

Инженерная школа информационных технологий и робототехники

Направление 09.03.01 «Информатика и вычислительная техника»

Отделение информационных технологий

Отчёт по теме

**«Разработка игрового веб-приложения»**

по дисциплине «Творческий проект»

Выполнил:

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Назаров М.В.

Проверил:

доцент отделения ИТ

оценка (до 30 б.): \_\_\_\_\_\_\_\_\_\_\_\_\_

к защите допускаю \_\_\_\_\_\_\_\_\_\_\_\_\_ Мыцко Е.А.

\_\_\_\_\_\_\_\_

Томск 2022

**Оглавление**

[Цель работы 3](#_Toc124812268)

[1. Краткое описание выполнения задач за предыдущий семестр 4](#_Toc124812269)

[2. Переработка эскиза веб-приложения 6](#_Toc124812270)

[3. Разработка алгоритма шахмат 7](#_Toc124812271)

[4. Разработка серверной части 8](#_Toc124812272)

[5. Диаграмма вариантов использования 8](#_Toc124812273)

[Заключение 9](#_Toc124812274)

[Список литературы 10](#_Toc124812275)

[Приложение 10](#_Toc124812276)

# Цель работы

Создать игровое веб-приложение настольной игры «Шахматы».

**Задание прошлого семестра**

* Выбрать фреймворк, который будет использоваться при создании приложения
* Выбрать языки разработки
* Разработать эскиз веб-страниц
* Реализовать вёрстку веб-страницы по имеющимся эскизам

**Задание на текущий семестр**

* Верстка веб-страниц
* Реализовать алгоритм игры в шашки
* Реализовать серверную часть проекта

# Краткое описание выполнения задач за предыдущий семестр

**Язык разработки**

JavaScript – это прототипно-ориентированный, мультипарадигменный язык с динамической типизацией, который поддерживает объектно-ориентированный, императивный и декларативный (например, функциональное программирование) стили программирования.

Современный JavaScript – это «безопасный» язык программирования. Он не предоставляет низкоуровневый доступ к памяти или процессору, потому что изначально был создан для браузеров, не требующих этого.

