ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

Институт компьютерных наук и технологий

**Отчет о прохождении учебной практики**

|  |
| --- |
| Шрамков Максим Андреевич |

*(Ф.И.О. обучающегося)*

|  |
| --- |
| 2 курс, гр.3530901/80003 |

|  |
| --- |
| 09.03.01 «Информатика и вычислительная техника» |

*(Направление подготовки (код и наименование)*

|  |
| --- |
| **Место прохождения практики**: Высшая школа информационных систем и суперкомпьютерных технологий (ВШИСиСТ) ИКНТ ФГАОУ ВО «СПбПУ» с использованием электронного обучения и дистанционных образовательных технологий. |

|  |
| --- |
| **Сроки практики:** с 22июня по 18июля 2020 г. |

|  |
| --- |
| **Руководитель практики:** |

|  |
| --- |
| ЖвариковВ.А., к.т.н., доцент ВШИСиСТ ИКНТ |

*(Ф.И.О., уч.степень, должность)*

|  |
| --- |
| **Оценка (зачет): \_\_\_\_\_\_\_\_\_\_\_** |

|  |
| --- |
|  |

|  |
| --- |
| Руководитель практики: Жвариков В.А. |

|  |
| --- |
| Обучающийся: Шрамков М.А. |

|  |
| --- |
| Дата: |

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

Институт компьютерных наук и технологий

**ИНДИВИДУАЛЬНЫЙ ПЛАН (ЗАДАНИЕ И ГРАФИК)**

**ПРОВЕДЕНИЯ ПРАКТИКИ**

|  |  |
| --- | --- |
| Ф.И.О. обучающегося Шрамков Максим Андреевич |  |

|  |  |
| --- | --- |
| **Направление подготовки** (код/наименование): 09.03.01, «Информатика и вычислительная техника» | |
| **Профиль** (код/наименование): 09.03.01 | |
| **Вид практики:** учебная | |
| **Тип практики:** ознакомительная | |
| **Место прохождения практики**: Высшая школа информационных систем и суперкомпьютерных технологий (ВШИСиСТ) ИКНТФГАОУ ВО «СПбПУ»с использованием электронного обучения и дистанционных образовательных технологий. | |
| |  | | --- | |  | | Руководитель практики: Жвариков В. А., к.т.н., доцент ВШИСиСТИКНТ | | *(Ф.И.О., уч.степень, должность)* | |

**Рабочий график проведения учебной практики**

Сроки практики: с **22.06.2020** г. по **18.07.2020**г.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Этапы (периоды) практики | Вид работ | Сроки прохождения этапа (периода) практики |
| 1 | Организационный этап | Установочная лекция (вебинар) для разъяснения целей, задач, содержания и порядка прохождения практики, выдача сопроводительных документов по практике  Основная тема: Написание аналога игры Toon Blast на языке JavaScript | 22.06.2020 г. |
| 2 | Основной  этап | Изучение программы Visual Studio Code, основ HTML, CSS и JavaScript, модуля SCSS  Сбор информации, обработка, систематизация и анализ фактического и теоретического материала.  Содержание практики: разработка сайта с аналогом игры Toon Blast  Планируемые результаты прохождения практики: получение навыков создания сайта, изучение возможностей JavaScript, готовая игра на созданном сайте | 23.06.2020 - 15.07.2020 г. |
| 3 | Заключительный этап | Подготовка отчета | 16.07.- 17.07.2020 г. |
| Защита отчета по практике (сдача зачета) | 17.07 - 18.07.2020 г |

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шрамков М.А.

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жвариков В. А.

Оглавление

[Техническое задание 4](#_Toc45659142)

[Метод решения 4](#_Toc45659143)

[Скриншоты программы 6](#_Toc45659144)

[Список использованных источников 8](#_Toc45659145)

[Приложение 8](#_Toc45659146)

# Техническое задание

Цель работы: освоить основы создания сайтов с помощью Html, Css и JavaScript. Изучить модуль Scss. На основе полученных знаний – написать сайт, с аналогом игры Toon Blast на JavaScript.

Правила игры Toon Blast: Игра состоит из 10 уровней. На каждом из уровней находится игровое поле размером в определенное количество квадратов. Каждый квадрат имеет свой цвет (цветов - ограниченное количество). Каждый уровень имеет свои параметры для прохождения, например – сломать 15 красных квадратов. Каждый уровень ограничен числом ходов. При нажатии пользователем на квадрат – он ломается, также ломаются все квадраты того же цвета, к нему прилегающие. Если сломать 5-6 квадратов за ход – появляется ракета, которая уничтожает ряд, в котором она находится (в случайном направлении). Если собрать 7-8 – появляется бомба которая уничтожает квадраты вокруг себя. Если собрать 9+ - появляется шар, уничтожающий все квадраты того цвета, которым мы получили шар.

Структура сайта: Сайт будет состоять из 4 страниц: главное страница, правила игры, уровни и сама игра. Перемещаться между страницами можно с помощью меню.

GitHub-репозиторий: <https://github.com/k1unk/toon_blast>

# Метод решения

Сайт состоит из 13 страниц (Главная страница, список уровней, правила игры, и 10 уровней).

Во всех html-файлах одинаковая «шапка», в которой находится меню, и отличаются они лишь контентом: на главной странице и правилах игры – просто текст, на странице со списком уровней – ссылки на уровни, на страницах уровней – поле и информация об уровне.

Все html-файлы используют стили из файла toon\_blast.css. К этому файлу был подключен файл toon\_blast.scss, благодаря чему можно писать стили в scss файле, где они будут иметь более удобную для пользователя структуру, и они автоматически переводятся в css файл.

Реализация самой игры выполнена с помощью JavaScript. Каждый уровень имеет свой JS файл, однако все эти 10 файлов отличаются только значением основных переменных, таких как размер поля, количество ходов, и количество блоков, которые необходимо сломать. В начале игры – создается массив размером [2\*size][size], для того чтобы новые блоки были созданы заранее, где каждая ячейка поля (блок) представлена цифрой, которая обозначает цвет блока. Далее просто используя этот массив - отрисовывается поле в функции draw\_on\_start(). Каждый блок представлен картинкой. При нажатии на блок – вызывается функция clicked(). Через ID узнаём координаты, изменяем массив («падение» блоков), если необходимо – добавляем бонус (особая цифра в массиве). Далее начинается анимация падения и исчезновения блоков. Происходим это путём изменения величин top и opacity в стилях каждого блока. После этого создаются новые блоки – путём перерисовывания всего поля. Также после каждого хода – меняется информация о текущем состоянии уровня.

Полный код программы приведен в Приложении.

