Муниципальное общеобразовательное учреждение

Центр образования №8 г.Тулы

Исследовательская работа

**Моя первая компьютерная программа**

**Автор: Гаев Григорий**,

учащийся 4-А класса

**Руководитель:** **Коробова М.А.**,

учитель начальных классов

2020

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ………………………………………………………………… | 3 |
| 1. О ЯЗЫКАХ ПРОГРАММИРОВАНИЯ ……………………………….. | 4 |
| * 1. Веб-программирование и веб-языки ……………………………. | 4 |
| * 1. Что такое HTML? ………………………………………………… | 5 |
| * 1. Из истории возникновения HTML ………………………………. | 6 |
| * 1. Современный HTML ………………………………………........... | 6 |
| * 1. Браузеры …………………………………………………………... | 7 |
| * 1. Таблицы каскадных стилей CSS ……..………………………….. | 7 |
| * 1. О языке JavaScript ……………………………..…………………. | 8 |
| 1. РАЗРАБОТКА ИГРЫ «КРЕСТИКИ - НОЛИКИ» ………………….. | 10 |
| * 1. Постановка задачи …………………………………………........... | 10 |
| * 1. Входные и выходные данные ……………………………………. | 10 |
| * 1. Разработка алгоритма решения задачи …………………………. | 11 |
| * 1. Реализация алгоритма …………………………………................. | 12 |
| ЗАКЛЮЧЕНИЕ …………………………………………………………… | 17 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ …………………….. | 18 |
| ПРИЛОЖЕНИЕ. Листинг программы …………………………………... | 19 |

**ВВЕДЕНИЕ**

Я люблю играть в компьютерные игры. Но в последнее время мне захотелось разобраться как их делают. Однако, как оказалось это большой и трудный процесс. Создание большой игры требует больших знаний, которые я захотел получить. Мой папа мне подсказал идею совместного обучения языку по итогам чтения познавательной литературы. Мы приобрели несколько интересных интерактивных книг и начали программировать. Моя работа посвящена этому интересному процессу.

***Актуальность исследования.*** Программирование – это актуальная на сегодняшний день сфера деятельности человека, результаты которой применяются повсеместно. При решении задач по программированию требуются знания логики, математики, информатики и других наук. При этом очень интересно изучать разные языки программирования, чтобы в дальнейшем быть хорошим специалистом.

***Цель и задачи исследования:***Рассмотреть некоторые языки программирования, выбрать простую игру для реализации на выбранных языках программирования, получить навыки работы с текстовыми редакторами по составлению алгоритма задачи и ее программированию.

***Объект исследования:***Web – языки программирования.

***Методы исследования:*** изучение литературы и обращение к сети Интернет.

**1. О ЯЗЫКАХ ПРОГРАММИРОВАНИЯ**

О языках программирования написано много книг. Их насчитывается огромное количество. Что выбрать в качестве объекта иcследования? Мы решили, что хотим познакомиться с такими языками, которые применяются во Всемирной паутине Internet.

Отличительной особенностью этих языков является то, что программу можно писать в любом текстовом редакторе и сразу запускать в браузере. Простота использования и легкость вхождения, а также обилие учебной литературы – вот основные мотивы, по которым мы выбрали именно веб-языки, о некоторых из которых речь пойдет далее.

Выделим особо термин программирование – это запись алгоритма на языке программирования, который понимает ЭВМ.

**1.1. Веб-программирование и веб-языки**

Веб-программирование - раздел программирования, ориентированный на разработку специальных программ, которые называют веб-приложениями.

Веб-приложение – это особое приложение, использующее средства и технологии работы в сети Internet для взаимодействия пользователя и сети. Такое приложение, как правило, запускается в специальной программе – браузере.

Языки веб-программирования - это языки, которые в основном предназначены для работы с веб-технологиями. Перечислим некоторые из них: HTML, CSS, Javascript, PHP, ASP, ASPX, Perl, SSI, XML, DHTML, XHTML, Perl, Python, Ruby, Java, Groovy и много других.