**Эскизы приложения**

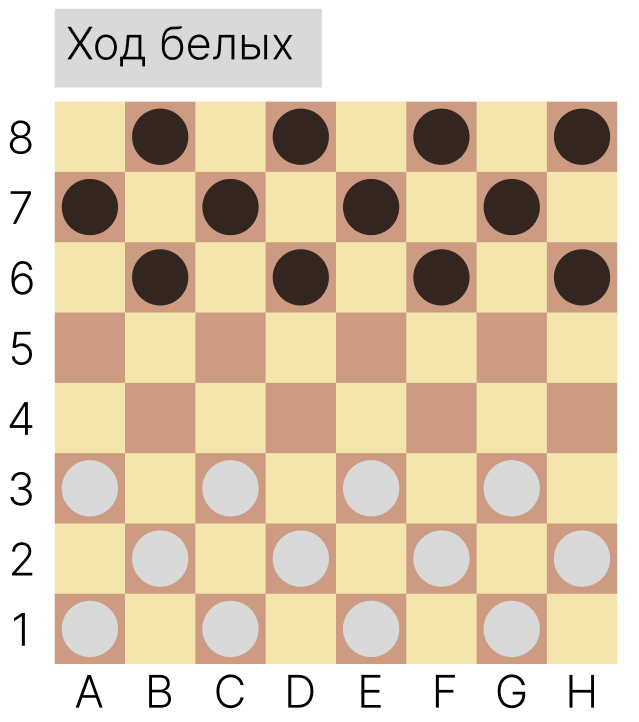


Рисунок 1: Эскиз первого варианта приложения

**Результаты верстки первого варианта веб-приложения**

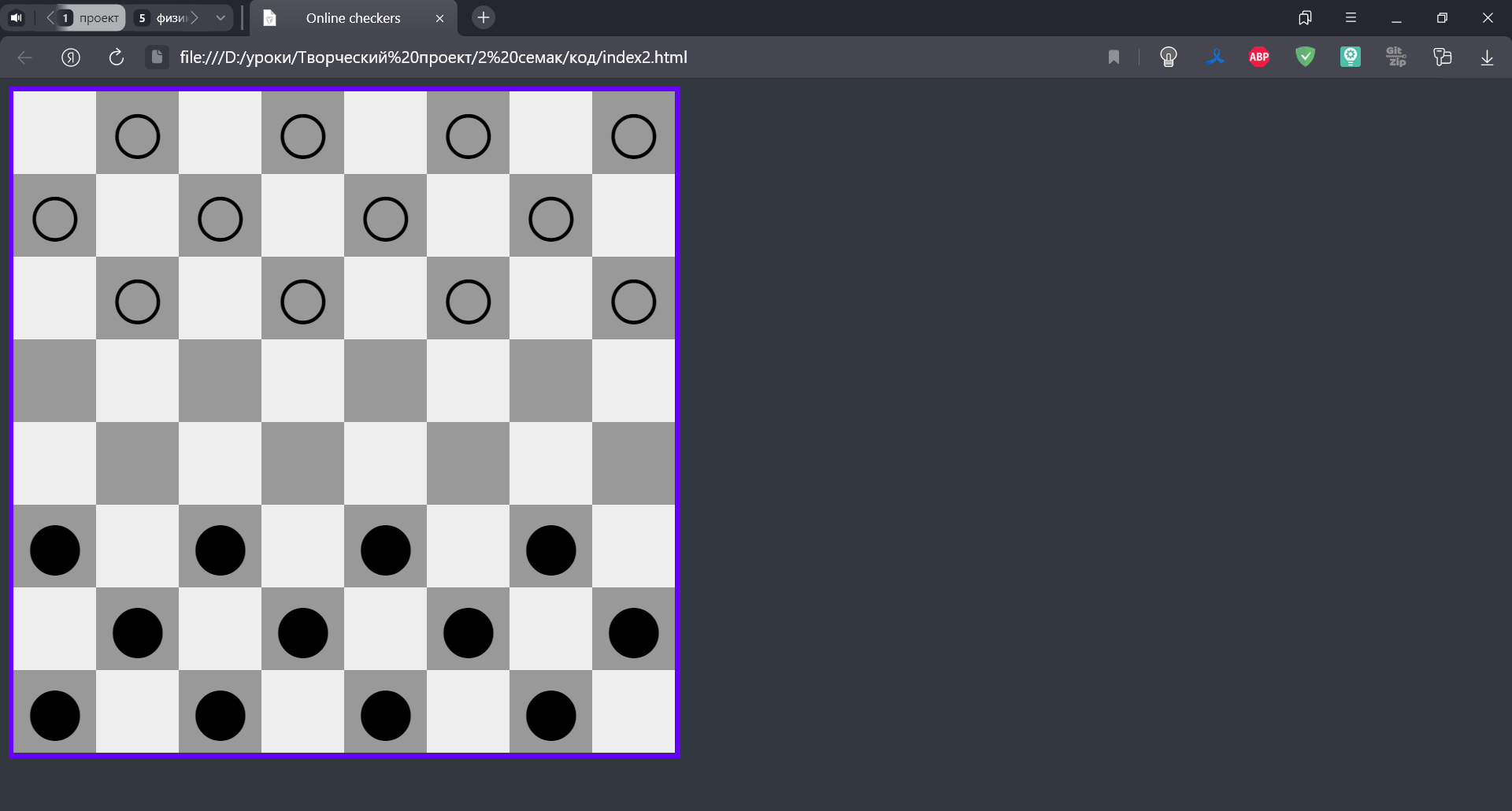


Рисунок 2:Верстка первого варианта приложения

# Доработка эскиза веб-приложения

В связи с переработкой функционала эскиз страницы был изменен

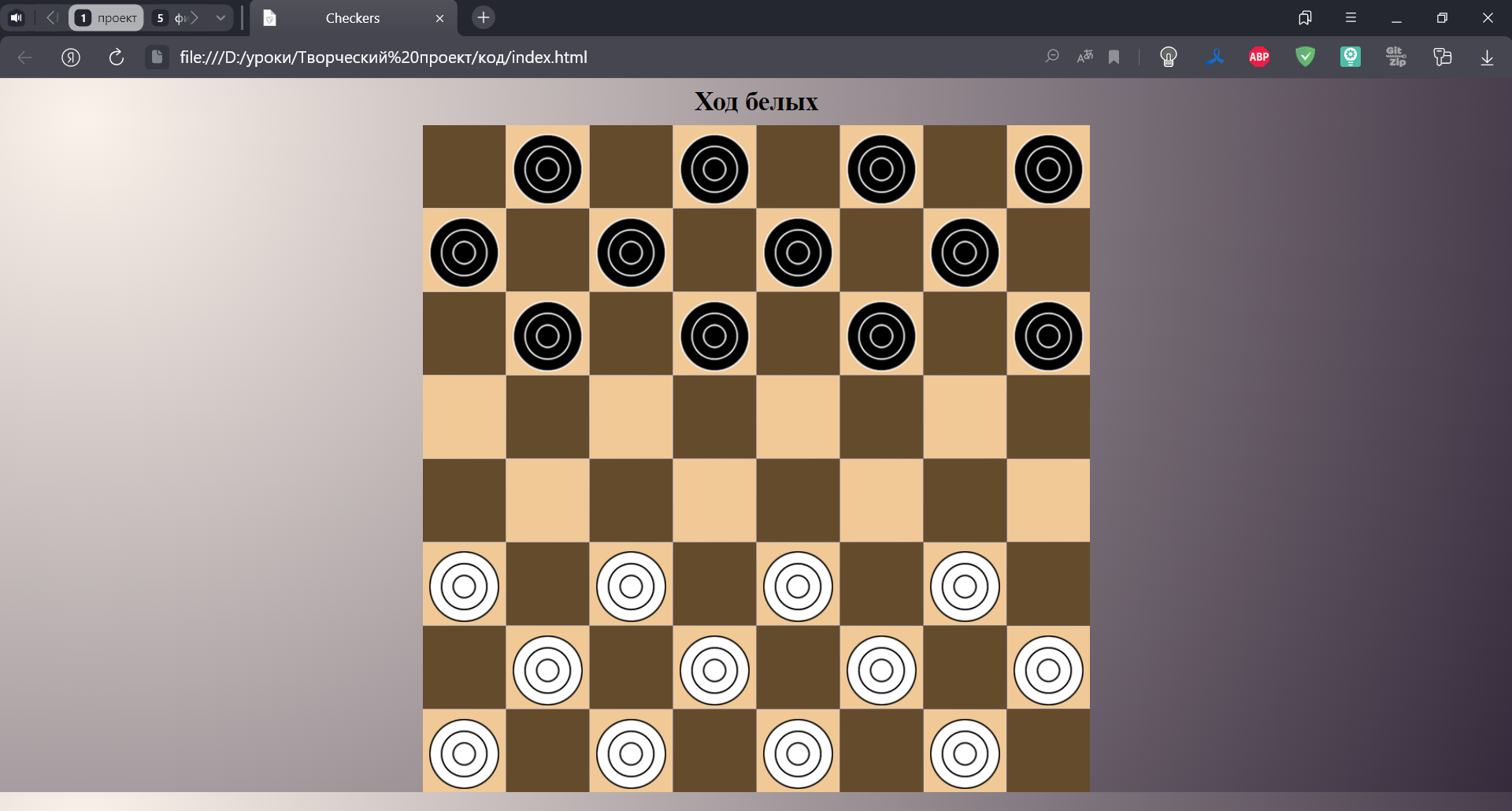


Рисунок 3: Страница с самой игровой партией

Главное изменение коснулось дизайна фигур и шахматной доски.

# Разработка алгоритма шахмат

Для разработки внутренней части работы приложения, а точнее алгоритма работы шахмат был использован JavaScript.

В данном веб-приложении шахматная доска будет представлять собой массив 8 на 8, в ячейках которого будут содержаться фигуры. Ходы для шашек расписаны в одной функции, которая перемещает их по ячейкам массива. Удаление фигур происходит через специальную функцию. Игроки ходят по очереди, передвигая фигуры нажатием по доске. Игра завершается после того, как у одного из игроков заканчиваются шашки.

# Заключение

В результате выполнения данной проектной работы, у веб-приложения появился базовый функционал: реализован алгоритм шашек, доработан эскиз сайта. Результаты верстки соответствуют эскизам.