# Скриншоты программы

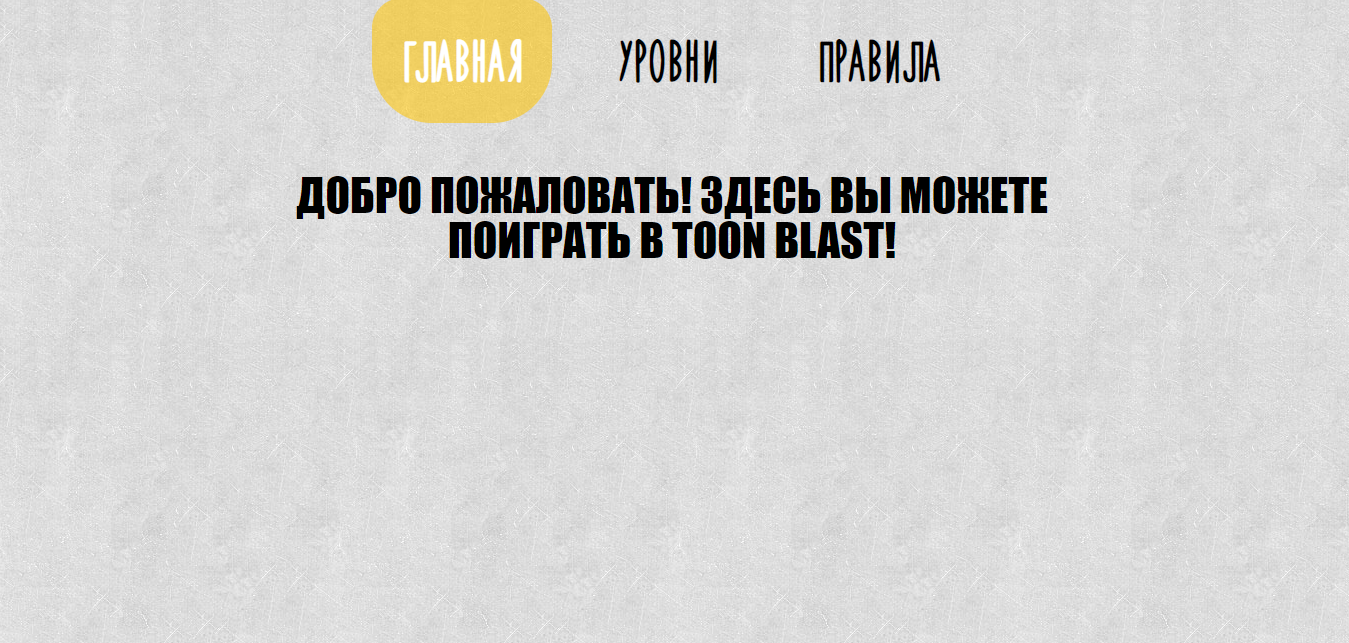


Рис. 1. Главная страница

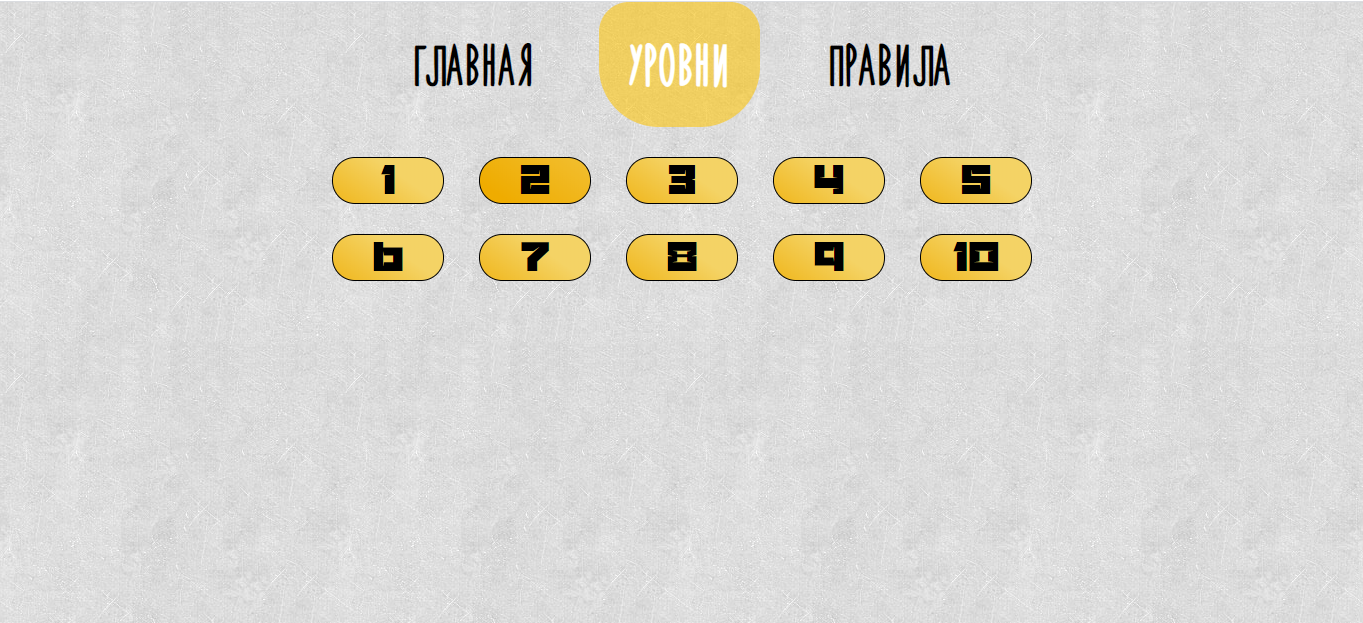


Рис. 2. Страница уровней игры

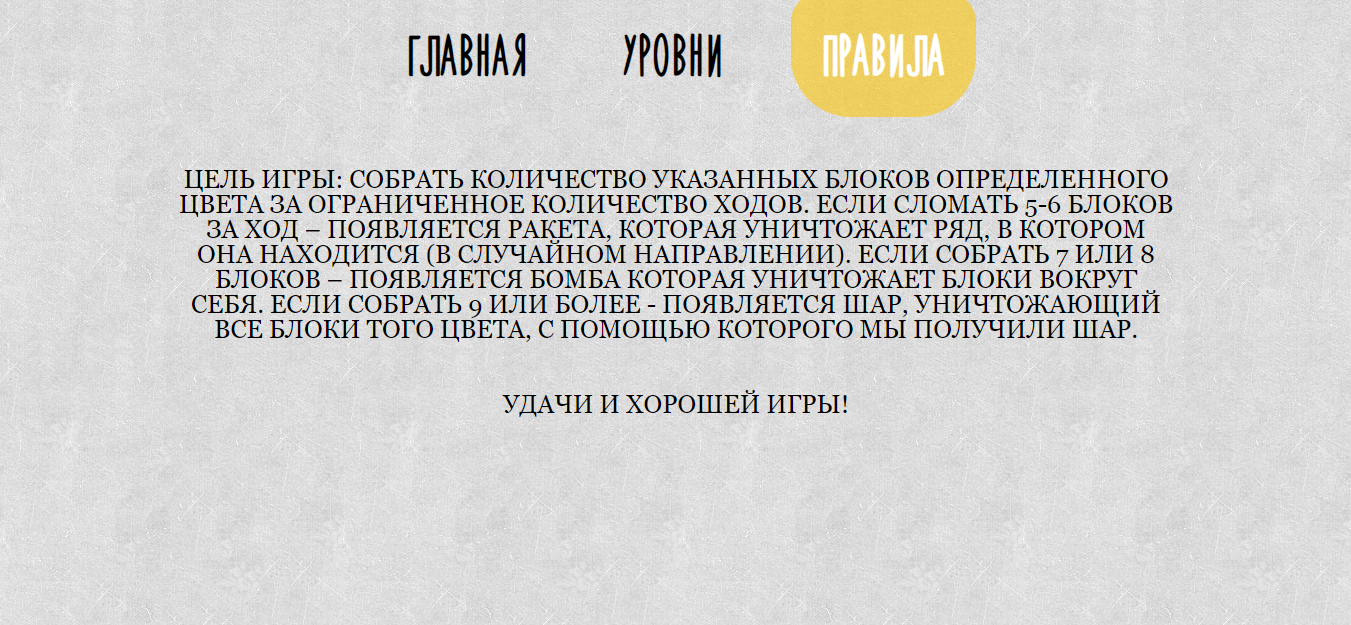


Рис. 3. Страница правил игры



Рис. 4. Один из уровней

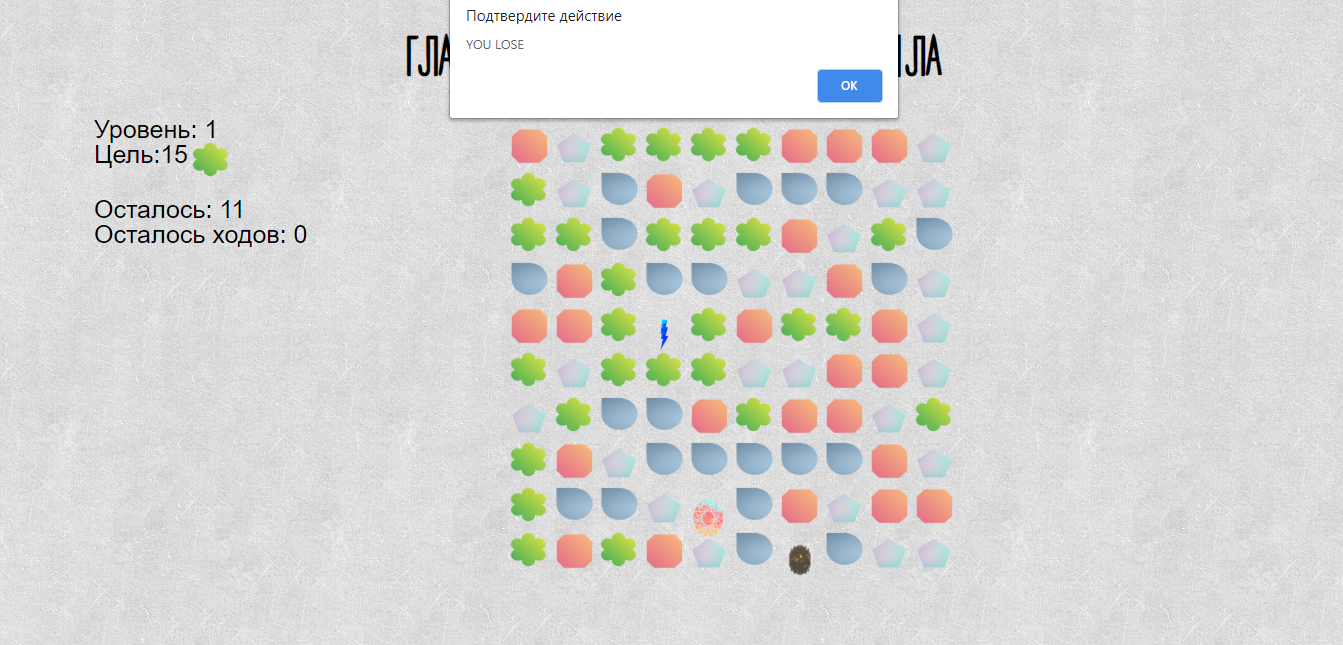


Рис. 5. Поле с полученными бонусами и объявление о поражении

# Список использованных источников

1. [sass-scss.ru](https://sass-scss.ru) – официальный сайт расширения, на котором можно найти инструкции по установке и использованию при разработке;
2. [learn.javascript.ru](https://learn.javascript.ru) – сайт с уроками и информацией о JavaScript

# Приложение

main.html

|  |  |
| --- | --- |
|  |  |
|  | <!-- Стандартное начало --> |
|  | <!DOCTYPE html> |
|  | <!-- указание языка --> |
|  | <html lang="ru"> |
|  |  |
|  | <!-- начало заголовка --> |
|  | <head> |
|  | <!-- кодировка --> |
|  | <meta charset="utf-8" /> |
|  | <!-- указание максимальной ширины страницы в зависимости от девайса --> |
|  | <meta name="viewport" content="width=device-width"> |
|  | <!-- Заголовок --> |
|  | <title>Toon Blast</title> |
|  | <!-- ссылка на css файл (файл стилей) --> |