Наибольшее распространения получила тройка языков – их иногда называют «святой троицей веб-языков»: HTML, CSS, Javascript. О них и пойдет речь дальше.

**1.2. Что такое HTML?**

HTML аббревиатура(от англ. *HyperText Markup Language* —«язык гипертекстовой разметки) - стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами. Полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML имеет много версий. Актуальная версия сегодня – это 5-ая версия. Полное описание языка занимает много страниц текста. Это описание формулируется в рамках объектной модели документа. При этом под объектной моделью документа подразумевается наличие разных элементов в данном документе, которыми можно управлять – назначать им оформление, менять содержимое в зависимости от условий и прочее.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по специальным сетевым протоколам HTTP в виде простого текста или с использованием шифрования. Вот пример документа:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>HTML Document</title>

</head>

<body>

<p><b>Этот текст будет полужирным, <i>а этот — ещё и курсивным</i> </b> </p>

</body>

</html>

В результате будет на экране выведено:

**Этот текст будет полужирным, *а этот - ещё и курсивным***.

**1.3. Из истории возникновения HTML**

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области программирования. HTML успешно справлялся с проблемой сложности путём определения небольшого набора специальных элементов - тегов.

С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Первым общедоступным описанием HTML был документ «Теги HTML», впервые упомянутый в Интернете Тимом Бернерсом-Ли в конце 1991 года. В нём описываются 18 элементов, составляющих первоначальный, относительно простой дизайн HTML Одиннадцать из этих элементов всё ещё существуют в HTML.

**1.4. Современный HTML**

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам отображения. В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью:

- цветной экран современного компьютера,

- монохромный экран органайзера,

- ограниченный по размерам экран мобильного телефона

- устройства и программы голосового воспроизведения текстов.

Однако современное применение HTML очень далеко от его изначальной задачи.

Вот некоторые примеры из новых тегов:

- тег <table> предназначен для создания в документах таблиц, но иногда используется и для оформления размещения элементов на странице;

- тег <div> предназначен для объявления специального класса, с которым потом можно управлять по-особому. Мы будем этим тегом активно пользоваться в нашей программе.

С течением времени основная идея платформенной независимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

**1.5. Браузеры**

Текстовые документы, содержащие разметку на языке HTML имеют расширение .html или .htm.

Они обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются Google Chrome, Mozilla Firefox, Opera, Internet Explorer и некоторые другие.

**1.6. Таблицы каскадных стилей CSS**

CSS (*Cascading Style Sheets* - *каскадные таблицы стилей*) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языка разметки HTML, но также применяется и к другим разметкам.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Он уменьшает код и позволяет удобней манипулировать оформлением содержимого документа. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как:

- экранное представление,

- печатное представление,

- чтение голосом (специальным голосовым браузером или программой чтения с экрана),

- выводе устройствами, использующими шрифт Брайля (для людей с ограниченными возможностями по зрению)

Стили по-разному подключаются к разметке HTML. Их можно подключать:

- по ссылке,

- из отдельного файла,

- внутри базовой разметки в заголовке

- в каждом конкретном элементе.

Мы будем использовать последние два способа.

**1.7. О языке JavaScript**

JavaScript **–** современный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах в качестве язык сценариев для придания интерактивности веб-страницам. Под интерактивностью понимается реагирование на движение и клики мыши, ввод клавиатуры и другие активные действия пользователя на странице в браузере.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java (это объектный высокоуровневый язык), но при этом лёгким для использования «непрограммистами». Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation в США.

Как и любой другой высокоуровневый язык, этот язык позволяет сделать практически любые операции с содержимым документа. Вот, например, скрипт, который выводит окно с надписью «Hello, World»

<script type="application/javascript">

alert('Hello, World!');

</script>

Спецификация этого языка занимает много печатных листов. Поэтому мы не будем детально описывать этот язык. В дальнейшем мы более детально рассмотрим код программы и выпишем некоторые из используемых конструкций и операторов языка.

