Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный университет имени И. Н. Ульянова»

Лабораторная работа №5.

ДИНАМИЧЕСКИЕ WEB-СТРАНИЦЫ. ОБРАБОТКА СОБЫТИЙ.

**Работу выполнил:**

студент ИВТ-42-21

Михеев Артемий

**Работу проверила:**

Первова Наталья

Викторовна

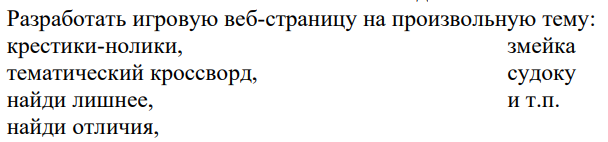
Чебоксары 2023г.

Отчет по лабораторной работе №5.

ДИНАМИЧЕСКИЕ WEB-СТРАНИЦЫ. ОБРАБОТКА СОБЫТИЙ.

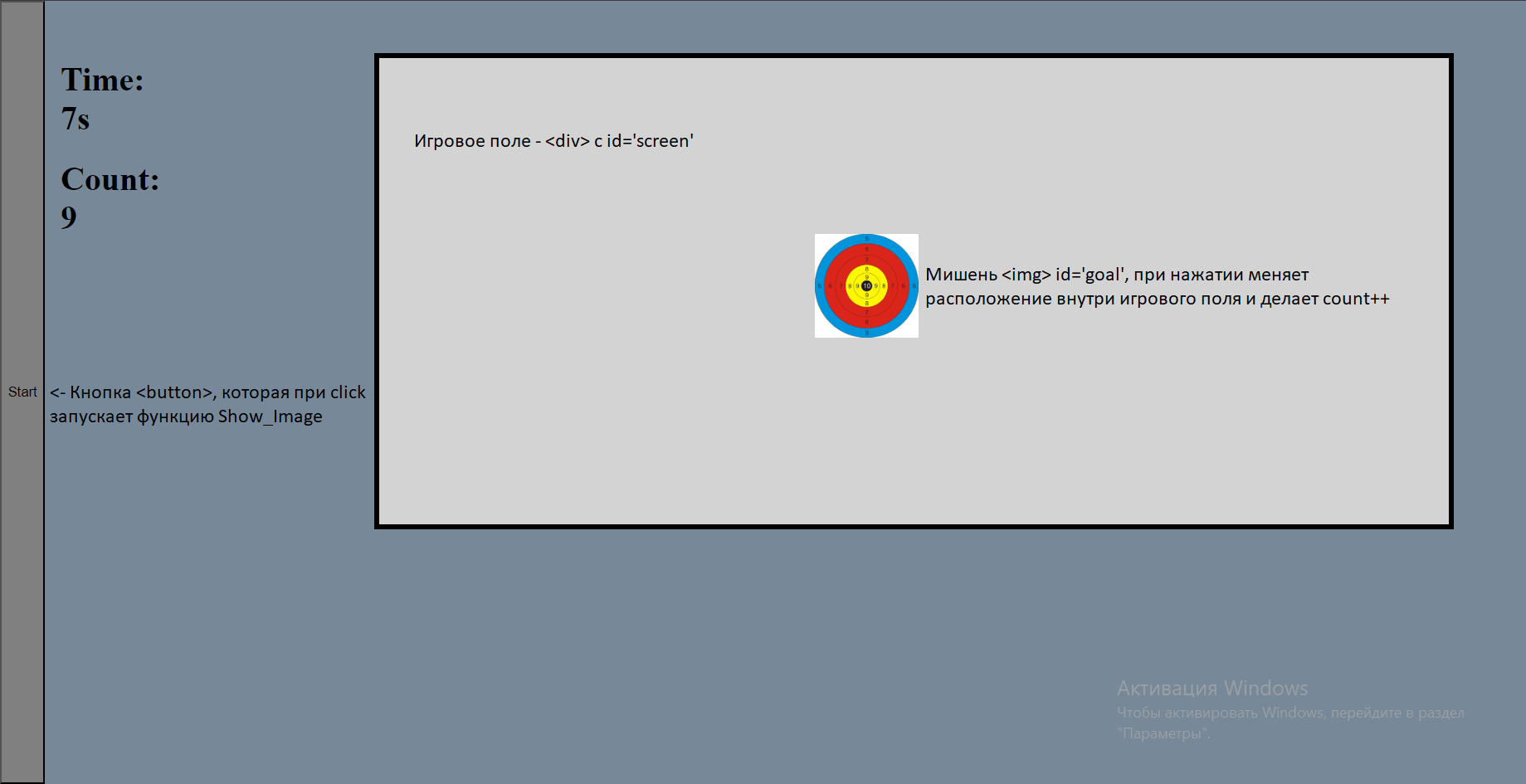
**2. Цель работы:** Изучить принципы разработки динамических web-страниц с использованием клиентских сценариев JavaScript и объектной модели документа. Научиться создавать кроссбраузерные html-страницы. Познакомиться с уровнями и особенностями API и обработкой событий DOM различных уровней.

