Министерство образования и науки Российской Федерации

Ульяновский Технический университет

Кафедра: вычислительная техника

**Дисциплина: Интернет программирование**

**Лабораторная работа.**

**«Игра магический квадрат на javascript»**

Выполнил:

Студент: ИВТАП Бд-11

Кондратьев Павел Сергеевич

Проверил:

Игонин Андрей Геннадьевич

Ульяновск, 2017

Содержание

|  |  |
| --- | --- |
| 1. Цели и задачи……………………………………………………………… | 3 |
| 2. Ход выполнения работы…………………………………………………. | 3 |
| 3. Выводы ……………………………………………………………………. | 16 |
| 4. Приложение ………………….…………………………………………… | 16 |
|  |  |

**Цели и задачи**

**Цель**

Развитие, расширение и понимание устройства Интернета, с помощью изучения языков программирования, которые используются в создании сайта. Для разработки использовались такие языки, как HTML, CSS, JavaScript, PHP, SQL, где во Front-end разработке (клиентская часть) использовались jQuery, bootstrap3, в Back-end (серверная часть) PHP-5.6 и MySQL.

**Задача**

Разработать программу (игра) Магический квадрат на языке программирования javascript, которая будет встроена в свой сайт.

**Ход работы**

Суть данной игры заключается в то, что нужно попытаться привести в порядок квадраты (в поле 3 на 3) так как они первоначально стояли в последовательности от 1 до 8. '0' это пустая клетка, и щелкнув квадрат рядом с 0, чтобы заставить их обменять места! После того как мы привели квадраты в порядок, выводиться диалоговое окно, в котором написано: Ты смог! Хочешь начать заново?'. В этом окне будет 2 кнопки (ДА, Нет), если мы нажимаем кнопку да то уровень возвращается к первоначальному виду (квадраты перешиваются), после чего нужно будет снова их собрать в правильном порядке, если же нажили кнопку Нет, то ничего не происходит (уровень не перезагружается), пока ы не нажмем кнопку: Начать заново, после чего произойдет перемешивание квадратов и ы снова сможем приступать играть.

Для реализации этой игры, потребуется несколько функций.

Первым делом нам потребуется функция заполнения массива кнопок, также проверка на выигрыш, функция перестановки квадратов, функция новой игры.

Сама реализация игры, проста, суть такова: есть таблица в которой приведены квадраты. Квадратами являются 9 тэгов <input> у которых тип button, также у них присутствует значение пункта списка, которое будет отправлено на сервер (value = порядковому числу квадрата).