**2. РАЗРАБОТКА ИГРЫ «КРЕСТИКИ - НОЛИКИ»**

**2.1. Постановка задачи**

Для того, чтобы на практике познакомиться с основными аспектами выбранных языков программирования, поставим несложную задачу реализовать компьютерную игру, которую можно будет запускать в браузере. Это будет известная всем игра – «крестики-нолики».

У нее следующие правила:

- Играют двое (компьютер исключаем пока).

- Игровое поле состоит и клеток 3×3.

- Соперники ставят по – очереди либо X, либо 0. Первым ставится X.

- Выигрывает тот, кто первым выставил свои элементы по вертикали, горизонтали или диагонали.

**2.2. Входные и выходные данные**

На входе нашего алгоритма имеем:

* два игрока №1 и №2.
* поле ячеек 3×3 в виде таблицы, каждую ячейку которой мы занумеруем специальным образом: (X, Y), где X – номер строки, Y – номер столбца:

|  |  |  |
| --- | --- | --- |
| (0, 0) | (0, 1) | (0, 2) |
| (1, 0) | (1, 1) | (1, 2) |
| (2, 0) | (2, 1) | (2, 2) |

Алгоритм будет останавливаться в двух ситуациях:

* либо какой-то игрок победил.
* либо заполнены все ячейки – ничья.

При этом условие победы можно сформулировать в виде совокупности трех ячеек. Всего выигрышных ситуаций на нашем поле – 8:

* По горизонтали:

1. (0,0), (0,1), (0,2),

2. (1,0), (1,1), (1,2),

3. (2,0), (2,1), (2,2),

* По вертикали:

4. (0,0), (1,0), (2,0),

5. (0,1), (1,1), (2,1),

6. (0,2), (1,2), (2,2),

* По диагонали:

7. (0,0), (1,1), (2,2),

8. (0,2), (1,1), (2,0).

**2.3. Разработка алгоритма решения задачи**

Опишем алгоритм очередного хода. Как мы уже договорились – начинает игру «Игрок № 1». Он ставит «X».

Алгоритм следующий:

1. Устанавливаем текущего игрока в игрока № 1.
2. При клике мышкой на ячейке выводим в ячейку элемент текущего игрока (X или 0).
3. Проверяем выигрышную ситуацию и фиксируем ее состояние (в каких ячейках она произошла)
4. Если выигрышная ситуация есть, то мы выводим информацию о победе и выходим, иначе изменяем текущего игрока (если был Игрок № 1, то становится Игрок № 2, иначе наоборот) и идем к п.2.

Информацию о ничьей мы не выводим.

**2.4. Реализация алгоритма**

В результате разработки мы получили программу, листинг которой представлен в ПРИЛОЖЕНИИ.

Разберем ее основные элементы.

Она состоит из следующих основных элементов:

1. Тип документа - <!DOCTYPE html> - специальный тег без закрывающегося тега. За ним идет общий тег <html></html>, внутри которого располагается содержимое нашей страницы. Именно это содержимое будет особым образом интерпретироваться браузерами и показываться.

Заметим, что далее речь идет уже только о содержимом тега<html></html>.

1. Заголовок – содержимое тега <head></head>:

Содержимое этого тега не показывается на странице и служит в основном специальным целям. Итак, по порядку мы там видим:

* 1. <title>Крестики - нолики</title> - этот текст показывается в заголовке страницы, но не самой странице.
  2. CSS – код – содержимое тега <style></style>. Здесь мы определяем оформление того, как показывать наши объекты-элементы на странице. У нас три стиля:

- заголовок игроков - .names

- ячейка к таблице - .row div

- строка в таблице - .row

Можно заметить, что их названия начинаются с точки.

1. Содержимое тега <body></body> - именно этот тег показывается на странице. Он условно разделен на две части:
   1. JavaScript – код – содержимое тега <script></script>.
   2. Базовая разметка HTML с содержимым.