|  | <link href="../css/toon\_blast.css" rel="stylesheet" type="text/css"> |
|  | <!-- конец заголовка --> |
|  | </head> |
|  |  |
|  | <!-- начало "тела" страницы --> |
|  | <body> |
|  | <!-- блок контента --> |
|  | <div class="content"> |
|  | <!-- блок заголовка --> |
|  | <header class="block\_\_header"> |
|  | <!-- блок меню --> |
|  | <nav class="menu"> |
|  | <!-- список пунктов меню --> |
|  | <ul> |
|  | <!-- ссылка на один из пунктов меню --> |
|  | <a class="selected" href="main.html"> |
|  | <!-- часть списка --> |
|  | <li> |
|  | <!-- текст, показывающий, на что ведет ссылка --> |
|  | <p>Главная</p> |
|  | </li> |
|  | </a> |
|  | <!-- тут по аналогии --> |
|  | <a class="not\_\_selected" href="levels.html"> |
|  | <li> |
|  | <p>Уровни</p> |
|  | </li> |
|  | </a> |
|  | <a class="not\_\_selected" href="help.html"> |
|  | <li> |
|  | <p>Правила</p> |
|  | </li> |
|  | </a> |
|  | </ul> |
|  | </nav> |
|  |  |
|  | </header> |
|  |  |
|  | <!-- текст --> |
|  | <p class="main\_\_text">Добро пожаловать! Здесь вы можете поиграть в Toon Blast!</p> |
|  |  |
|  | </div> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |

levels.html

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html lang="ru"> |
|  |  |
|  | <head> |
|  | <meta charset="utf-8" /> |
|  | <meta name="viewport" content="width=device-width"> |
|  | <title>Toon Blast - уровни</title> |
|  | <link href="../css/toon\_blast.css" rel="stylesheet" type="text/css"> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div class="content"> |
|  |  |
|  | <header class="block\_\_header"> |
|  | <nav class="menu"> |
|  | <ul> |
|  | <a class="not\_\_selected" href="main.html"> |
|  | <li> |
|  | <p>Главная</p> |
|  | </li> |
|  | </a> |
|  | <a class="selected" href="levels.html"> |
|  | <li> |
|  | <p>Уровни</p> |
|  | </li> |
|  | </a> |
|  | <a class="not\_\_selected" href="help.html"> |
|  | <li> |
|  | <p>Правила</p> |
|  | </li> |
|  | </a> |
|  | </ul> |
|  | </nav> |
|  | </header> |
|  |  |
|  | <!-- блок уровней --> |
|  | <div class="levels"> |
|  | <!-- ряд --> |
|  | <div class="row"> |
|  | <!-- ссылки на уровни --> |
|  | <a class="lvl" href="levels/lvl1.html"><p>1</p></a> |
|  | <a class="lvl" href="levels/lvl2.html"><p>2</p></a> |
|  | <a class="lvl" href="levels/lvl3.html"><p>3</p></a> |
|  | <a class="lvl" href="levels/lvl4.html"><p>4</p></a> |
|  | <a class="lvl" href="levels/lvl5.html"><p>5</p></a> |
|  | </div> |
|  | <!-- ряд --> |
|  | <div class="row"> |
|  | <!-- ссылки на уровни --> |
|  | <a class="lvl" href="levels/lvl6.html"><p>6</p></a> |
|  | <a class="lvl" href="levels/lvl7.html"><p>7</p></a> |
|  | <a class="lvl" href="levels/lvl8.html"><p>8</p></a> |
|  | <a class="lvl" href="levels/lvl9.html"><p>9</p></a> |
|  | <a class="lvl" href="levels/lvl10.html"><p>10</p></a> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | </body> |
|  |  |
|  | </html> |

Help.html

|  |
| --- |
|  |
|  | <!DOCTYPE html> |
|  | <html lang="ru"> |
|  |  |
|  | <head> |
|  | <meta charset="utf-8" /> |
|  | <meta name="viewport" content="width=device-width"> |
|  | <title>Toon Blast - помощь</title> |
|  | <link href="../css/toon\_blast.css" rel="stylesheet" type="text/css"> |
|  |  |
|  | </head> |
|  |  |
|  | <body> |
|  | <div class="content"> |
|  | <header class="block\_\_header"> |
|  | <nav class="menu"> |
|  | <ul> |
|  | <a class="not\_\_selected" href="main.html"> |
|  | <li> |
|  | <p>Главная</p> |
|  | </li> |
|  | </a> |
|  | <a class="not\_\_selected" href="levels.html"> |
|  | <li> |
|  | <p>Уровни</p> |
|  | </li> |
|  | </a> |
|  | <a class="selected" href="help.html"> |
|  | <li> |
|  | <p>Правила</p> |
|  | </li> |
|  | </a> |
|  | </ul> |
|  | </nav> |
|  |  |
|  | </header> |
|  |  |
|  | <!-- текст раздела "помощь" --> |
|  | <p class="help\_\_text">Цель игры: собрать количество указанных блоков определенного цвета за ограниченное |
|  | количество ходов. |
|  | Если сломать 5-6 блоков за ход – появляется ракета, которая уничтожает ряд, в котором |
|  | она находится (в случайном направлении). Если собрать 7 или 8 блоков – появляется бомба которая уничтожает |
|  | блоки |
|  | вокруг себя. Если собрать 9 или более - появляется шар, уничтожающий все блоки того цвета, с помощью |
|  | которого мы получили |
|  | шар. |
|  | </p> |
|  | <p class="help\_\_text">Удачи и хорошей игры! |
|  | </p> |
|  |  |
|  | </div> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |

lvl1.html

|  |
| --- |
|  |
|  | <!DOCTYPE html> |
|  | <html lang="ru"> |
|  |  |
|  | <head> |
|  | <meta charset="utf-8" /> |
|  | <meta name="viewport" content="width=device-width"> |
|  | <title>Toon Blast</title> |
|  | <link href="../../css/toon\_blast.css" rel="stylesheet" type="text/css"> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div class="content"> |
|  | <header class="block\_\_header"> |
|  | <nav class="menu"> |
|  | <ul> |
|  | <a class="not\_\_selected" href="../main.html"> |
|  | <li> |
|  | <p>Главная</p> |
|  | </li> |
|  | </a> |
|  | <a class="selected" href="../levels.html"> |
|  | <li> |
|  | <p>Уровни</p> |
|  | </li> |
|  | </a> |
|  | <a class="not\_\_selected" href="../help.html"> |
|  | <li> |
|  | <p>Правила</p> |
|  | </li> |
|  | </a> |
|  | </ul> |
|  | </nav> |
|  |  |
|  | </header> |
|  |  |
|  | <!-- блок контента --> |
|  | <div class="container"> |
|  | <!-- блок информации об уровне --> |
|  | <div class="data"> |
|  | <!-- текст, показывающий на каком мы уровне --> |
|  | <p class="lvlNum">Уровень: 1</p> |
|  | <!-- блок, показывающий цель уровня --> |
|  | <div class="aim"> |
|  | <!-- текст, показывающий цель уровня --> |
|  | <p class="aim">Цель: </p> |
|  | <p class="aim">15</p> |
|  | <!-- картинка, показывающая цель уровня --> |
|  | <img src="../../images/elem0.png" class="aimed"> |
|  | </div> |
|  | <!-- блок, в котором показывается, сколько осталось элементов для победы --> |
|  | <div class="div\_need\_to\_win"> |
|  | <!-- текст, в котором показывается, сколько элементов осталось для победы --> |
|  | <p class="p\_need\_to\_win">Осталось: 15</p> |
|  | </div> |
|  | <!-- блок, в котором показывается количество ходов --> |
|  | <div class="div\_steps"> |
|  | <!-- текст, в котором показывается количество ходов --> |
|  | <p class="p\_steps">Осталось ходов: 5</p> |
|  | </div> |
|  |  |
|  | </div> |
|  | <!-- поле игры --> |
|  | <div class="field"> </div> |
|  | </div> |
|  | </div> |
|  | </body> |
|  | <!-- ссылка на JS файл --> |
|  | <script src="../../js/lvl1.js"></script> |
|  |  |
|  | </html> |