В следующем семестре планируется:

* Реализация серверной части
* Доработать алгоритм игры
* Добавление большего функционала (истории ходов, просмотр правил)
* Дополнительная доработка дизайна приложения

# Список литературы

1. Правила игры в шахматы: полное руководство // Xchess.ru URL: https://www.wikihow.com/Play-Checkers (дата обращения: 14.01.2023)
2. Современный учебник JavaScript // URL: https://learn.javascript.ru/
3. Css учебник // URL: https://www.schoolsw3.com/css/index.php
4. HTML учебник // URL: https://www.schoolsw3.com/html/index.php

# Приложение

**checkers.html**

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="Checkers.css">

<title>Checkers</title>

<style>

body{

}

</style>

</head>

<body>

<h1 id="tog">Ход белых</h1>

<ul>

<div class="divv" id="row8">

<li class="box" id="b801"></li>

<li class="box" id="b802">Bchecker</li>

<li class="box" id="b803"></li>

<li class="box" id="b804">Bchecker</li>

<li class="box" id="b805"></li>

<li class="box" id="b806">Bchecker</li>

<li class="box" id="b807"></li>

<li class="box" id="b808">Bchecker</li>

</div>

<div class="divv" id="row7">

<li class="box" id="b701">Bchecker</li>

<li class="box" id="b702"></li>

<li class="box" id="b703">Bchecker</li>

<li class="box" id="b704"></li>

<li class="box" id="b705">Bchecker</li>

<li class="box" id="b706"></li>

<li class="box" id="b707">Bchecker</li>

<li class="box" id="b708"></li>

</div>

<div class="divv" id="row6">

<li class="box" id="b601"></li>

<li class="box" id="b602">Bchecker</li>

<li class="box" id="b603"></li>

<li class="box" id="b604">Bchecker</li>

<li class="box" id="b605"></li>

<li class="box" id="b606">Bchecker</li>

<li class="box" id="b607"></li>

<li class="box" id="b608">Bchecker</li>

</div>

<div class="divv" id="row5">

<li class="box" id="b501"></li>

<li class="box" id="b502"></li>

<li class="box" id="b503"></li>

<li class="box" id="b504"></li>

<li class="box" id="b505"></li>

<li class="box" id="b506"></li>

<li class="box" id="b507"></li>

<li class="box" id="b508"></li>

</div>

<div class="divv" id="row4">

<li class="box" id="b401"></li>

<li class="box" id="b402"></li>

<li class="box" id="b403"></li>

<li class="box" id="b404"></li>

<li class="box" id="b405"></li>

<li class="box" id="b406"></li>

<li class="box" id="b407"></li>

<li class="box" id="b408"></li>

</div>

<div class="divv" id="row3">

<li class="box" id="b301">Wchecker</li>

<li class="box" id="b302"></li>

<li class="box" id="b303">Wchecker</li>

<li class="box" id="b304"></li>

<li class="box" id="b305">Wchecker</li>

<li class="box" id="b306"></li>

<li class="box" id="b307">Wchecker</li>

<li class="box" id="b308"></li>

</div>

<div class="divv" id="row2">

<li class="box" id="b201"></li>

<li class="box" id="b202">Wchecker</li>

<li class="box" id="b203"></li>

<li class="box" id="b204">Wchecker</li>

<li class="box" id="b205"></li>

<li class="box" id="b206">Wchecker</li>

<li class="box" id="b207"></li>

<li class="box" id="b208">Wchecker</li>

</div>

<div class="divv" id="row1">

<li class="box" id="b101">Wchecker</li>

<li class="box" id="b102"></li>

<li class="box" id="b103">Wchecker</li>

<li class="box" id="b104"></li>

<li class="box" id="b105">Wchecker</li>

<li class="box" id="b106"></li>

<li class="box" id="b107">Wchecker</li>

<li class="box" id="b108"></li>

</div>

</ul>

<script src="Checkers.js"></script>

</body>

</html>

**checkers.css**

\* {

margin: 0;

padding: 0;

}

body {

height: 100%;

display: grid;

place-items: center;

overflow-x: hidden;

background: radial-gradient(circle farthest-corner at 100px 50px, #FBF2EB, #352A3B);

}

h1 {

margin: 10px;

}

ul {

display: grid;

grid-template-columns: 1fr;

}

.divv {

display: flex;

}

li {

list-style: none;

display: grid;

place-items: center;

font-size: 0px;

}

.box {

margin: .5px;

width: 100px;

height: 100px;

}

#tog {

margin: 10px 0px;