Остановимся более детально теперь на содержимом JavaScript – кода.

Все состояние игры мы записываем в пяти переменных:

- currentPlayer – текущий игрок (X или 0),

- won – была ли победа,

- winIdFirst – первая победная ячейка,

- winIdSecond – вторая победная ячейка,

- winIdThird – третья победная ячейка.

Эти и другие локальные переменные мы так или иначе обрабатываем в следующих функциях:

1. Основная наша функция place(box) – обработчик клика мышкой по ячейке. О ней чуть ниже.
2. Вспомогательные функции, которые изменяют содержимое наших элементов страницы – текущего статуса хода игроков и таблицу крестиков-ноликов с помощью CSS стилей:

- printResults() – вывод и содержимого текущего статуса игроков и в случае победы вывод победной ситуации;

- printCurrentPlayer(lastPlayer, currentPlayer,

lastInnerText, currentInnerText)

если нет победы, то сюда приходим и показываем текущий статус игроков;

- printWinner(winner, luzer,

winnerInnerText, luzerInnerText)

сюда приходим, если надо выводить результаты победы какого-либо игрока.

1. Функции проверки выигрыша, в которых выставляется значение переменным won, winIdFirst, winIdSecond, winIdThird. Вот эти функции:

- checkGameBoard() – здесь проверяются все 8 выигрышных ситуаций путем вызова сheckWinner. Если хоть одна из них сработала, то фиксируем это и сразу выходим – это приведет к концу игры и выводу победных результатов.

- сheckWinner(firstId, secondId, thirdId) – если первая ячейка не пустая и ее содержимое совпадает со второй и третьей, то победа и запись переменных won, winIdFirst, winIdSecond, winIdThird.

Теперь вернемся и разберем более детальней основную нашу функцию place(box). Сюда поступает ячейка – box. Алгоритм там такой:

1. Мы проверяем, если победа или клик прошел по непустой ячейке, то выходим.

if(box.innerText != "" || won) return;

2. Иначе меняем элемент текущего игрока.

currentPlayer == "0" ? currentPlayer = "X" : currentPlayer = "0";

3. Устанавливаем X или 0 в ячейку.

box.innerText = currentPlayer;

4. Устанавливаем у ячейки другое оформление с помощью CSS.

box.style.background = "#0000FF";

box.style.color = "#00FF00";

5. Проверяем выигрыш и фиксируем состояние победы.

checkGameBoard();

6. Выводим результаты.

printResults();

Итак, самая сложная часть нашего кода позади. Осталось описать базовую часть HTML. Взглянем на нее:

<div class="names">

<div id="player\_1"><p>Игрок 1</p></div>

<div id="player\_2"><p>Игрок 2</p></div>

</div>

Это мы выводим состояние игроков вверху. Можно увидеть, что это выполнено в виде тегов <div></div>. Им присвоен ранее нами введенный CSS - стиль names, а игрокам присвоены id – некоторые идентификаторы. По ним мы будем в скрипте брать объектные элементы и менять у них цвет, текст и др.

Далее аналогичным образом мы определяем объектный элемент – таблицу строк, а в них – объекты - ячейки:

<div>

<div class="row">

<div id="0\_0" onclick="place(this)"></div>

<div id="0\_1" onclick="place(this)"></div>

<div id="0\_2" onclick="place(this)"></div>

</div>

<div class="row">

<div id="1\_0" onclick="place(this)"></div>

<div id="1\_1" onclick="place(this)"></div>

<div id="1\_2" onclick="place(this)"></div>

</div>

<div class="row">

<div id="2\_0" onclick="place(this)"></div>

<div id="2\_1" onclick="place(this)"></div>

<div id="2\_2" onclick="place(this)"></div>

</div>

</div>

Каждой ячейке мы присваиваем стиль и помечаем специальным id. Именно по ним мы берем соответствующие объекты из объектной модели всего документа, проверяем выигрышную ситуацию и обновляем содержимое ячейки.

