Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Брянский государственный инженерно-технологический университет»

Кафедра «Информационные технологии»

**Лабораторная работа №5**

по дисциплине « SPA и MPA фреймворки»

**«**Цель события**»**

Выполнил ст. гр. ИСТм-101

Алешин. К.В.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Руководитель

Н.А. Афанасьева

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Брянск 2023

Оглавление

[1. Решение простых задач 3](#_Toc129643785)

[1.1. Три счетчика 3](#_Toc129643786)

[1.2. Переключатели 4](#_Toc129643787)

[1.3. Сапер 6](#_Toc129643788)

# Решение простых задач

Выполните решение 3-х задач в одном проекте. Сделайте главную страницу index.html, на которой разместите содержание проекта (названия 3-х задач). При клике на название – открывается новая страница с решением задачи. Сделайте с помощью кнопки или гиперссылки переход назад к главной странице.

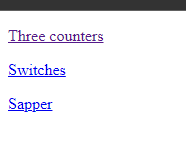


Рис. 1 index.html

Реализовано 3 гиперссылки на 3 задания.

## Три счетчика

Выполните верстку, стилизацию и работу 3-х счетчиков. Пример: 3\_счетчика.gif При нажатии на квадрат, его значение должно увеличиваться на 1. Сделайте кнопку сброса на 0 значений всех счетчиков.

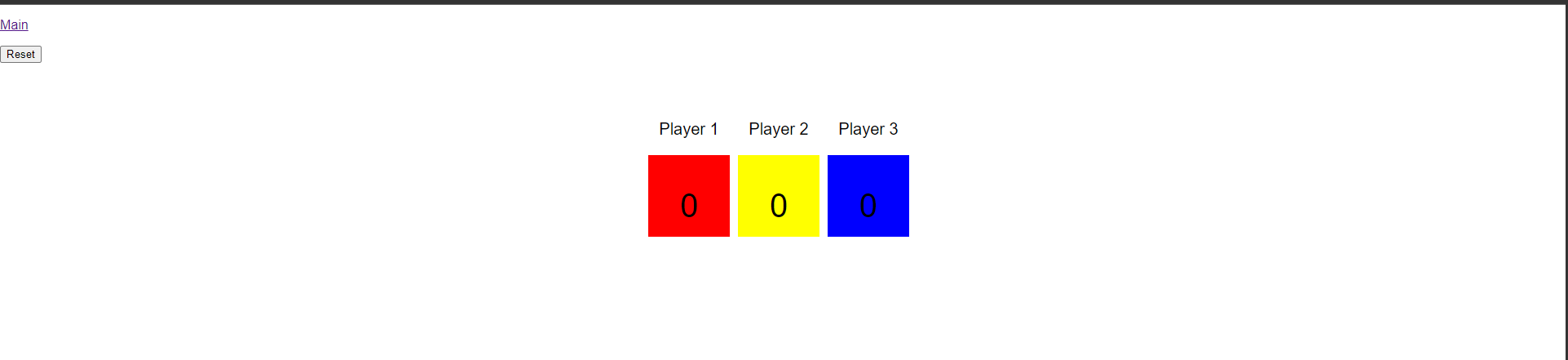


Рис. 2 Начально состояние счетчиков

|  |
| --- |
| // покрасим кнопки  for (let i = 0; i < countersNode.length; i++) {      countersNode[i].style.backgroundColor = colors[i]  } |

Листинг 1 покраска кнопок

Реализовано гиперссылка на главную форму. Так же согласно заданию добавлены 3 объекта счетчика.