Toon\_blast.scss

|  |
| --- |
|  |
|  |  |
|  | //импорт |
|  | @import "\_fonts"; |
|  | @import "\_null"; |
|  | @import "\_help"; |
|  | @import "\_levels"; |
|  | @import "\_main"; |
|  | @import "\_lvl"; |
|  |  |
|  | //константы шрифтов и цвета |
|  | $ff1: my1; |
|  | $ff2: my2; |
|  | $yellow: rgba(255, 196, 0, 0.562); |
|  |  |
|  | //тело страниц |
|  | body { |
|  | background-image: url(../images/bg.jpg); |
|  | max-width: 1240px; |
|  | margin: auto; |
|  | padding: 0 20px; |
|  | } |
|  |  |
|  | //меню (шапка) сайта (блок) |
|  | .block\_\_header { |
|  | display: flex; |
|  | //меню |
|  | .menu { |
|  | margin: auto; |
|  | //список |
|  | ul { |
|  | width: 600px; |
|  | display: flex; |
|  | justify-content: space-between; |
|  | } |
|  | //элемент списка |
|  | li { |
|  | padding: 0 30px; |
|  | display: flex; |
|  | border-radius: 30px 30px 60px 60px; |
|  | //текст |
|  | p { |
|  | margin: 40px 0; |
|  | font-size: 45px; |
|  | font-family: $ff1; |
|  | font-weight: 900; |
|  | text-transform: uppercase; |
|  | color: black; |
|  | } |
|  | } |
|  | //не выбранный элемент (выбранный - тот, на странице которого мы находимся) |
|  | .not\_\_selected li { |
|  | background-color: none; |
|  | transition: background-color 1s ease; |
|  | //при наведении |
|  | &:hover { |
|  | background-color: white; |
|  | transition: background-color 1s ease; |
|  | } |
|  | } |
|  | //выбранный элемент |
|  | .selected li { |
|  | background: $yellow; |
|  | //текст |
|  | p { |
|  | color: white; |
|  | } |
|  | } |
|  | } |
|  | } |

Fonts.scss

|  |  |
| --- | --- |
|  | @font-face { |
|  | font-family: "my1"; |
|  | src: url("../fonts/17659.eot"); |
|  | src: url("../fonts/17659.eot?iefix") format("eot"), url("../fonts/17659.ttf") format("woff"), |
|  | url("../fonts/17659.ttf") format("truetype"), url("../fonts/17659.ttf") format("svg"); |
|  | font-weight: normal; |
|  | font-style: normal; |
|  | } |
|  | @font-face { |
|  | font-family: "my2"; |
|  | src: url("../fonts/8684.eot"); |
|  | src: url("../fonts/8684.eot?iefix") format("eot"), url("../fonts/8684.ttf") format("woff"), |
|  | url("../fonts/8684.ttf") format("truetype"), url("../fonts/8684.ttf") format("svg"); |
|  | font-weight: normal; |
|  | font-style: normal; |
|  | } |

Help.scss

|  |
| --- |
|  |
|  | .help\_\_text { |
|  | margin: 50px 100px; |
|  | font-size: 25px; |
|  | font-family: Georgia, 'Times New Roman', Times, serif; |
|  | font-weight: 100; |
|  | text-transform: uppercase; |
|  | color: black; |
|  | text-align: center;; |
|  | } |

Levels.scss

|  |
| --- |
|  |
|  |  |
|  | //импортируем шрифты |
|  | @import "\_fonts"; |
|  |  |
|  | //константы шрифтов и желтого цвета |
|  | $ff1: my1; |
|  | $ff2: my2; |
|  | $yellow: rgba(255, 196, 0, 0.562); |
|  |  |
|  | //блок уровней |
|  | .levels { |
|  | margin: 30px auto 0 auto; |
|  | width: 700px; |
|  | //ряд |
|  | .row { |
|  | display: flex; |
|  | justify-content: space-between; |
|  | margin: 30px 0; |
|  | //ссылка на уровень |
|  | a { |
|  | width: 16%; |
|  | border: 1px solid black; |
|  | border-radius: 40px; |
|  | background: linear-gradient(35deg, $yellow, #ffffff); |
|  | //текст в ссылке, показывающий номер уровня |
|  | p { |
|  | margin: 0; |
|  | text-align: center; |
|  | font-size: 45px; |
|  | font-family: $ff2; |
|  | color: black; |
|  | } |
|  |  |
|  | transition: all 0.4s ease-in-out; |
|  | background-size: 140% 340%; |
|  | background-image: linear-gradient(35deg, $yellow 90%, #bba2a2 30%), |
|  | linear-gradient(35deg, #da8d00 30%, #e6e6e6 70%); |
|  | //при наведении на ссылку |
|  | &:hover { |
|  | background-size: 440% 400%; |
|  | } |
|  | } |
|  | } |
|  | } |

Lvl.scss

|  |  |
| --- | --- |
|  |  |
|  | //контент |
|  | .container { |
|  | max-width: 1240px; |
|  | margin: auto; |
|  | padding: 0 20px; |
|  | display: flex; |
|  |  |
|  | } |
|  | //размеры картинок (блоков поля в игре) |
|  | img { |
|  | width: 45px; |
|  | height: 55px; |
|  | } |
|  | //само поле |
|  | .field { |
|  | margin: 10px auto auto auto; |
|  | width: 550px; |
|  | position: relative; |
|  | transition: all 1s; |
|  | -webkit-transition: all 1s; |
|  | } |
|  | //блоки поля в игре |
|  | .blocks { |
|  | position: absolute; |
|  | transition: all 1s; |
|  | -webkit-transition: all 1s; |
|  | } |
|  | //данные об уровне |
|  | .data { |
|  | p { |
|  | font-size: 25px; |
|  | font-family: Arial, Helvetica, sans-serif; |
|  | font-weight: 300; |
|  | color: black; |
|  | } |
|  | } |
|  | //данные о цели уровня и количестве оставшихся ходов |
|  | .aim, .div\_steps { |
|  | display: flex; |
|  | } |