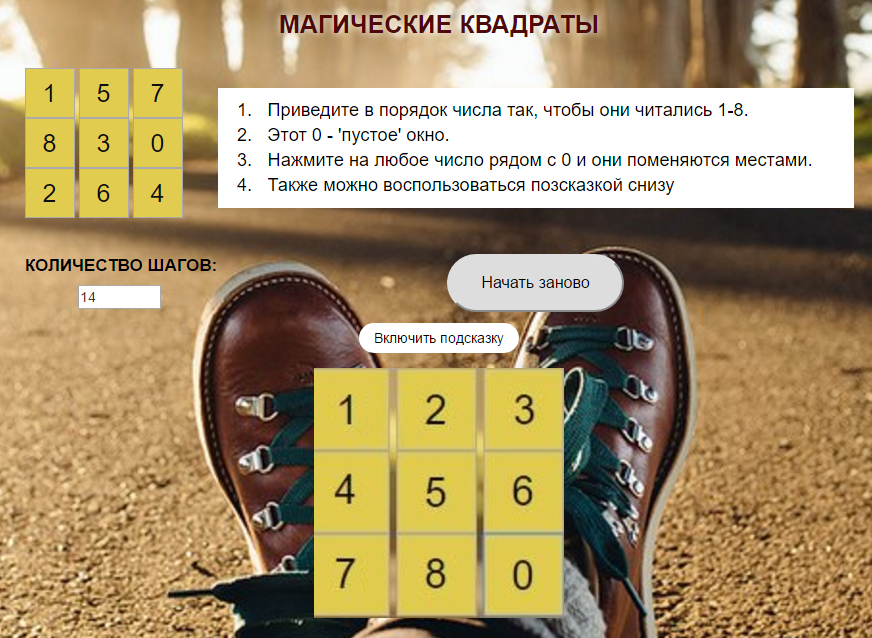
<table border=0 celpadding=0 cellspacing=10>  
 <tr>  
 <td class="cool">  
 <input type="button" value=" 1 " onClick="window.move(0)">  
 <input type="button" value=" 2 " onClick="window.move(1)">  
 <input type="button" value=" 3 " onClick="window.move(2)"><br>  
 <input type="button" value=" 4 " onClick="window.move(3)">  
 <input type="button" value=" 5 " onClick="window.move(4)">  
 <input type="button" value=" 6 " onClick="window.move(5)"><br>  
 <input type="button" value=" 7 " onClick="window.move(6)">  
 <input type="button" value=" 8 " onClick="window.move(7)">  
 <input type="button" value=" 0 " onClick="window.move(8)">  
 <br>  
 </td>  
 </td>  
 <td valign=top>  
 <br>  
 <ol>  
 <li>Приведите в порядок числа так, чтобы они читались 1-8.</li>  
 <li>Этот 0 - 'пустое' окно.</li>  
 <li>Нажмите на любое число рядом с 0 и они поменяются местами.</li>  
 <li>Также можно воспользоваться позсказкой снизу</li>  
 </ol>  
 </td>  
 </tr>  
 <tr>  
 <td align=center>  
 <h3>Количество шагов:</h3>  
 <input name="moves" size=7 value="0">  
 <br>  
 </td>  
 <td align=center><br>  
 <input type="submit" class="button" value="Начать заново" onClick="window.newgame()">  
 </td>  
 </tr>  
</table>

Также создадим подсказку для пользователей. При нажатии на кнопку <Показать подсказку> будет появляться картинка как должен выглядеть квадрат, после повторного нажатия картинка будет исчезать. Это делается с помощью функции в которой будет изменяться css, при нажатии блок будет показываться, т.е. display станет block, а при повторном нажатии (иначе) display станет снова none, как и был первоначально.

<div id="game">  
 <center>  
 <br>  
 <img src="img/game.jpg" width="250" height="250" alt="JPG">  
 </center>  
 <div class="tn-progress"></div>  
</div>

<script>  
 **function** gamebtn(){  
 **if** ($("#game").css("display") == "block") {  
 $("#game").css("display","none");  
 } **else** {$("#game").css("display","block");}  
 }  
</script>

Внешний вид игры представлен на Рис.1



**Рис. 1 Дизайн игры**

Также программа подсчитывает количество шагов, сделанное пользователе и выводи его ниже самого Магического квадрата. Для этого создается переменная nummoves, которая увеличивается при каждом движении квадрата.

**function** move(num) {  
 **if** (num < 8 && pos[num + 1] == 0) {  
 pos[num + 1] = pos[num];  
 pos[num] = 0;nummoves++;  
 }  
 **else if** (num > 0 && pos[num - 1] == 0) {  
 pos[num - 1] = pos[num];  
 pos[num] = 0;nummoves++;  
 }  
 **else if** (num > 2 && pos[num - 3] == 0) {  
 pos[num - 3] = pos[num];  
 pos[num] = 0;nummoves++;  
 }  
 **else if** (num < 6 && pos[num + 3] == 0) {  
 pos[num + 3] = pos[num];  
 pos[num] = 0;nummoves++;  
 }  
 display(pos);  
}

Функция движения отвечает за перемещение квадратов в таблице по input, в котором проверяется движение на каждую сторону (4 условие), после чего происходит смена значений квадратов (смена места соседнего квадрата с 0) и происходит увеличение счетчика ходов (nummoves).

Наконец то, когда мы получили правильный порядок квадратов, проверяется условие победу (функция win). В которой всего одно условие на то, что каждый ли квадрат на своем месте.

if (pos[0] == 1 & pos[1] == 2 & pos[2] == 3 & pos[3] == 4 &

pos[4] == 5 & pos[5] == 6 & pos[6] == 7 & pos[7] == 8 & pos[8] == 0) {

if (confirm('Ты смог! Хочешь начать заново?')) newgame();}

**Вывод**

Изучил такие аспекты javascript такие как динамический массив и html элементами.