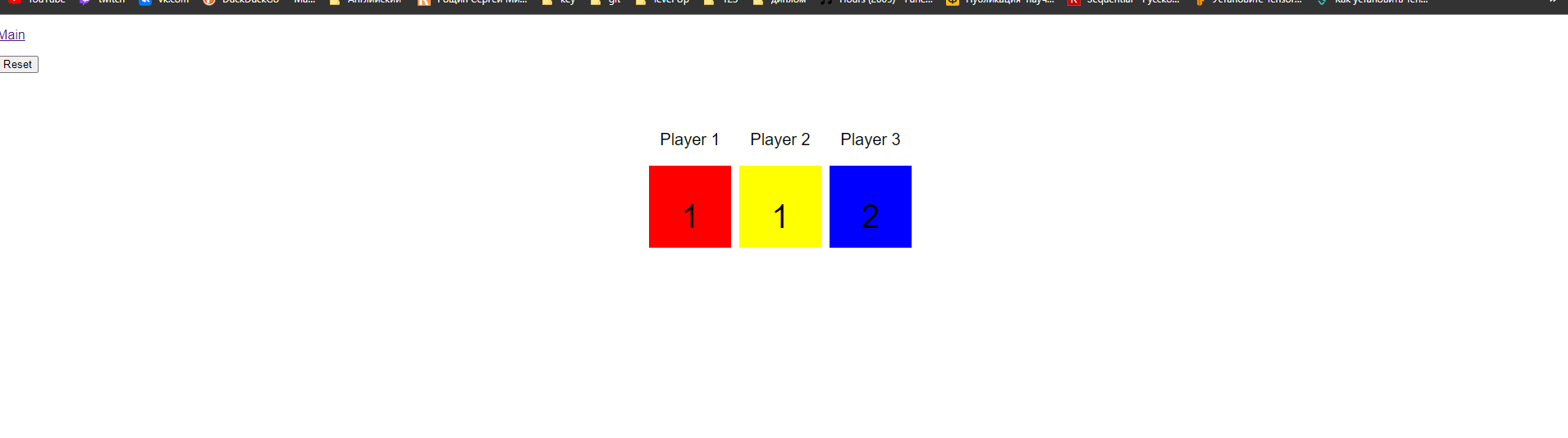


Рис. 3Измененное состояние

|  |
| --- |
| //добовляем обработчик событый  for (let i = 0; i < countersNode.length; i++) {      countersNode[i].onclick = function () {          ValueCounters[i] += 1;          countersNode[i].innerHTML = `<p>` + ValueCounters[i] + `</p>`;      }  } |

Листинг 2 Обработчик события нажатия на счетчик

Так же добавлена кнопка сброса состояния счетчика.

|  |
| --- |
| // кнопак сброса  let resetButtonNode = document.querySelector(`#reset`);  resetButtonNode.onclick = function () {      for (let i = 0; i < countersNode.length; i++) {          ValueCounters[i] = 0;          countersNode[i].innerHTML = `<p>` + ValueCounters[i] + `</p>`;      }  } |

Листинг 3 обработчик кнопки сброса

## Переключатели

Переключатель, toggle или switch — это очень популярный элемент в интерфейсах. Что-то похожее было в домашнем задании, помните? Но не используйте в решении задачи определение значений координат. Используйте заранее подготовленные классы. Пример: переключатели.gif Реализуйте работу 3-х переключателей на странице.

