

Создание браузерной игры на языке JavaScript для проверки на точность и реакцию

Халиманенков Андрей Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматриваются разработка игры для проверки на точность и реакцию на языке JavaScript для браузеров. В меню запускается выбранный таймер и начинается игра. После истечения таймера выводятся набранные очки.

Ключевые слова: браузерная игра, JavaScript, HTML, CSS, RESTAPI, веб-приложение.

Creating a browser game in JavaScript to test for accuracy and reaction

Khalimanenkov Andrey Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the development of a game for checking for accuracy and reaction in JavaScript for browsers. The selected timer starts in the menu and then the game starts.

Keywords: browser game, JavaScript, HTML, CSS, RESTAPI, web application.

Компьютерные игры являются отличным способом отдохнуть и переключить текущую деятельность на что-то несложное и расслабляющее. Они могут поднять соревновательный дух в перерыве от работы или помочь в обучении. Также создание игр помогает изучать основы языков программирования. Хороший способ изучить базу JavaScript – разработать игру, которая будет запущена в браузере. В качестве примера в этой статье будет написана игра, с помощью которой можно соревноваться в меткости и скорости реакции. Суть состоит в том, чтобы за отведённое количество времени, которое выбирает игрок, набрать как можно больше очков. Одно очко даётся за клик по шарикун произвольного размера и цвета в игровой области. Шарикун появляется в случайном месте этой области.

Ключевые вопросы разработки однопользовательских двумерных игр на языке JavaScript рассматриваются в учебном пособии С. А. Беляева [1]. Е. В. Пантелеева отразила сущность разработки сайта с использованием языка разметки HTML, особенности таблицы стилей CSS и языка программирования JavaScript [2]. В работе Е. В. Соловьева и Т. А.

Максимова рассматривается разработка браузерной игры на языке JavaScript [3]. В своей статье Е.Н. Поварницын рассматривает создание игры «Тетрис» на языке JavaScript, HTML и таблицей стилей CSS [4]. Nabhani S. и соавторы рассматривают сетевой прототип игры «Pharmacy Challenge» на студентах-фармацевтах для проведения исследования на оценку игры с точки зрения дизайна и оценивают влияние на успеваемость и уверенность учащихся [5].

Для создания игры использовался чистый JavaScript [6] без сторонних библиотек и фреймворков, а также язык разметки HTML5 [7] и язык каскадных стилей CSS [8].

Начальная страница имеет заголовок «Проверь свою точность и реакцию» и кнопку «Начать игру» (рис. 1). Эта кнопка имеет слушатель событий нажатия левой кнопки мыши `addEventListener`, в который передана функция по перемещению общего полотна сайта. Это код всего HTML документа, его немного, т.к. вся логика завязана на JavaScript:

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link rel="stylesheet" href="styles.css" />
    <title>Проверь свою точность и реакцию</title>
    <script src="script.js" defer></script>
  </head>
  <body>
    <div class="screen">
      <h1>Проверь свою точность и реакцию</h1>
      <button class="start">Начать игру</button>
    </div>

    <div class="screen">
      <h1>Выберите время</h1>
      <ul class="time-list">
        <li>
          <button class="time-btn">
            1 сек
          </button>
        </li>
        <li>
          <button class="time-btn">
            5 сек
          </button>
        </li>
        <li>
          <button class="time-btn">
            30 сек
          </button>
        </li>
      </ul>
    </div>

    <div class="screen">
      <h3>Осталось <span id="time">00:00</span></h3>
      <div class="board" id="board"></div>
    </div>
  </body>
</html>
```

Каждый `div` с классом `screen` имеет следующие стили:

```
.screen {
  overflow: hidden;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
  width: 100vw;
  transition: 0.5s ease-out;
  background: linear-gradient(90deg, #b5d0dc 0%, #494f5f 150%);
}
```

Это позволяет ему занимать всю область внутри браузера. В данном случае это три элемента и общая высота сайта в пикселях вычисляется следующим образом – высота внутренней области браузера для отображения сайтов умножается на три. Чтобы скрыть полосу прокрутки, нужно использовать `overflow: hidden;` в стилях `body`.

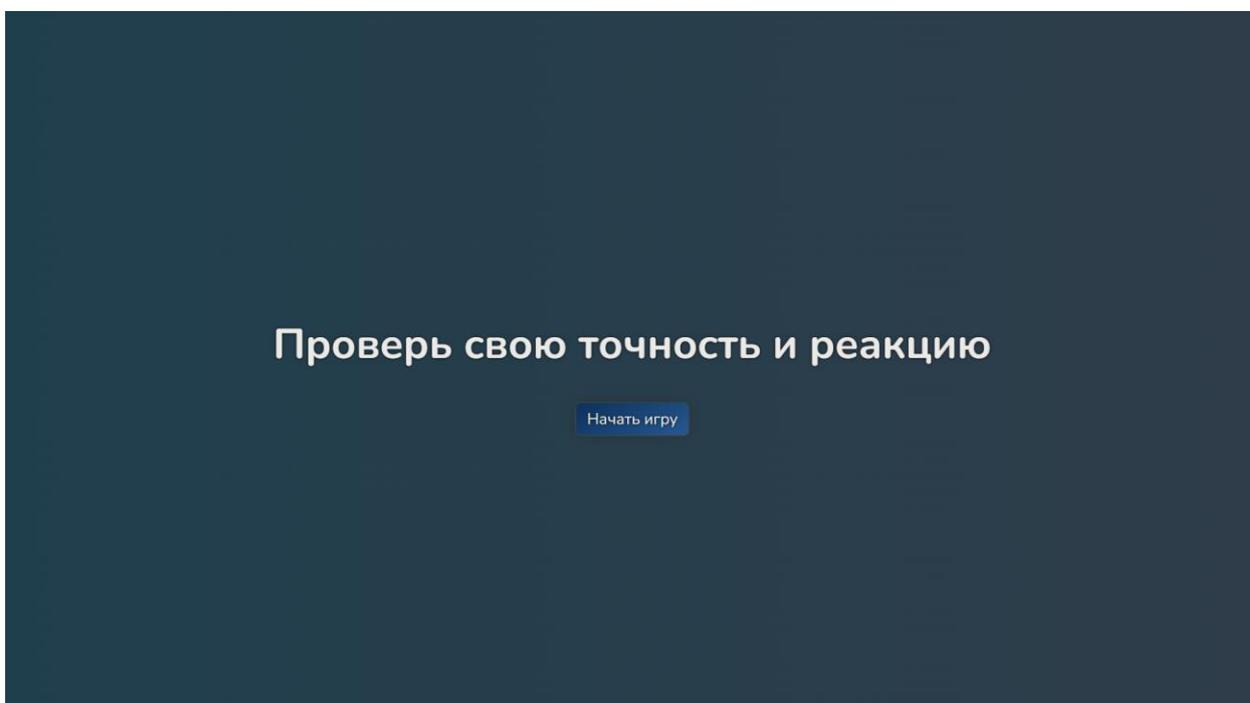


Рисунок 1 – Начальный экран

При нажатии на кнопку «Начать игру» сайт «пролистывается» до следующего экрана с таймером (рис. 2). Здесь на выбор даётся три кнопки. Они