Была реализована задача – написать игру (магический квадрат) на js и были реализованы свои функции движения для игры, после чего игра была добавлена на сайт.

**Приложение №1**

<script language="JavaScript">

function initArray() {

this.length = initArray.arguments.length

for (var i = 0; i < this.length; i++)

this[i + 1] = initArray.arguments[i]

}

var pos = new initArray(9, 9, 9,

9, 9, 9,

9, 9, 9);

var nummoves = 0;

function random() {

today = new Date();

num = today.getTime();

num = Math.round(Math.abs(Math.sin(num) \* 1000000)) % 9;

return num;

}

function display(pos) {

for (var i = 0; i < 9; i++) {

document.forms[0].elements[i].value = pos[i];

}

document.forms[0].moves.value = nummoves;

win();

}

function move(num) {

if (num < 8 && pos[num + 1] == 0) {

pos[num + 1] = pos[num];

pos[num] = 0;

nummoves++;

}

else if (num > 0 && pos[num - 1] == 0) {

pos[num - 1] = pos[num];

pos[num] = 0;

nummoves++;

}

else if (num > 2 && pos[num - 3] == 0) {

pos[num - 3] = pos[num];

pos[num] = 0;

nummoves++;

}

else if (num < 6 && pos[num + 3] == 0) {

pos[num + 3] = pos[num];

pos[num] = 0;

nummoves++;

}

display(pos);

}

function win() {

if (pos[0] == 1 & pos[1] == 2 & pos[2] == 3 & pos[3] == 4 &

pos[4] == 5 & pos[5] == 6 & pos[6] == 7 & pos[7] == 8 & pos[8] == 0) {

if (confirm('Ты смог! Хочешь начать заново?')) newgame();

}

}

function newgame() {

var x = 1;

for (var i = 0; i < 9; i++) {

pos[i] = 9;

}

for (var i = 0; i < 9; i++) {

randomnum = random();

if (randomnum == 9) randomnum = 8;

x = 1;

for (var j = 0; j < 9; j++) {

if (j == i)

continue;

if (randomnum == pos[j]) {

x = 0;

break;

}

}

if (x == 0) {

i--;

continue;

}

pos[i] = randomnum;

}

nummoves = 0;

display(pos);

}

</script>

<body onLoad="window.newgame()">

<form>

<table border=0 celpadding=0 cellspacing=10>

<tr>

<td class="cool">

<input type="button" value=" 1 " onClick="window.move(0)">

<input type="button" value=" 2 " onClick="window.move(1)">

<input type="button" value=" 3 " onClick="window.move(2)"><br>

<input type="button" value=" 4 " onClick="window.move(3)">

<input type="button" value=" 5 " onClick="window.move(4)">

<input type="button" value=" 6 " onClick="window.move(5)"><br>

<input type="button" value=" 7 " onClick="window.move(6)">

<input type="button" value=" 8 " onClick="window.move(7)">

<input type="button" value=" 0 " onClick="window.move(8)">

<br>

</td>

</td>

<td valign=top>

<br>

<ol>

<li>Приведите в порядок числа так, чтобы они читались 1-8.</li>

<li>Этот 0 - 'пустое' окно.</li>

<li>Нажмите на любое число рядом с 0 и они поменяются местами.</li>

<li>Также можно воспользоваться позсказкой снизу</li>

</ol>

</td>

</tr>

<tr>

<td align=center>

<h3>Количество шагов:</h3>

<input name="moves" size=7 value="0">

<br>

</td>

<td align=center><br>

<input type="submit" class="button" value="Начать заново" onClick="window.newgame()">

</td>

</tr>

</table>

</form>

<a onclick="gamebtn();" id="gamebtn" class="help">Включить подсказку</a>

<div id="game">

<center>

<br>

<img src="img/game.jpg" width="250" height="250" alt="JPG">

</center>

<div class="tn-progress"></div>

</div>

<script>

function gamebtn(){

if ($("#game").css("display") == "block") {

$("#game").css("display","none");

} else {$("#game").css("display","block");}

}

</script>