}

**checkers.js**

//Функция для подставления картинки

function insertImage() {

document.querySelectorAll('.box').forEach(image => { //перебираем все поле

if (image.innerText.length !== 0) { //если в ячейка есть текст загружает картинку

image.innerHTML = `${image.innerText} <img src="${image.innerText}.png" alt="">`

image.style.cursor = 'pointer'

}

})

}

insertImage()

darkbox = 'rgb(100, 75, 43)'

lightbox = 'rgb(240, 201, 150)'

//Функция для раскрашивания поля

function coloring() {

const color = document.querySelectorAll('.box')

color.forEach(color => {

getId = color.id //(b801)

arr = Array.from(getId) // ["b", "8", "0", "1"]

arr.shift() //удаляет первый символ массива ["8", "0", "1"]

aside = eval(arr.pop()) //удаляем последний символ и выводим его (номер столбца 1)

aup = eval(arr.shift()) //удаляем последний символ и выводим его (номер строки 8)

a = aside + aup //(81)

if (a % 2 == 0) {

color.style.backgroundColor = lightbox //четный -> светлый цвет

}

else {

color.style.backgroundColor = darkbox //нечетный -> теменый цвет

}

})

}

coloring()

//Функция для раскраски ходов

function reddish() {

document.querySelectorAll('.box').forEach(i1 => { //перебираем все поле

if (i1.style.backgroundColor == 'pink') {

document.querySelectorAll('.box').forEach(i2 => {

if (i2.style.backgroundColor == 'green' && i2.innerText.length !== 0) {

greenText = i2.innerText //текст ячейки(Bchecker или Wchecker)

pinkText = i1.innerText

greenColor = ((Array.from(greenText)).shift()).toString()

pinkColor = ((Array.from(pinkText)).shift()).toString()

getId = i2.id //(b801)

arr = Array.from(getId) // ["b", "8", "0", "1"]

arr.shift() //удаляет первый символ массива ["8", "0", "1"]

aside = eval(arr.pop()) //удаляем последний символ и выводим его (номер столбца 1)

aup = eval(arr.shift()) //удаляем последний символ и выводим его (номер строки 8)

a = aside + aup //(81)

if (a % 2 == 0 && pinkColor == greenColor) { ////если ячейка четная и выбранная попдатет на союзную цвет - светлый

i2.style.backgroundColor = lightbox

}

if (a % 2 !== 0 && pinkColor == greenColor) { //если ячейка нечетная и выбранная попдатет на союзную цвет - темный

i2.style.backgroundColor = darkbox

}

}

})

}

})

}

tog = 1

document.querySelectorAll('.box').forEach(item => {

item.addEventListener('click', function () {

//Для замены ячейки

if (item.style.backgroundColor == 'green' /\*&& item.innerText.length == 0\*/) { //если фон зеленый и в ячейке пусто

tog = tog + 1

document.getElementById(redId).innerText = ''

}

getId = item.id //(b801)

arr = Array.from(getId) //["b", "8", "0", "1"]

arr.shift() //удаляет первый символ массива ["8", "0", "1"]

aside = eval(arr.pop()) //удаляем последний символ и выводим его (номер столбца 1)

arr.push('0') //добавляем символ в конец массива ["8", "0", "0"]

aup = eval(arr.join('')) //объединяем массив в строку (800)

a = aside + aup //(801)

// Функция для отображения доступного пути

function whosTurn(toggle) {

//Шашка

if (item.innerText == `${toggle}checker`) {

//Для черных

if (tog % 2 == 0 && aup > 100) {

redId = ""

if (aside < 8 && document.getElementById(`b${a - 100 + 1}`).innerText.length == 0) { //вниз и вправо(для пустой ячейки) нужно дописать условие проверки на красную клетку

document.getElementById(`b${a - 100 + 1}`).style.backgroundColor = 'green'

}

if (aside < 7 && document.getElementById(`b${a - 100 + 1}`).innerText !== 'Bchecker' && document.getElementById(`b${a - 100 + 1}`).innerText.length !== 0 && document.getElementById(`b${a - 200 + 2}`).innerText.length == 0){ //рубить вниз и вправо

redId = (`b${a - 100 + 1}`)

document.getElementById(`b${a - 100 + 1}`).style.backgroundColor = 'red'

document.getElementById(`b${a - 200 + 2}`).style.backgroundColor = 'green'