А также каждой ячейке мы присваиваем обработчик нажатия клика (однократного) мыши и передаем себя, т.е. нажатую ячейку: onclick="place(this)". Далее при клике на ячейку вызовется наша основная функция place и все обновится на странице.

Теперь вернемся немного к CSS-стилям. Не заметили ничего странного в стилях? Нет. Ну тогда я опишу некоторые «фокусы» с ними.

1. Мы задействовали стиль names, видим стиль row на строке. А куда делся стиль row div? Ответ не очень очевиден. Он присвоился автоматически ячейкам, т.к. мы употребили специального слово в названии стиля – div, тем самым указали на подчинение ячейки строке. Браузер сам найдет этот стиль по первому слову row и присвоит ячейке.
2. Попробуйте убрать стиль row у всех трех строк таблицы. Как вы думаете, что получится? Ответ кроется в содержимом этого стиля, состоящего из одной переменной. Наше поле в виде таблицы 3×3 превратится в таблицу 1×9 – таблицу из одной строки из 9 ячеек. Всего одна команда (clear: both) сильно меняет расположение элементов на странице. Мы запрещаем показывать на одной строке справа от row другие row, и они автоматически переносятся на другую строку, создавая иллюзию таблицы, состоящей из трех строк и трех ячеек в них.

Такие интересные сведения можно подчерпнуть при изучении CSS - стилей. Поэтому отметим некоторые их важные использованные свойства:

* font-size – размер шрифта;
* text-align - как располагать текст по горизонтали (справа, слева или по центру);
* background-color – какого цвета будет заливка под шрифтом;
* padding – внутренний отступ от бордюра;
* border – толщина и цвет бордюра;
* height – высота элемента;
* width – ширина элемента;
* clear – запрет на определенное расположение элементов, например, нельзя располагать справа.

**ЗАКЛЮЧЕНИЕ**

Таким образом, в ходе исследования мы выполнили следующие задачи:

1. Теоретически познакомились с основными веб-языками программирования: HTML, CSS и JavaScript.
2. Поставили задачу реализовать несложный алгоритм с помощью этих языков.
3. Получили навыки набора теста в текстовом редакторе, используя копирование, вставку, табуляцию текста и другое.
4. Реализовали игру «Крестики - нолики» в виде файла с расширением .html.

Игру можно считать законченной, она имеет следующий функционал:

1. Она предназначена для двух игроков - человек. Компьютер не играет.
2. Она показывает текущий ход. И одновременно перекрашивает клетки. В них уже невозможно поменять ход.
3. Игра показывает выигрышную ситуацию, т.е. подкрашивает «Х» или «0» другим отличным от основного цвета шрифта цветом
4. Игра поздравляет с победой.

Для перезапуска игры надо обновить страницу. Ее можно запускать в любом известном браузере, в котором выставить поддержку java-скриптов.

Можно в дальнейшем доработать эту программу:

1. Реализовать стратегию игры компьютера.
2. Реализовать историю ходов с возможностью возвращаться вперед-назад по истории.
3. Увеличить поле и превратить крестики-нолики в известную китайско-японскую спортивную игру «рендзю», в которой надо первым выставить 5 в ряд по горизонтали, диагонали или вертикали на большом по размеру поле (15×15).

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. <https://ru.wikipedia.org/wiki/Веб-программирование>
2. <https://ru.wikipedia.org/wiki/HTML>
3. <https://ru.wikipedia.org/wiki/CSS>
4. <https://ru.wikipedia.org/wiki/JavaScript>
5. Девид Уитни. Программирование для детей. Пять самых крутых игр на HTML и Javascript. – СПб: Питер, 2020. - 224с.