Main.scss

|  |  |
| --- | --- |
|  | .main\_\_text { |
|  | margin: 50px 200px; |
|  | font-size: 45px; |
|  | font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif; |
|  | font-weight: 500; |
|  | text-transform: uppercase; |
|  | color: black; |
|  | text-align: center;; |
|  | } |

Main.scss

|  |  |
| --- | --- |
|  | \* { |
|  | padding: 0; |
|  | margin: 0; |
|  | border: 0; |
|  | } |
|  |  |
|  | \*, \*:before, \*:after { |
|  | -moz-box-sizing: border-box; |
|  | -webkit-box-sizing: border-box; |
|  | box-sizing: border-box; |
|  | } |
|  |  |
|  | :focus, :active { |
|  | outline: none; |
|  | } |
|  |  |
|  | a:focus, a:active { |
|  | outline: none; |
|  | } |
|  |  |
|  | nav, footer, header, aside { |
|  | display: block; |
|  | } |
|  |  |
|  | html, body { |
|  | height: 100%; |
|  | width: 100%; |
|  | font-size: 100%; |
|  | line-height: 1; |
|  | font-size: 14px; |
|  | -ms-text-size-adjust: 100%; |
|  | -moz-text-size-adjust: 100%; |
|  | -webkit-text-size-adjust: 100%; |
|  | } |
|  |  |
|  | input, button, textarea { |
|  | font-family: inherit; |
|  | } |
|  |  |
|  | input::-ms-clear { |
|  | display: none; |
|  | } |
|  |  |
|  | button { |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | button::-moz-focus-inner { |
|  | padding: 0; |
|  | border: 0; |
|  | } |
|  |  |
|  | a, a:visited { |
|  | text-decoration: none; |
|  | } |
|  |  |
|  | a:hover { |
|  | text-decoration: none; |
|  | } |
|  |  |
|  | li { |
|  | list-style: none; |
|  | } |
|  |  |
|  | img { |
|  | vertical-align: top; |
|  | } |
|  |  |
|  | h1, h2, h3, h4, h5, h6 { |
|  | font-size: inherit; |
|  | font-weight: 400; |
|  | } |
|  |  |
|  | img { |
|  | width: 20px; |
|  | height: 20px; |
|  | } |