**Задание.**



**3. Сценарий разрабатываемой игры и её интерфейс.**

Игра – Aim Trainer. Нужно нажимать на мишень, чтобы получать баллы. При каждом нажатии, мишень передвигается внутри игрового поля. Нужно набрать как можно большее количество баллов за отведенное время.

****

**5. Краткое описание игрового сценария с указанием: используемых структур данных, переменных и их назначения, перехватываемых событий и последовательности их обработки.**

При нажатии на кнопку вызывается обработчик событий, который при click вызывает функцию Show\_Image.

Функция Show\_Image меняет Dom дерево и добавляет в div картинку (мишень), затем запускается функция get\_random\_number\_and\_move, которая при нажатии на цель, двигает её в случайном направлении в пределах игрового поля и вызывает функцию Click\_count (счетчик), также в функции Show\_Image, сразу после вызова get\_random\_number\_and\_move - запускается функция Time\_to\_zero (таймер).

**6. Перечисление и описание используемых объектов, свойств, функций DOM API для реализации игры: document, document.getElementById(), document.getElementsByTagName(), document.createElement(), parentNode.appendChild(), element.textContent, element.style, element.setAttribute() и др.**

const button1 = document.querySelector('button'); – метод querySelector() возвращает первый элемент ([Element](https://developer.mozilla.org/ru/docs/Web/API/Element)) документа, который соответствует указанному селектору или группе селекторов. Если совпадений не найдено, возвращает значение null.

var goal = document.createElement("img"); – В HTML-документах создаёт элемент c тем тегом, что указан в аргументе или HTMLUnknownElement, если имя тега не распознаётся.

let container1 = document.getElementById('img\_on\_screen'); – Возвращает ссылку на элемент по его идентификатору

container1.appendChild(goal); - Node.appendChild() добавляет узел в конец списка дочерних элементов указанного родительского узла.

img.style.left = pos\_x + x + 'px'; – с помощью изменения стилей меняю расположение картинки

var pos = img.getBoundingClientRect(); - Метод Element.getBoundingClientRect() возвращает размер элемента и его позицию

cnt.innerHTML = count; - Свойство интерфейса [Element](https://developer.mozilla.org/ru/docs/Web/API/Element) innerHTML устанавливает или получает HTML или XML разметку дочерних элементов.

setTimeout(Time\_to\_zero,1000) - Вызов функции или выполнение фрагмента кода после указанной задержки.

button1.addEventListener('click', Show\_Image); - Метод EventTarget.addEventListener() регистрирует определённый обработчик события, вызванного на [EventTarget](https://developer.mozilla.org/ru/docs/Web/API/EventTarget).

В моем случае при нажатии (click) на кнопку (button), вызывается функция (Show\_Image).

**8. Вывод:** я изучил принципы разработки динамических web-страниц с использованием клиентских сценариев JavaScript и объектной модели документа. Научился создавать кроссбраузерные html-страницы. Познакомился с уровнями и особенностями API и обработкой событий DOM различных уровней.