|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5** | |
| **по дисциплине** | |
| «Разработка клиент-серверных приложений» | |
|  | |
| Выполнил студент группы ИКБО-12-18 | Копотов М.А. |
| Принял ассистент кафедры ИППО | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись руководителя)* |

Москва 2020

**Содержание**

[1. Цель практической работы 3](#_Toc528748841)

[2. Задание 3](#_Toc528748842)

[3. Поэтапное выполнение практической работы 3](#_Toc528748843)

[5. Текст разметки 4](#_Toc528748844)

[6. Результат работы 4](#_Toc528748845)

[7. Вывод 4](#_Toc528748846)

[Список использованных источников 5](#_Toc528748847)

# 1 Цель практической работы

Изучить основы клиентского скриптового языка JavaScript, работу с объектной моделью документа DOM (Document Object Model), познакомиться с возможностями, предоставляемые фреймворком jQuery.

# Задание

Создать клиентский скрипт на языке JavaScript, выполняющий действия в соответствии с вариантом. Использовать возможности, предоставляемые объектной моделью документа DOM, использовать фреймворк jQuery (или аналогичный).

**Вариант 3:**

Игра «Жизнь».

Игра моделирует жизнь поколений гипотетической колонии живых клеток на прямоугольном игровом поле,

которые выживают, размножаются или погибают в соответствии со следующими правилами.

Для каждого поколения (шага игры) применяются следующие правила: каждая живая клетка, количество

соседей которой меньше двух или больше трёх, погибает; каждая живая клетка, у которой от двух до трёх соседей,

живёт до следующего хода; каждая мёртвая клетка, у которой есть ровно три со-седа, оживает.

Соседи клетки – это все соседние с ней клетки по горизонтали, вертикали и диа-гонали, всего восемь соседей.

Правила применяются ко всему игровому полю одновременно, а не к каждой из кле-ток по очереди.

То есть подсчёт количества соседей происходит в один момент перед следующим шагом, и изменения, происходящие в соседних клетках, не влияют на новое состо-яние клетки.

Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

# Поэтапное выполнение практической работы

1. создать html-документ;
2. написать скрипт в соответствии с заданным вариантом. Для

справки по языку Javascript можно использовать источники, расположенные по адресам http://learn.javascript.ru и http://javascript.ru. Для справки по фреймворку jQuery можно использовать источники, расположенные по адресам http://jquery.com и http://jquery-docs.ru;

1. протестировать созданный документ.

# Текст разметки

Index.html

<!DOCTYPE html>

<!-- Лабораторная работа выполнена Копотовым М.А. -->

<html>

<head>

<meta charset="utf-8">

<title>Лабораторная работа №5</title>

<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript" src="Script.js"></script>

<link rel="stylesheet" href="Style.css"/>

</head>

<body>

<div id="counter">00</div>

<div id="osmos"></div>

</body>

</html>

Style.css

body{

background: #222222;

}

#counter{

position: absolute;

top: 1vh;

bottom: calc(51vh + 40vmin);

left: 0px;

right: 0px;

text-align: center;

font-size: calc(47vh - 40vmin);

color: #63635C;

}

#osmos{

position: absolute;

top: 50%;

left: 50%;

width: 80vmin;

height: 80vmin;

margin: -40vmin 0 0 -40vmin;

background: #63635C;

}

.cell{

position: absolute;

background-repeat: no-repeat;

background-size: cover;

border-color: #00000000;

}

.alive{

background-image: url(alive.png);

}

.dead{

background-image: url(dead.png);

}

Script.js

/\*

Игра «Жизнь».

Игра моделирует жизнь поколений гипотетической колонии живых клеток на прямоугольном игровом поле,

которые выживают, размножаются или погибают в соответствии со следующими правилами.

Для каждого поколения (шага игры) применяются следующие правила: каждая живая клетка, количество

соседей которой меньше двух или больше трёх, погибает; каждая живая клетка, у которой от двух до трёх соседей,

живёт до следующего хода; каждая мёртвая клетка, у которой есть ровно три соседа, оживает.

Соседи клетки – это все соседние с ней клетки по горизонтали, вертикали и диагонали, всего восемь соседей.

Правила применяются ко всему игровому полю одновременно, а не к каждой из клеток по очереди.

То есть подсчёт количества соседей происходит в один момент перед следующим шагом, и изменения, происходящие в соседних клетках, не влияют на новое состояние клетки.

Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

\*/

var osmos;

var cells;

var count = 0;

var size = 30;

$(function () {

size = prompt("Введите размер поля", 10);

cells = new Array(size);

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

cells[i] = new Array(size);

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

$("#osmos").append("<div class='cell' id='"+idOf(i,j)+"'>");

var c = new Cell($("#osmos").find("#" + idOf(i,j)));

c.element.css("top", (100.0 \* i / size) +'%');

c.element.css("left", (100.0 \* j / size) +'%');

c.element.css("height", (100.0 / size) +'%');

c.element.css("width", (100.0 / size) +'%');

if(getRandomInt(10) > 6)

c.Spawn();

cells[i][j] = c;

}

}

setTimeout(update, 500);

});

function update(){

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

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

var c = cells[i][j];

var n = countNeighbors(i,j);

if(c.alive && (n < 2 || n > 4)){

c.Kill();

}

if(!c.alive && n == 3)

c.Spawn();

}

}

count++;

$("#counter").text(count);

setTimeout(update, 500);

}

function countNeighbors(x, y){

var c = 0;

for(var i = -1; i <= 1; i++){

for(var j = -1; j <= 1; j++){

var nx = x + i;

var ny = y + j;

if(nx >= 0 && nx < size

&& ny >= 0 && ny < size

&& i != 0 && j != 0

&& cells[nx][ny].alive){

c = c + 1;

}

}

}

return c;

}

function idOf(x,y){

return "cell\_" + (x \* size + y);

}

function getRandomInt(max) {

return Math.floor(Math.random() \* Math.floor(max));

}

class Cell {

constructor(element) {

this.element = element;

this.Kill();

}

Spawn(){

this.alive = true;

this.element.removeClass("dead");

this.element.addClass("alive");

}

Kill(){

this.alive = false;

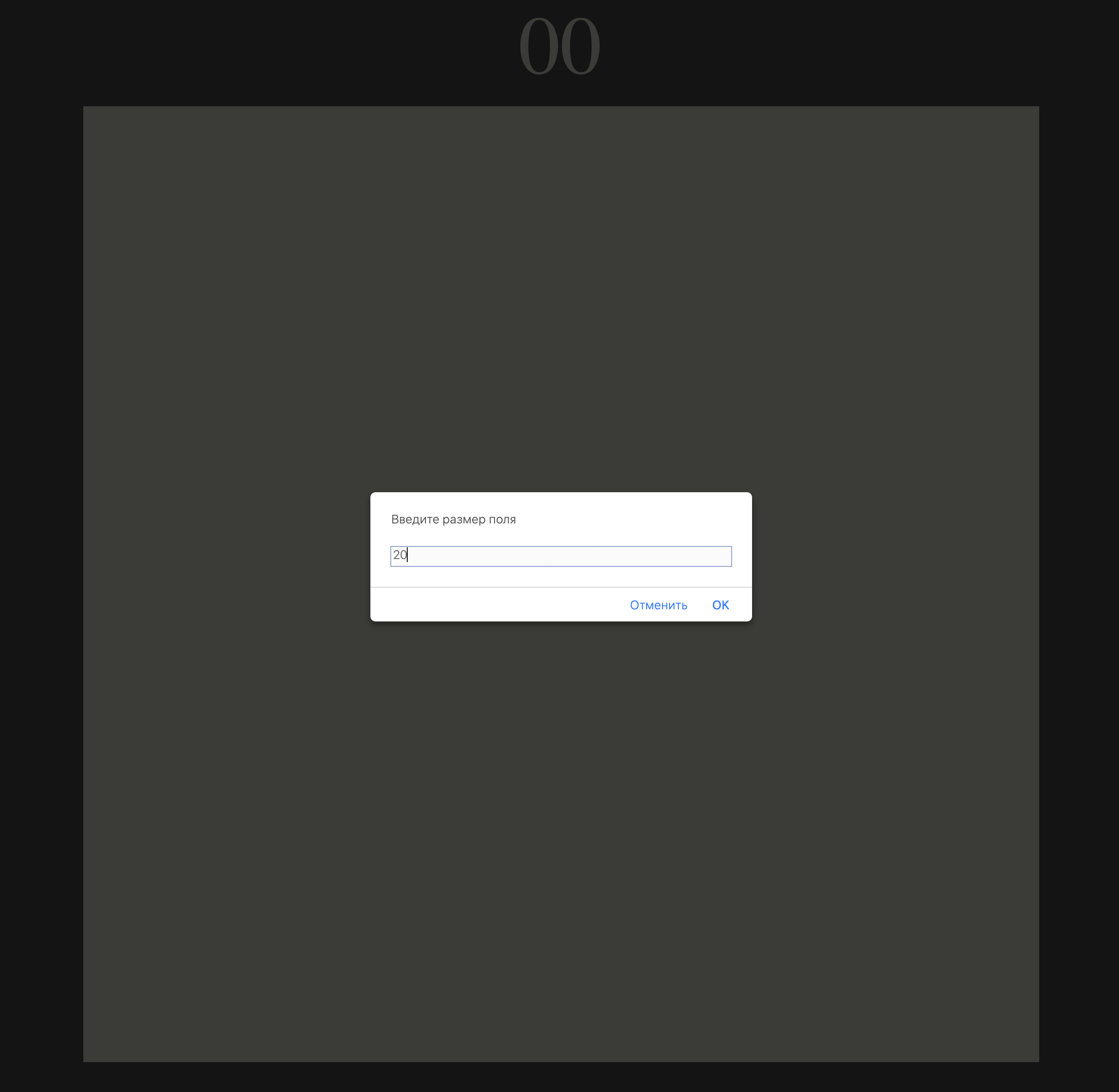
this.element.removeClass("alive");

this.element.addClass("dead");