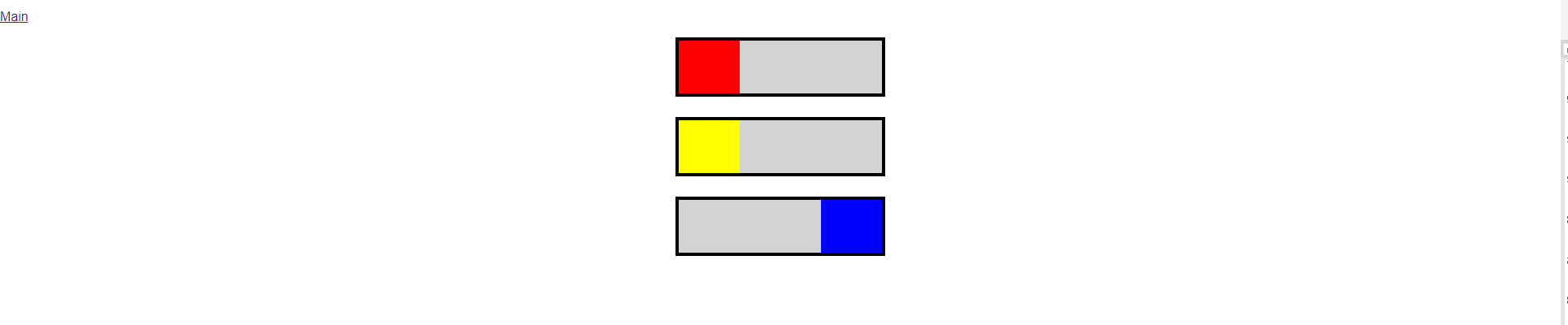


Рис. 4 Начально состояние переключателей

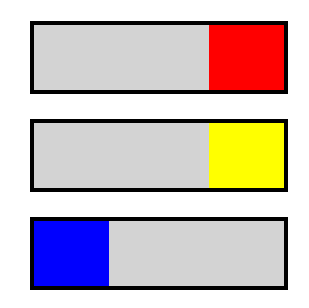


Рис. 5 Измененное состояние переключателей

|  |
| --- |
| // обработчик нажатия  for(let i=0;i<itemsSwitchNode.length;i++)  {      itemsSwitchNode[i].onclick=function()      {          if(itemsSwitchNode[i].classList.contains(`switchesOn`))          {              itemsSwitchNode[i].classList.remove(`switchesOn`);              itemsSwitchNode[i].classList.add(`switchesOff`);          }          else{              itemsSwitchNode[i].classList.remove(`switchesOff`);              itemsSwitchNode[i].classList.add(`switchesOn`);          }      }  } |

Листинг 4 Обработчик нажатия на переключатель

|  |
| --- |
| .switchesOn {      width: 75px;      height: 65px;      background-color: rgb(255, 0, 0);      display: flex;      justify-content: center;      align-items: center;      font-size: 20px;      color: white;      position: absolute;      top: 0px;      left: 0px;      transition: 1s;  }  .switchesOff {      width: 75px;      height: 65px;      background-color: rgb(255, 0, 0);      display: flex;      justify-content: center;      align-items: center;      font-size: 20px;      color: white;      position: absolute;      top: 0px;      left: 175px;      transition: 1s;  } |

Листинг 5 2 ccs стиля для переключателя

## Сапер

Реализуйте классическое и известное всем приложение – сапер! Шаблон проекта: minesweeper.zip.

Пример работы: сапер.gif

1. Зелёная клеточка – мины нет рядом

2. Оранжевая клеточка – мина на соседней клеточке

3. Тёмно-зелёная клеточка и есть мина

Добавьте проверки:

1. На одну карточку нельзя нажимать дважды.

2. Если нажали на карточку с миной, игра заканчивается проигрышем.

3. Если все карточки кроме мины открыты, игра заканчивается победой.