**ПРИЛОЖЕНИЕ. Листинг программы**

<!DOCTYPE html>

<html>

<head>

<title>Крестики - нолики</title>

<style>

.names { font-size: 32pt; text-align: left; background-color: yellow; }

.row div {

padding: 10px; border: 3px solid black; height: 70px;

width: 70px; float: left; background-color: purple;

font-size: 32pt; text-align: center;

}

.row { clear: both; }

</style>

</head>

<body>

<script>

var currentPlayer="0";

var won = false;

var winIdFirst;

var winIdSecond;

var winIdThird;

function place(box) {

if(box.innerText != "" || won) return;

currentPlayer == "0" ? currentPlayer = "X" : currentPlayer = "0";

box.innerText = currentPlayer;

box.style.background = "#0000FF"

box.style.color = "#00FF00"

checkGameBoard();

printResults();

}

function printResults(){

var playerItem1 = document.getElementById("player\_1");

var playerItem2 = document.getElementById("player\_2");

if(won) {

if(currentPlayer=="X")

printWinner(playerItem1,playerItem2,"Игрок 1 (X) - Победитель!", "Игрок 2 (0) - проиграл");

else

printWinner(playerItem2,playerItem1,"Игрок 2 (0) - Победитель!", "Игрок 1 (X) - проиграл");

}

else {

if(currentPlayer=="X")

printCurrentPlayer(playerItem1,playerItem2, "Игрок 1 (X) сделал ход", "Текущий ход - Игрок 2 (0)");

else

printCurrentPlayer(playerItem2,playerItem1, "Игрок 2 (0) сделал ход", "Текущий ход - Игрок 1 (X)");

}

}

function printCurrentPlayer(lastPlayer, currentPlayer,

lastInnerText, currentInnerText){

lastPlayer.innerText = lastInnerText;

lastPlayer.style.color = "#00FF00";

currentPlayer.innerText = currentInnerText;

currentPlayer.style.color = "#000000";

}

function printWinner(winner, luzer, winnerInnerText, luzerInnerText){

winner.innerText = winnerInnerText;

winner.style.color = "#FF0000";

luzer.innerText = luzerInnerText;

luzer.style.color = "#000000";

document.getElementById(winIdFirst).style.color = "#FF0000";

document.getElementById(winIdSecond).style.color = "#FF0000";

document.getElementById(winIdThird).style.color = "#FF0000";

}

function checkGameBoard() {

checkWinner("0\_0","1\_0","2\_0"); if(won) return;

checkWinner("0\_1","1\_1","2\_1"); if(won) return;

checkWinner("0\_2","1\_2","2\_2"); if(won) return;

checkWinner("0\_0","0\_1","0\_2"); if(won) return;

checkWinner("1\_0","1\_1","1\_2"); if(won) return;

checkWinner("2\_0","2\_1","2\_2"); if(won) return;

checkWinner("0\_0","1\_1","2\_2"); if(won) return;

checkWinner("0\_2","1\_1","2\_0");

}

function checkWinner(firstId,secondId,thirdId){

var firstText = document.getElementById(firstId).innerText;

var secondText = document.getElementById(secondId).innerText;

var thirdText = document.getElementById(thirdId).innerText;

if(firstText != "" &&

firstText == secondText &&

firstText == thirdText){

won = true;

winIdFirst = firstId;

winIdSecond = secondId;

winIdThird = thirdId;

}

}

</script>

<div class="names">

<div id="player\_1">Игрок 1</div>

<div id="player\_2">Игрок 2</div>

</div>

<div>

<div class="row">

<div id="0\_0" onclick="place(this)"></div>

<div id="0\_1" onclick="place(this)"></div>

<div id="0\_2" onclick="place(this)"></div>

</div>

<div class="row">

<div id="1\_0" onclick="place(this)"></div>

<div id="1\_1" onclick="place(this)"></div>

<div id="1\_2" onclick="place(this)"></div>

</div>

<div class="row">

<div id="2\_0" onclick="place(this)"></div>

<div id="2\_1" onclick="place(this)"></div>

<div id="2\_2" onclick="place(this)"></div>

</div>

</div>

</body>

</html>