functions (and your hw6.php appendix!) from your own words in the report, which should remain in the default (non-monospace) font
   3. Copy hw6.php to your TurnIn folder
   4. Create a link to your www's hw6.php file from your hw.htm, so we can see your code in action.
   5. Also create a link to your report from hw.htm.

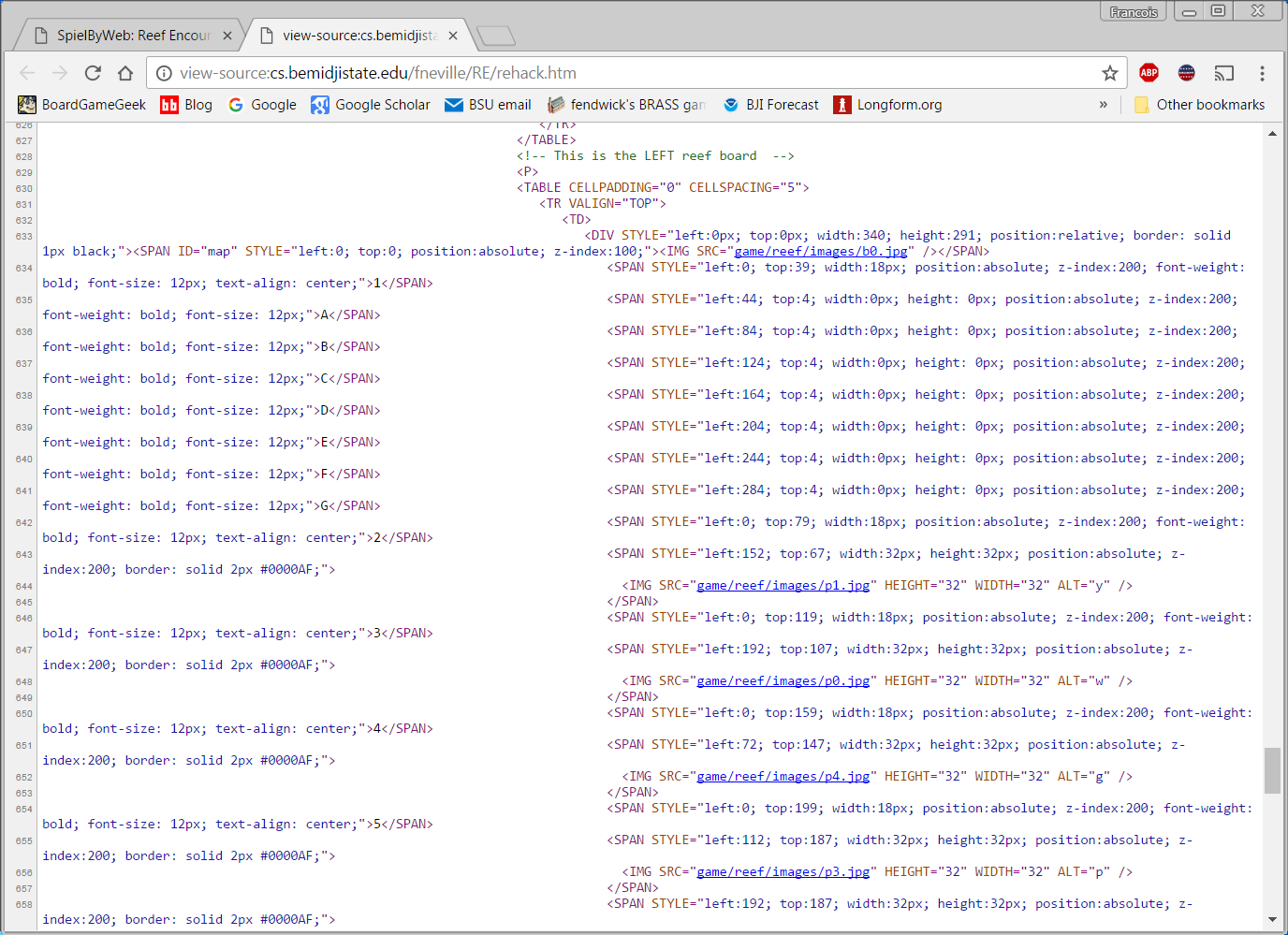


Figure 1: Example screengrab for #2 above

Hints:

Refer to Prof Neville's page for the following examples: <http://cs.bemidjistate.edu/fneville/RE/rehack.htm>

I created the following constant to append to the end of each of my PHP-generated HTML lines:

define("kEOL", chr(13).chr(10)); (Windows version)

define("kEOL", chr(10)); (Unix/Linux version)

Given the two reef-boards displayed in my 2P game, these are the arrays I defined:

$rf0 = array(0=>'x','x','x','','','','x','x','','','y','','','x','','','x','','w','','','','g','','\_','','x','','','','p','x','o','','','','','','','','','');

$rf3 = array(0=>'x','','','','','','x','','','','o','x','','','','','x','','g','','','','w','','\_','','','','x','','p','','y','','x','x','','','x','','','x');

$bord[0] = $rf0;

$bord[1] = $rf3;

Here's a function I wrote to center **[+]** over valid reef locations on the Left board (board #0)

function foob0() {

$b0 = array(1=>3,4,5,8,9,10,11,12,14,15,17,18,19,20,21,22,23,24,25,27,28,29,30,32,33,34,35,36,37,38,39,40,41);

**foreach ($b0 As $el) {**

echo "<span style='left:" . targcel2left($el) . "; top:" . targcel2top($el) . "; width:0px; height: 0px; position:absolute; z-index:501;'> <a href='rehack.php?act=placeshrimp&b=0&cell=" . $el . "'>[+]</a></span>" . kPHPEOL;

**}**

}

targcel2left() and targcel2top(), respectively, translate the cell number to the appropriate Left and Top values to center the **[+]** target

Be warned: these are not necessarily \*good\* implementation-choices; this is merely what \*I\* chose to do… at the last minute, about 20secs before AdvWeb class!