ставят таймер игры на 1, 5 и 30 секунд соответственно. После нажатия на кнопку сайт пролистывается дальше до 3 области с игрой (рис. 3) и игра начинается. Кружки генерируют случайно свои цвет, размер и положение внутри игрового экрана.

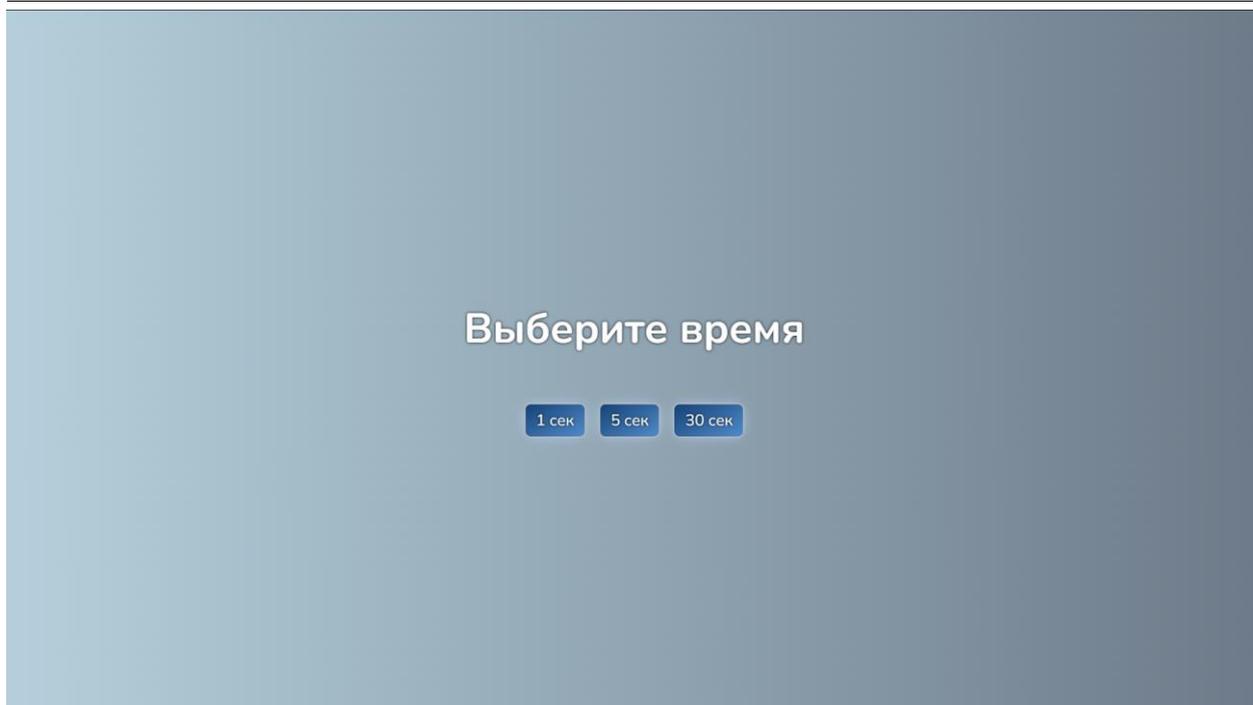


Рисунок 2 – Экран с таймером

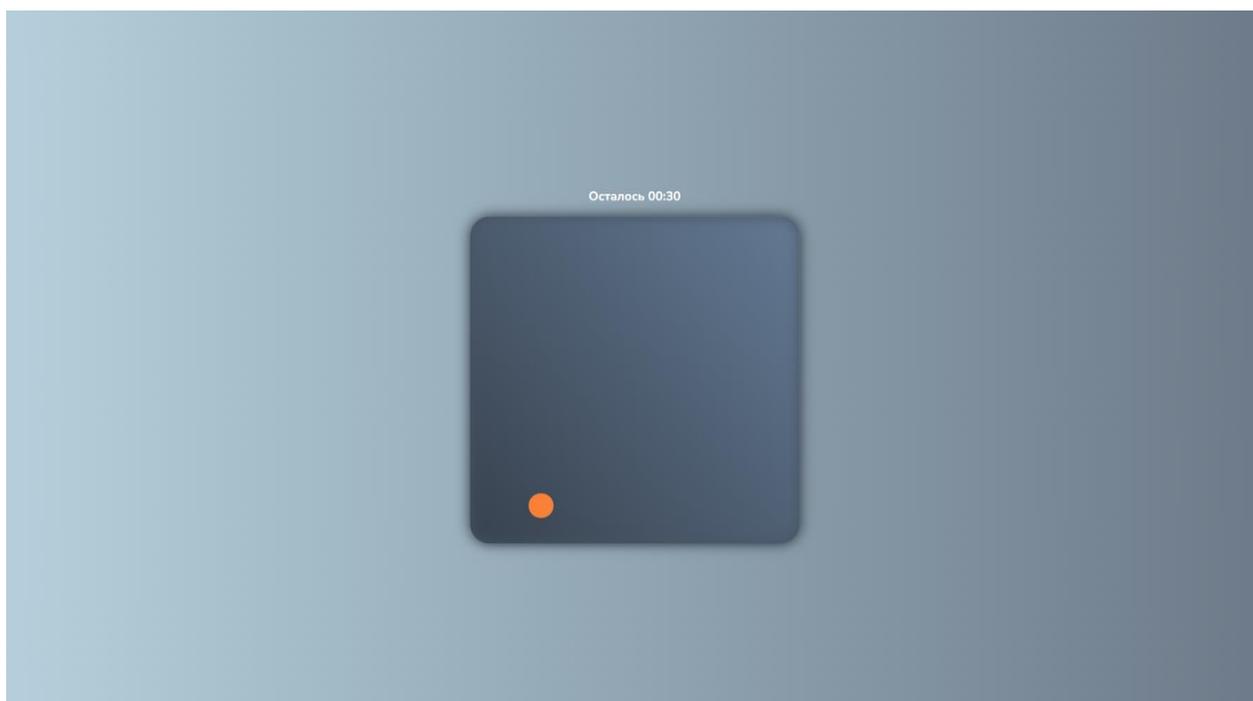


Рисунок 3 – Игровая область



Рисунок 4 – Возможные расположения, цвета и размеры кружков

Игра запускается функцией `startGame`, где запускает таймер функцией `startTimer`, в которую попадает время. Оно берётся из значения кнопки в DOM дереве и обрезается функцией строки `trim`. После этого создаётся первый кружок с помощью функции `createCircle`. Вот код игры:

```
const body = document.body;
const start = document.querySelector('.start');
const timeBtn = document.querySelectorAll('.time-btn');
const timeList = document.querySelector('.time-list');
const timeSection = document.querySelector('#time');
const board = document.querySelector('#board');

let points = 0;

start.addEventListener('click', goToTimers);
timeList.addEventListener('click', startGame);

function getRandom(min, max) {
  let rand = min + Math.random() * (max + 1 - min);
  return Math.floor(rand);
}

function getRandomColor() {
  return `hsl(${Math.random() * 360}, 95%, 60%)`;
}

function goToTimers(e) {
  e.preventDefault();
  body.style.transform = 'translateY(-100vh)';
}

function createCircle() {
  const size = getRandom(25, 70);
  const {width, height} = board.getBoundingClientRect();
  const circle = document.createElement('div');
  circle.classList.add('circle');
  board.appendChild(circle);
}
```

```
circle.style.height = `${size}px`;
circle.style.width = `${size}px`;
circle.style.background = getRandomColor();

circle.style.top = `${getRandom(0, height - size)}px`;
circle.style.left = `${getRandom(0, width - size)}px`;

circle.addEventListener('click', (e) => {
  ++points;
  e.target.remove();
  createCircle();
})
}

async function startGame(e) {
  timeSection.parentNode.classList.remove('hide');
  if (e.target.matches('.time-btn')) {
    body.style.transform = 'translateY(-200vh)';
    let time = parseInt(e.target.textContent.trim());
    createCircle();
    startTimer(time);
  }
}

function createEndScreen() {
  const endScreen = document.createElement('div');
  endScreen.classList.add('end-screen');
  endScreen.textContent = `Ваш счёт: ${points}`;

  const reloadBtn = document.createElement('button');
  reloadBtn.classList.add('reload-btn');
  reloadBtn.textContent = `Начать заного`;
  reloadBtn.addEventListener('click', () => {
    body.style.transform = 'translateY(-100vh)';
    board.textContent = '';
    endScreen.remove();
    points = 0;
  });

  board.append(endScreen);
  endScreen.append(reloadBtn);
}

function gameOver() {
  timeSection.parentNode.classList.add('hide');

  const circle = board.querySelector('.circle');
  circle.remove();

  createEndScreen();
}

function setTimer(value) {
  if (value < 10) {
    timeSection.textContent = `00:0${value}`;
  } else {
    timeSection.textContent = `00:${value}`;
  }
}

function startTimer(time) {
  setTimer(time);
  let timer = setInterval(() => {
```