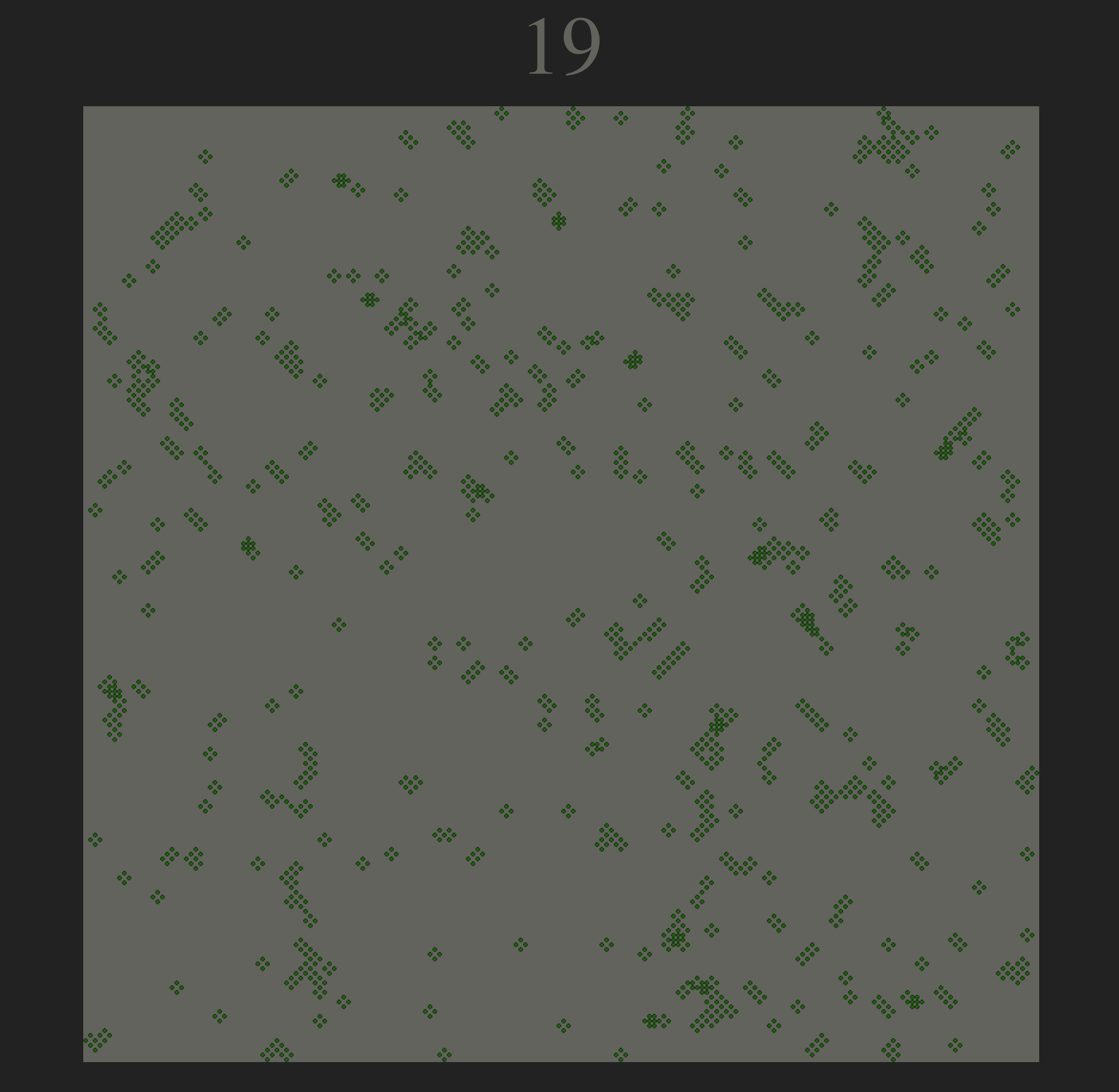
}

}

# 5 Результат работы



**Рисунок 1. Запуск страницы**

****

**Рисунок 2. Внешний вид поля спустя 19 жизненных циклов**

# 6 Вывод

В данной работе мы ознакомились и научились работать со скриптовым языком JavaScript, объектной моделью документа DOM и возможностями фреймворка jQuery. Все полученные знания мы использовали для создания настраиваемой анимации текста в соответствии с вариантом.

# Список использованных источников

1. http://umnik.rikt.ru/informat/Library/html2/
2. https://www.w3schools.com/html/default.asp
3. <https://html5book.ru/html-html5>
4. <https://www.w3.org/standards/techs/html>
5. http://validator.w3.org