|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>A Pretty Strange Game</title> |
|  | <style> |
|  | td.circle:before { |
|  | content: attr(marker); |
|  | display: block; |
|  | background: #f00; |
|  | width: 10px; |
|  | height: 10px; |
|  | line-height:10px; |
|  | text-align:center; |
|  | vertical-align: middle; |
|  | border-radius: 50%; |
|  | margin:0 auto; |
|  | } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <h1>A Pretty Strange Game</h1> |
|  | <!-- div tag for the building the grid --> |
|  | <div id="divTable"> |
|  | <!-- javascript section for grid --> |
|  | <script> |
|  | var divTable = document.getElementById('divTable'); |
|  | var table\_structure; |
|  | var i, j; |
|  | table\_structure = '<table border="1" width="100%">'; |
|  | for(i = 0; i < 10; i++) { |
|  | table\_structure += '<tr>'; |
|  | for(j = 0; j < 10; j++) { |
|  | table\_structure += '<td height="25" width="25">'; |
|  | } |
|  | table\_structure += '</td>'; |
|  | } |
|  | table\_structure += '</table>'; |
|  | divTable.innerHTML = table\_structure; |
|  |  |
|  | //Move counter metrics: |
|  | var cell\_touch\_count = 0.0; |
|  | var max\_cell\_touches = 1000001; |
|  | /\* |
|  | missing metrics (as of 8/24/17 @ 7:52am): |
|  | - min. times a particular cell was touched |
|  | - max. times a particular cell was touched |
|  | - number of times each particular cell was touched |
|  | \*/ |
|  |  |
|  | //Game board position GLOBAL vars: |
|  | var original\_square\_pos = 90; |
|  | var previous\_square\_pos; |
|  | var new\_square\_pos; |
|  |  |
|  | var td = document.getElementsByTagName("td")[original\_square\_pos]; //get the 90th (bottom left corner) <td> |
|  | var att = document.createAttribute("class"); //creates a class attribute |
|  | att.value = "circle"; //sets the value of the class attribute |
|  | td.setAttributeNode(att); |
|  |  |
|  | //checkForWinner(position) called on each directional move: |
|  | function checkForWinnerOnBtnClick(position) { |
|  | if (position == 9) { |
|  | var button = document.getElementsByTagName("button")[0]; |
|  | button.hidden = true; |
|  | document.getElementById('game\_result').innerHTML = "You Win!!!" |
|  | console.log('DEBUG: ' + 'position parameter = ' + position + ' YOU WIN!') |
|  | } |
|  | } |
|  |  |
|  | //checkForMaxTouches(touch\_count) called on each directional move: |
|  | function checkForMaxTouchesOnBtnClick(touch\_count) { |
|  | if (touch\_count >= max\_cell\_touches) { |
|  | var button = document.getElementsByTagName("button")[0]; |
|  | button.hidden = true; |
|  | document.getElementById('game\_result').innerHTML = "1,000,001 Touches - Game Over!" |
|  | console.log('DEBUG: ' + 'touch\_count parameter = ' + touch\_count + ' YOU LOSE!') |
|  | } |
|  | } |
|  |  |
|  | //gameSimulator() called recursively when the sim game button is clicked: |
|  | function gameSimulator() { |
|  | while (new\_square\_pos != 9 || cell\_touch\_count > 1000000) { |
|  | moveFunction(); |
|  | } |
|  | var mv\_button = document.getElementsByTagName("button")[0]; |
|  | var sim\_button = document.getElementsByTagName("button")[1]; |
|  | mv\_button.hidden = true; |
|  | sim\_button.hidden = true; |
|  | if (new\_square\_pos == 9) { |
|  | document.getElementById('result\_stats').innerHTML = "GAME RESULT = Marker Finish: WINNER"; |
|  | document.getElementById('touch\_total').innerHTML = 'Total Cell Touches = ' + cell\_touch\_count; |
|  | document.getElementById('avg\_touches\_per\_cell').innerHTML = 'Avg. Touches Per Cell = ' + (cell\_touch\_count/100.0); |
|  | } else { |
|  | document.getElementById('result\_stats').innerHTML = "GAME RESULT = 1,000,001 Touches: LOSER"; |
|  | document.getElementById('touch\_total').innerHTML = 'Total Cell Touches = ' + cell\_touch\_count; |
|  | document.getElementById('avg\_touches\_per\_cell').innerHTML = 'Avg. Touches Per Cell = ' + (cell\_touch\_count/100.0); |
|  | } |
|  | console.log('DEBUG: ' + 'touch\_count parameter = ' + cell\_touch\_count + ' YOU LOSE!'); |
|  | } |
|  |  |
|  | //moveFunction() controls the movement logic of the marker on the board: |
|  | function moveFunction() { |
|  | //Place the marker in bottom-left corner to begin the game: |
|  | console.log('cell touch count = ' + cell\_touch\_count); |
|  | if (cell\_touch\_count <= 1) { |
|  | previous\_square\_pos = original\_square\_pos; //set this for the first turn |
|  | } else { |
|  |  |
|  | previous\_square\_pos = new\_square\_pos; |
|  | } |
|  | console.log('DEBUG: previous\_sqaure\_pos = ' + previous\_square\_pos); |
|  |  |
|  | console.log('DEBUG: moveFunction() executed') |
|  | //Randomly choose a direction: up, down, left, or right: |
|  | var D = Math.floor((Math.random() \* 4) + 1); |
|  | //Randomly choose a number of steps: 0, 1, or 2: |
|  | var N = Math.floor((Math.random() \* 3) + 0); |