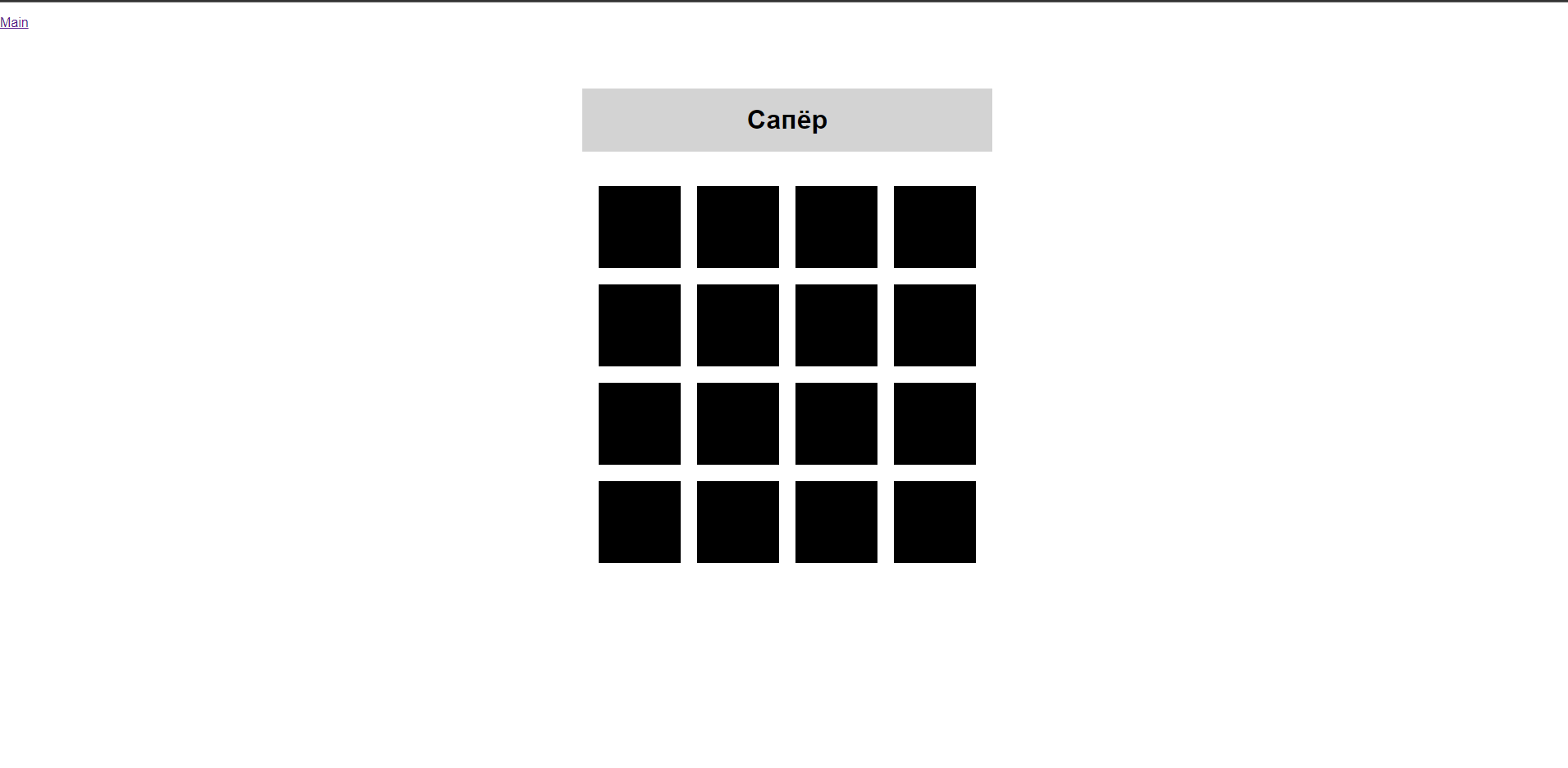


Рис. 6 Начально состояние игры сапер

Игровая логика реализована согласно заданию.

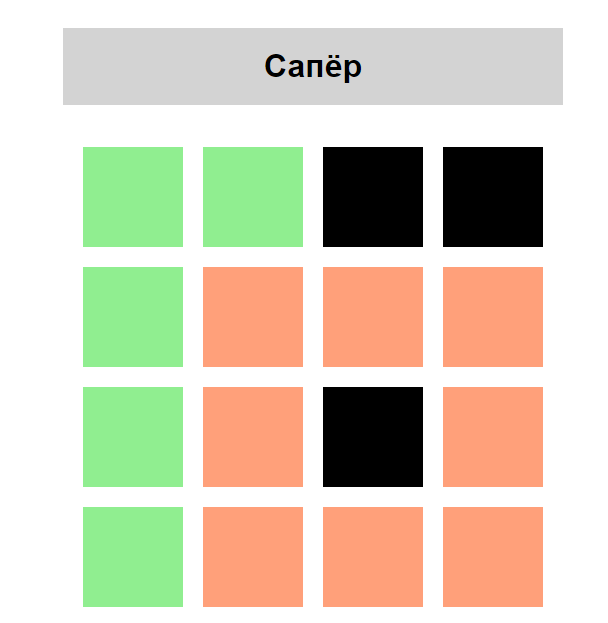


Рис. 7 Демонстрация работы

Если игрок нажмет на поле в кортом есть бобам то он проиграет.



Рис. 8 Игрок проиграл



Рис. 9 Игрок выиграл

Если игрок не разу не выберет бомбу и только бомбы останутся, то он выиграет.

|  |
| --- |
| function StatrGame() {      for (let i = 0; i < CellsNode.length; i++) {          CellsNode[i].onclick = function () {              if (CellsNode[i].classList.contains(`item\_closed`)) {                  if(!CellsNode[i].classList.contains(`item\_bomb`))                  {                      CellsNode[i].classList.remove(`item\_closed`)                      quantityClosed--;                  }                  else {                      CellsNode[i].classList.remove(`item\_closed`);                      labelNode.innerHTML = `<div class="container">                  <h1 class="gameover">Game Over</h1>`                  }              }              if (quantityClosed == quantityBombs) {                  labelNode.innerHTML = ` <div class="container">                  <h1 class="gameover">You Win</h1>`              }          }      }  } |

Листинг 6 Обработчик события нажатия на игровое поле

Основная функция логии игры.