```
setTimer(time);  
time--;  
if (time < 0) {  
    clearInterval(timer)  
    gameOver();  
}  
}, 1000);  
}
```

В конце игры создаётся экран с подсчитанными очками (рис. 5). При нажатии на кнопку «Начать заново» игрок вернётся к выбору времени.

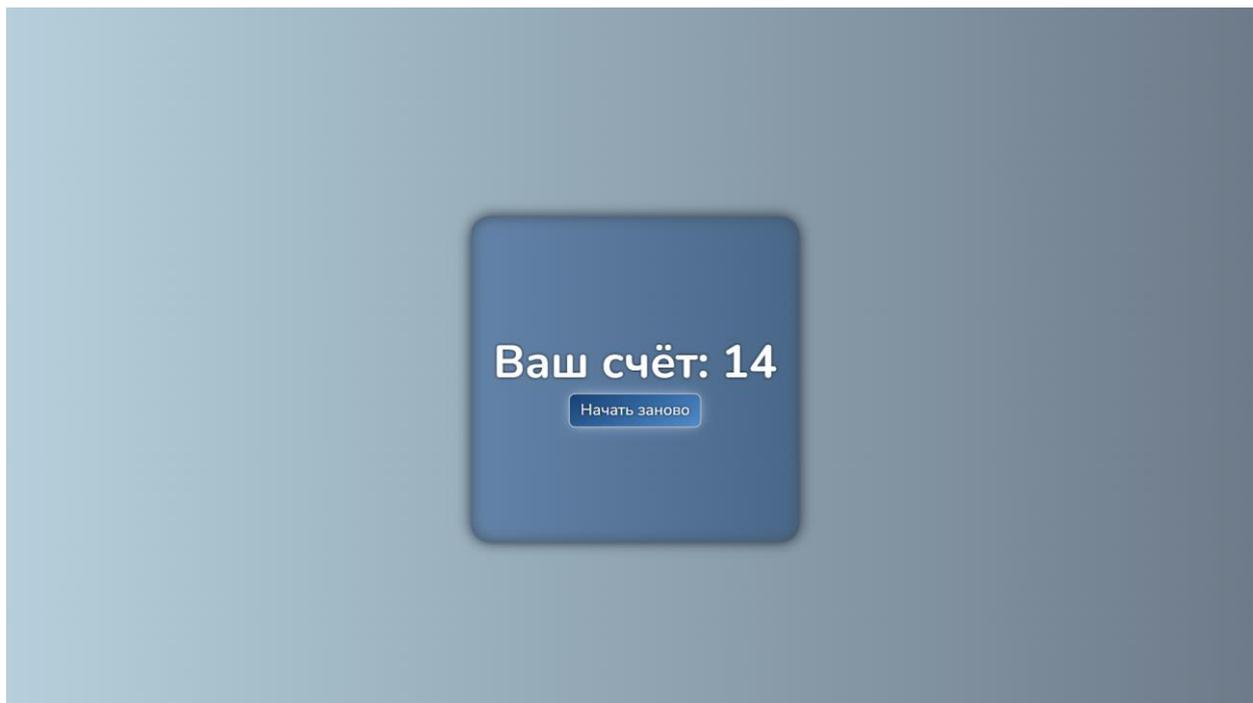


Рисунок 5 – Подсчитанные очки в конце игры

Таким образом, была разработана браузерная игра для проверки точности и скорости реакции, в которой использовались обработчики событий `addEventListener` и передача стилей CSS DOM элементам страницы через JavaScript, чтобы пролистывать области меню и игры. Проверить работоспособность игры возможно на сервисе Netlify [9], который предоставляет бесплатный хостинг для маленьких проектов.

Библиографический список

1. Беляев С.А. Разработка игр на языке JavaScript: Учебное пособие. СПб.: Лань, 2016. 128с
2. Пантелеева Е. В. Разработка сайта с использованием языка разметки HTML, таблицы стилей CSS и языка программирования JavaScript. // Информационные системы и технологии в образовании, науке и бизнесе. 2020. С. 94-96.
3. Соловьева Е. В., Максимова Т. А. Компьютерная игра-головоломка «Судoku» на языке JavaScript // Студенческие научные достижения. 2020.

С. 21-25.

4. Поварницын Е. Н. Создание браузерной игры «Тетрис» // Форум молодых ученых. 2020. №. 2. С. 289-296.
5. Nabhani S. et al. Development and evaluation of an educational game to support pharmacy students //Currents in Pharmacy Teaching and Learning. 2020. Т. 12. №. 7. С. 786-803.
6. JavaScript URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 03.01.2022).
7. HTML URL: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 03.01.2022).
8. CSS URL: <https://ru.wikipedia.org/wiki/CSS> (дата обращения: 03.01.2022).
9. Игра из статьи URL: <https://ushlerdop.github.io/Aim-game> (дата обращения: 03.01.2022).