|  |  |
|  | console.log('DEBUG: D= ' + D + ' and N= ' + N); |
|  |  |
|  | switch(D) { |
|  | case 1: |
|  | new\_square\_pos = previous\_square\_pos-(N\*10); |
|  | document.getElementsByTagName("td")[previous\_square\_pos].removeAttribute("class"); |
|  | console.log('DEBUG: new\_sqaure\_pos = ' + new\_square\_pos); |
|  |  |
|  | if (new\_square\_pos >= 0 && new\_square\_pos < 100) { |
|  | //count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | td = document.getElementsByTagName("td")[new\_square\_pos]; |
|  | att = document.createAttribute("class"); //creates a class attribute |
|  | att.value = "circle"; //sets the value of the class attribute |
|  | td.setAttributeNode(att); |
|  | } else { |
|  | //don't move |
|  | console.log('DEBUG: location ' + new\_square\_pos + ' out of bounds'); |
|  | new\_square\_pos = previous\_square\_pos; //reset so no move on o.o.b. |
|  | //count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | } |
|  | checkForMaxTouchesOnBtnClick(cell\_touch\_count); |
|  | checkForWinnerOnBtnClick(new\_square\_pos); |
|  | break; |
|  | case 2: |
|  | new\_square\_pos = previous\_square\_pos+(N\*10); |
|  | previous\_square\_pos = new\_square\_pos-(N\*10); |
|  | document.getElementsByTagName("td")[previous\_square\_pos].removeAttribute("class"); |
|  |  |
|  | if (new\_square\_pos >= 0 && new\_square\_pos < 100) { |
|  | //count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | td = document.getElementsByTagName("td")[new\_square\_pos]; |
|  | att = document.createAttribute("class"); //creates a class attribute |
|  | att.value = "circle"; //sets the value of the class attribute |
|  | td.setAttributeNode(att); |
|  | } else { |
|  | // don't move |
|  | console.log('DEBUG: location ' + new\_square\_pos + ' out of bounds'); |
|  | new\_square\_pos = previous\_square\_pos; //reset so no move on o.o.b. |
|  | // count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | } |
|  | checkForMaxTouchesOnBtnClick(cell\_touch\_count); |
|  | checkForWinnerOnBtnClick(new\_square\_pos); |
|  | break; |
|  | case 3: |
|  | new\_square\_pos = previous\_square\_pos-(N\*1); |
|  | previous\_square\_pos = new\_square\_pos+(N\*1); |
|  | document.getElementsByTagName("td")[previous\_square\_pos].removeAttribute("class"); |
|  | console.log('DEBUG: new\_sqaure\_pos = ' + new\_square\_pos); |
|  |  |
|  | if (new\_square\_pos >= 0 && new\_square\_pos < 100) { |
|  | //count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | td = document.getElementsByTagName("td")[new\_square\_pos]; |
|  | att = document.createAttribute("class"); //creates a class attribute |
|  | att.value = "circle"; //sets the value of the class attribute |
|  | td.setAttributeNode(att); |
|  | } else { |
|  | // don't move |
|  | console.log('DEBUG: location ' + new\_square\_pos + ' out of bounds'); |
|  | new\_square\_pos = previous\_square\_pos; //reset so no move on o.o.b. |
|  | // count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | } |
|  | checkForMaxTouchesOnBtnClick(cell\_touch\_count); |
|  | checkForWinnerOnBtnClick(new\_square\_pos); |
|  | break; |
|  | case 4: |
|  | new\_square\_pos = previous\_square\_pos+(N\*1); |
|  | previous\_square\_pos = new\_square\_pos-(N\*1); |
|  | document.getElementsByTagName("td")[previous\_square\_pos].removeAttribute("class"); |
|  | console.log('DEBUG: new\_sqaure\_pos = ' + new\_square\_pos); |
|  |  |
|  | if (new\_square\_pos >= 0 && new\_square\_pos < 100) { |
|  | //count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | td = document.getElementsByTagName("td")[new\_square\_pos]; |
|  | att = document.createAttribute("class"); //creates a class attribute |
|  | att.value = "circle"; //sets the value of the class attribute |
|  | td.setAttributeNode(att); |
|  | } else { |
|  | // don't move |
|  | console.log('DEBUG: location ' + new\_square\_pos + ' out of bounds'); |
|  | new\_square\_pos = previous\_square\_pos; //reset so no move on o.o.b. |
|  | // count square touch (add to million count for analytics) |
|  | cell\_touch\_count += 1; |
|  | } |
|  | checkForMaxTouchesOnBtnClick(cell\_touch\_count); |
|  | checkForWinnerOnBtnClick(new\_square\_pos); |
|  | break; |
|  | } |
|  | } |
|  | </script> |
|  | </div> |
|  | <!-- move button --> |
|  | <div> |
|  | <button onclick="moveFunction()">single move btn</button> |
|  | </div> |
|  | <!-- game simulator button --> |
|  | <div> |
|  | <button onclick="gameSimulator()">game simulator btn</button> |
|  | </div> |
|  | <!-- game result message --> |
|  | <div> |
|  | <h2 id="game\_result"></h2> |
|  | </div> |
|  | <!-- game stats --> |
|  | <div> |
|  | <h3 id="touch\_total"></h3> |
|  | <h3 id="avg\_touches\_per\_cell"></h3> |
|  | <h3 id="result\_stats"></h3> |
|  | </div> |
|  | </body> |
|  | </html> |
|  |  |