}

if(aside > 1 && document.getElementById(`b${a - 100 - 1}`).innerText.length == 0) { //вниз и влево(для пустой ячейки) нужно дописать условие проверки на красную клетку

document.getElementById(`b${a - 100 - 1}`).style.backgroundColor = 'green'

}

if (aside > 2 && document.getElementById(`b${a - 100 - 1}`).innerText !== 'Bchecker' && document.getElementById(`b${a - 100 - 1}`).innerText.length !== 0 && document.getElementById(`b${a - 200 - 2}`).innerText.length == 0){ //рубить вниз и влево

redId = (`b${a - 100 - 1}`)

document.getElementById(`b${a - 100 - 1}`).style.backgroundColor = 'red'

document.getElementById(`b${a - 200 - 2}`).style.backgroundColor = 'green'

}

}

//Для белых

if (tog % 2 !== 0 && aup < 800) {

redId = ""

if (aside < 8 && document.getElementById(`b${a + 100 + 1}`).innerText.length == 0) { //вверх и вправо(для пустой ячейки)

document.getElementById(`b${a + 100 + 1}`).style.backgroundColor = 'green'

}

if (aside < 7 && document.getElementById(`b${a + 100 + 1}`).innerText !== 'Wchecker' && document.getElementById(`b${a + 100 + 1}`).innerText.length !== 0 && document.getElementById(`b${a + 200 + 2}`).innerText.length == 0){ //рубить вверх и вправо

redId = (`b${a + 100 + 1}`)

document.getElementById(`b${a + 100 + 1}`).style.backgroundColor = 'red'

document.getElementById(`b${a + 200 + 2}`).style.backgroundColor = 'green'

}

if (aside > 1 && document.getElementById(`b${a + 100 - 1}`).innerText.length == 0) { //вверх и влево(для пустой ячейки)

document.getElementById(`b${a + 100 - 1}`).style.backgroundColor = 'green'

}

if (aside > 2 && document.getElementById(`b${a + 100 - 1}`).innerText !== 'Wchecker' && document.getElementById(`b${a + 100 - 1}`).innerText.length !== 0 && document.getElementById(`b${a + 200 - 2}`).innerText.length == 0){ //рубить вверх и влево

redId = (`b${a + 100 - 1}`)

document.getElementById(`b${a + 100 - 1}`).style.backgroundColor = 'red'

document.getElementById(`b${a + 200 - 2}`).style.backgroundColor = 'green'

}

}

item.style.backgroundColor = 'pink'

}

}

//Смена хода

if (tog % 2 !== 0) {

document.getElementById('tog').innerText = "Ход белых"

whosTurn('W')

}

if (tog % 2 == 0) {

document.getElementById('tog').innerText = "Ход черных"

whosTurn('B')

}

reddish()

//Победа

numW = 0

numB = 0

document.querySelectorAll('.box').forEach(win => {

if (win.innerText == 'Wchecker') {

numW += 1

}

})

document.querySelectorAll('.box').forEach(win => {

if (win.innerText == 'Bchecker') {

numB += 1

}

})

if (numB == 0 || numW == 0) {

setTimeout(() => {

if (numW > 1) {

alert('Белые победили!')

location.reload()

}

else if (numB > 1) {

alert('Черные победили!')

location.reload()

}

}, 100)

}

})

})

//Передвижение шашек

document.querySelectorAll('.box').forEach(hathiTest => {

hathiTest.addEventListener('click', function () {

if (hathiTest.style.backgroundColor == 'pink') {

pinkId = hathiTest.id

pinkText = hathiTest.innerText

document.querySelectorAll('.box').forEach(hathiTest2 => {

hathiTest2.addEventListener('click', function () {

if (hathiTest2.style.backgroundColor == 'green' && hathiTest2.innerText.length == 0) {

document.getElementById(pinkId).innerText = ''

hathiTest2.innerText = pinkText

coloring()

insertImage()

}

})

})

}

})

})

//Отключение выбора нескольких шашек

z = 0

document.querySelectorAll('.box').forEach(ee => {

ee.addEventListener('click', function () {

z = z + 1

if (z % 2 == 0 && ee.style.backgroundColor !== 'green') {

coloring()

}

})

})