Lvl1.js

|  |
| --- |
|  |
|  | window.onload = function () { |
|  | //константы |
|  |  |
|  | //количество оставшихся шагов на уровне |
|  | let steps = 5 |
|  | //количество блоков для победы |
|  | let need\_to\_win = 15 |
|  | //блок какого цвета нам нужно собирать |
|  | let what\_color\_need\_to\_win = 0 |
|  | //количество цветов блоков на данном уровне |
|  | let colors = 4; |
|  | //размер поля |
|  | let size = 10; |
|  | //поле в html'e |
|  | let field = document.querySelector('.field'); |
|  | //поле в js'e (двумерный массив) |
|  | var arr = []; |
|  | //размер блока в пикселях |
|  | let sizeOfBlockPx = 45; |
|  | //список уничтоженных блоков |
|  | var killed\_blocks = []; |
|  | for (var i = 0; i < colors; i++) { |
|  | killed\_blocks[i] = 0 |
|  | } |
|  |  |
|  | //заполняем массив блоков |
|  |  |
|  | //0,1,2,3,4 - блок какого-либо цвета |
|  | //6 - уничтожение всех элементов горизонтального ряда (бонус) |
|  | //7 - уничтожение всех элементов вертикального ряда (бонус) |
|  | //8 - уничтожение всех элементов вокруг себя (бонус) |
|  | //9,10,11,12,13 - уничтожение всех элементов данного цвета (бонус) |
|  |  |
|  | //от 0 до size\*2 потому что нужны блоки, которые будут падать при исчезновении старых |
|  | for (var i = 0; i < size \* 2; i++) { |
|  | arr[i] = []; |
|  | for (var j = 0; j < size; j++) { |
|  | //каждая ячейка получает блок случайного цвета |
|  | arr[i][j] = Math.floor(Math.random() \* colors); |
|  | } |
|  | } |
|  |  |
|  | //функция отрисовки поля |
|  | draw\_on\_start(field) |
|  | function draw\_on\_start(field) { |
|  | out = ''; |
|  | //от size до size\*2 потому что нам не нужно отрисовывать запасные блоки |
|  | for (var i = size; i < size \* 2; i++) { |
|  | for (var j = 0; j < size; j++) { |
|  | for (var k = 0; k < 9 + colors; k++) { |
|  | if (arr[i][j] == k) { |
|  | //создание html эелмента |
|  | out += '<img src="../../images/elem'; |
|  | out += k; |
|  | out += '.png" alt="" class="blocks" id = '; |
|  | out += 'block' + (i - size) + j; |
|  | out += ' style="opacity: 1; top: ' |
|  | out += (i - size) \* sizeOfBlockPx |
|  | out += 'px; left: ' |
|  | out += j \* sizeOfBlockPx |
|  | out += 'px">'; |
|  | } |
|  | } |
|  | } |
|  | } |
|  | //добавление html элемента к html файлу |
|  | field.innerHTML += out; |
|  |  |
|  | //заполняем массив, в котором находятся ID всех наших блоков |
|  | blocks = [] |
|  | for (var i = 0; i < size; i++) { |
|  | blocks[i] = [] |
|  | for (var j = 0; j < size; j++) { |
|  | var id = ('#block' + i + j); |
|  | blocks[i][j] = document.querySelector(id); |
|  | } |
|  | } |
|  |  |
|  | //при нажатии на блок - вызывается функция clicked |
|  | for (var i = 0; i < size; i++) { |
|  | for (var j = 0; j < size; j++) { |
|  | blocks[i][j].onclick = clicked(arr, i + size, j); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //функция вызывается при нажатии пользователем на какой либо блок |
|  | function clicked(arr, i, j) { |
|  | return function () { |
|  | //запоминаем значение ячейки |
|  | now = arr[i][j]; |
|  |  |
|  | //смотрим значение нажатой ячейки и в зависимости от этого чтото делаем |
|  |  |
|  | //если это обычный блок - надо удалить все блоки, которые к нему прислонены, и все их такие блоки, и все их ... |
|  | if (now < colors && now > -1) { |
|  | //создание массивов |
|  | var deleted = []; |
|  | deleted[0] = 1 |
|  | list\_of\_i\_1 = [] |
|  | list\_of\_j\_1 = [] |
|  | list\_of\_i\_2 = [] |
|  | list\_of\_j\_2 = [] |
|  | list\_of\_i\_1.push(i) |
|  | list\_of\_j\_1.push(j) |
|  | list\_of\_i\_2.push(i) |
|  | list\_of\_j\_2.push(j) |
|  | array\_i\_j\_value = 0 |
|  | killed\_blocks[now] += 1 |
|  |  |
|  | //сам перебор блоков |
|  | while (list\_of\_i\_1.length != 0) { |
|  |  |
|  | i\_2 = list\_of\_i\_1[0] |
|  | j\_2 = list\_of\_j\_1[0] |
|  | list\_of\_i\_1.splice(0, 1); |
|  | list\_of\_j\_1.splice(0, 1); |
|  | arr[i\_2][j\_2] = -1; |
|  |  |
|  | if (i\_2 != size) { |
|  | clicked\_find\_all(i\_2 - 1, j\_2, arr, killed\_blocks, deleted, now) |
|  | } |
|  | if (i\_2 != size \* 2 - 1) { |
|  | clicked\_find\_all(i\_2 + 1, j\_2, arr, killed\_blocks, deleted, now) |
|  | } |
|  | if (j\_2 != 0) { |
|  | clicked\_find\_all(i\_2, j\_2 - 1, arr, killed\_blocks, deleted, now) |
|  | } |
|  | if (j\_2 != size - 1) { |
|  | clicked\_find\_all(i\_2, j\_2 + 1, arr, killed\_blocks, deleted, now) |
|  | } |
|  |  |
|  | function clicked\_find\_all(i\_3, j\_3, arr, killed\_blocks, deleted, now) { |
|  | if (arr[i\_3][j\_3] == now) { |
|  | list\_of\_i\_2.push(i\_3) |
|  | list\_of\_j\_2.push(j\_3) |
|  | list\_of\_i\_1.push(i\_3) |
|  | list\_of\_j\_1.push(j\_3) |
|  | arr[i\_3][j\_3] = -1; |
|  | killed\_blocks[now] += 1 |
|  | deleted[0] += 1 |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //если удалено более 4 блоков - надо добавить бонус |
|  | if (deleted[0] > 4 && deleted[0] < 7) { |
|  | if (Math.random() > 0.5) { |
|  | arr[i][j] = 6 |
|  | array\_i\_j\_value = 6 |
|  | } |
|  | else { |
|  | arr[i][j] = 7 |
|  | array\_i\_j\_value = 7 |
|  | } |
|  | } |
|  | if (deleted[0] > 6 && deleted[0] < 9) { |
|  | arr[i][j] = 8 |
|  | array\_i\_j\_value = 8 |
|  | } |
|  | if (deleted[0] > 8) { |
|  | arr[i][j] = 9 + now |
|  | array\_i\_j\_value = 9 + now |
|  | } |
|  |  |
|  | //вызываем функцию анимации удаленных блоков |
|  | animation\_death\_block(list\_of\_i\_2, list\_of\_j\_2, array\_i\_j\_value, deleted[0]) |
|  | } |
|  |  |
|  | // если это не обычный блок, а бонус - удаляем необходимые блоки в массиве (bonus\_death\_in\_arr), а |
|  | // затем вызываем анимацию (bonus\_death\_animation) |
|  | if (now == 6) { |
|  | for (var jj = 0; jj < size; jj++) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i, jj, arr) |
|  | } |
|  | bonus\_death\_animation(i, j, now) |
|  | } |
|  | if (now == 7) { |
|  | for (var ii = size; ii < size \* 2; ii++) { |
|  | bonus\_death\_in\_arr(killed\_blocks, ii, j, arr) |
|  | } |
|  | bonus\_death\_animation(i, j, now) |
|  | } |
|  | if (now == 8) { |
|  | arr[i][j] = -1 |
|  | if (i != size && j != 0) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i - 1, j - 1, arr); |
|  | } |
|  | if (i != size && j != size - 1) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i - 1, j + 1, arr); |
|  | } |
|  | if (i != size) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i - 1, j, arr); |
|  | } |
|  | if (i != size \* 2 - 1 && j != 0) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i + 1, j - 1, arr); |
|  | } |
|  | if (i != size \* 2 - 1 && j != size - 1) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i + 1, j + 1, arr); |
|  | } |
|  | if (i != size \* 2 - 1) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i + 1, j, arr); |
|  | } |
|  | if (j != 0) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i, j - 1, arr); |
|  | } |
|  | if (j != size - 1) { |
|  | bonus\_death\_in\_arr(killed\_blocks, i, j + 1, arr); |
|  | } |
|  | bonus\_death\_animation(i, j, now) |
|  | } |
|  | if (now > 8) { |
|  | n = now - 9 |
|  | arr[i][j] = -1 |
|  | bonus\_death\_animation(i, j, now) |
|  | for (var i\_2 = size; i\_2 < size \* 2; i\_2++) { |
|  | for (var j\_2 = 0; j\_2 < size; j\_2++) { |
|  | if (arr[i\_2][j\_2] == n) { |
|  | killed\_blocks[n] += 1 |
|  | arr[i\_2][j\_2] = -1 |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //удаление необходимых блоков при нажатии на бонус |
|  | function bonus\_death\_in\_arr(killed\_blocks, i\_3, j\_3, arr) { |
|  | for (var k = 0; k < colors; k++) { |
|  | if (arr[i\_3][j\_3] == k) { |
|  | killed\_blocks[k] += 1 |
|  | } |
|  | } |
|  | arr[i\_3][j\_3] = -1 |
|  | } |
|  |  |
|  | //падение блоков, анимация |
|  | for (var i\_1 = 0; i\_1 < size; i\_1++) { |
|  | for (var j\_1 = 0; j\_1 < size; j\_1++) { |
|  | for (var i\_2 = i\_1; i\_2 < size; i\_2++) { |
|  | if (blocks[i\_2][j\_1].style.opacity == 0) { |
|  | var re = /px/gi; |
|  | var str = blocks[i\_1][j\_1].style.top; |
|  | var newstr = str.replace(re, ""); |
|  | var mean = parseInt(newstr) |
|  | mean += sizeOfBlockPx |
|  | blocks[i\_1][j\_1].style.top = mean + "px" |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //падение блоков, в массиве |
|  | for (var i\_1 = 0; i\_1 < size; i\_1++) { |
|  | for (var j\_1 = size \* 2 - 1; j\_1 > size - 1; j\_1--) { |
|  | for (var j\_2 = size - 1; j\_2 > -1; j\_2--) { |
|  | if (arr[j\_1][j\_2] == -1) { |
|  | for (var i\_2 = j\_1; i\_2 > 0; i\_2--) { |
|  | if (i\_2 != 0) { |
|  | arr[i\_2][j\_2] = arr[i\_2 - 1][j\_2] |
|  | arr[i\_2 - 1][j\_2] = -1 |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //создание новых блоков, анимация (через 1100мс, потому что это время старые блоки будут падать и исчезать) |
|  | setTimeout(fallBlocks, 1100); |
|  | function fallBlocks() { |
|  | //удаляется старое поле, и создаётся новое |
|  | let elem = document.querySelector('.field'); |
|  |  |
|  | elem.parentNode.removeChild(elem); |
|  | let container = document.querySelector('.container'); |
|  | out = '' |
|  | out += '<div class="field"></div>' |
|  | container.innerHTML += out; |
|  |  |
|  | let field = document.querySelector('.field'); |
|  |  |
|  | //отрисовка нового поля |
|  | draw\_on\_start(field) |
|  | } |
|  |  |
|  | //создание новых блоков в массиве |
|  | for (var i\_1 = 0; i\_1 < size \* 2; i\_1++) { |
|  | for (var j\_1 = 0; j\_1 < size; j\_1++) { |
|  | if (arr[i\_1][j\_1] == -1) { |
|  | arr[i\_1][j\_1] = Math.floor(Math.random() \* colors); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //вывод в консоль |
|  | console.log(arr) |
|  | console.log(killed\_blocks) |
|  |  |
|  | //меняем информацию о процессе уровня |
|  |  |
|  | //сколько осталось уничтожить блоков |
|  | n = need\_to\_win - killed\_blocks[what\_color\_need\_to\_win] |
|  | if (n < 0) n = 0 |
|  | let elem = document.querySelector('.p\_need\_to\_win'); |
|  | elem.parentNode.removeChild(elem); |
|  | let container = document.querySelector('.div\_need\_to\_win'); |
|  | out = '' |
|  | out += '<p class="p\_need\_to\_win">Осталось: ' |
|  | out += n |
|  | out += '</p>' |
|  | container.innerHTML += out; |
|  |  |
|  | //сколько осталось ходов |
|  | steps -= 1 |
|  | let elem2 = document.querySelector('.p\_steps'); |
|  | elem2.parentNode.removeChild(elem2); |
|  | let container2 = document.querySelector('.div\_steps'); |
|  | out = '' |
|  | out += '<p class="p\_steps">Осталось ходов: ' |
|  | out += steps |
|  | out += '</p>' |
|  | container2.innerHTML += out; |
|  |  |
|  | //если кончились ходы - поражение, если сломали необходимое количество блоков - победа |
|  | if (steps == 0 && n > 0) { |
|  | setTimeout(alert, 1100, "YOU LOSE") |
|  | } |
|  | if (n < 1 && steps > -1) setTimeout(alert, 1100, "YOU WIN") |
|  | } |
|  |  |
|  | } |
|  |  |
|  | //анимация исчезновения блоков при нажатии на обычный блок |
|  | function animation\_death\_block(list\_of\_i\_2, list\_of\_j\_2, array\_i\_j\_value, counter) { |
|  | if (counter < 5) { |
|  | for (var i = 0; i < list\_of\_i\_2.length; i++) { |
|  | blocks[list\_of\_i\_2[i] - size][list\_of\_j\_2[i]].style.opacity = '0' |
|  | } |
|  | } |
|  | if (counter > 4 && counter < 7) { |
|  | blocks[list\_of\_i\_2[0] - size][list\_of\_j\_2[0]].src = '../../images/elem' |
|  | blocks[list\_of\_i\_2[0] - size][list\_of\_j\_2[0]].src += array\_i\_j\_value |
|  | blocks[list\_of\_i\_2[0] - size][list\_of\_j\_2[0]].src += '.png' |
|  | for (var i = 1; i < list\_of\_i\_2.length; i++) { |
|  | blocks[list\_of\_i\_2[i] - size][list\_of\_j\_2[i]].style.opacity = '0' |
|  | } |
|  | } |
|  | if (counter > 6 && counter < 9) { |
|  | blocks[list\_of\_i\_2[0] - size][list\_of\_j\_2[0]].src = '../../images/elem8.png' |
|  | for (var i = 1; i < list\_of\_i\_2.length; i++) { |
|  | blocks[list\_of\_i\_2[i] - size][list\_of\_j\_2[i]].style.opacity = '0' |
|  | } |
|  | } |
|  | if (counter > 8) { |
|  | out = '../../images/elem' |
|  | out += array\_i\_j\_value |
|  | out += '.png' |
|  | blocks[list\_of\_i\_2[0] - size][list\_of\_j\_2[0]].src = out |
|  | for (var i = 1; i < list\_of\_i\_2.length; i++) { |
|  | blocks[list\_of\_i\_2[i] - size][list\_of\_j\_2[i]].style.opacity = '0' |
|  | } |
|  | } |
|  | } |
|  |  |
|  | //анимация исчезновения блоков при нажатии на бонус |
|  | function bonus\_death\_animation(i, j, now) { |
|  | i = i - size |
|  | if (now == 6) { |
|  | for (var jj = 0; jj < size; jj++) { |
|  | blocks[i][jj].style.opacity = '0' |
|  | } |
|  | } |
|  | if (now == 7) { |
|  | for (var ii = 0; ii < size; ii++) { |
|  | blocks[ii][j].style.opacity = '0' |
|  | } |
|  | } |
|  | if (now == 8) { |
|  | blocks[i][j].style.opacity = '0' |
|  | if (j != 0) blocks[i][j - 1].style.opacity = '0' |
|  | if (j != size - 1) blocks[i][j + 1].style.opacity = '0' |
|  | if (i != 0) { |
|  | blocks[i - 1][j].style.opacity = '0' |
|  | if (j != 0) blocks[i - 1][j - 1].style.opacity = '0' |
|  | if (j != size - 1) blocks[i - 1][j + 1].style.opacity = '0' |
|  | } |
|  | if (i != size - 1) { |
|  | blocks[i + 1][j].style.opacity = '0' |
|  | if (j != 0) blocks[i + 1][j - 1].style.opacity = '0' |
|  | if (j != size - 1) blocks[i + 1][j + 1].style.opacity = '0' |
|  | } |
|  | } |
|  | if (now > 8) { |
|  | blocks[i][j].style.opacity = '0' |
|  | for (var ii = size; ii < size \* 2; ii++) { |
|  | for (var jj = 0; jj < size; jj++) { |
|  | if (arr[ii][jj] == now - 9) { |
|  | blocks[ii - size][jj].style.opacity = '0' |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |