

Разработка интерактивного веб-приложения «Три в ряд — Фруктовая Буря» с использованием HTML, CSS и JavaScript

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Статья посвящена разработке и практической реализации интерактивного веб-приложения «Три в ряд — Фруктовая Буря» — логической головоломки, в которой пользователь собирает комбинации из фруктов для получения очков и звёзд. В работе подробно описаны архитектура приложения, используемые технологии (HTML5, CSS3, JavaScript) и основные функциональные возможности: добавление и интерактивное управление фруктами на игровом поле 6×6, система целей до трёх звёзд на пяти уровнях, механика бомб при сборе пяти и более одинаковых фруктов, автоматическое обновление поля при отсутствии ходов, а также динамические визуальные эффекты и интуитивный пользовательский интерфейс. Приложение создаёт увлекательный игровой опыт и может быть использовано как образовательный инструмент для изучения основ клиентской веб-разработки или как развлекательное средство для развития логического мышления.

Ключевые слова: HTML, CSS, JavaScript, веб-приложение, игра «три в ряд», интерактивность, анимация, уровни, бомбы.

Implementation of an Interactive Web Application “Match-3: Fruit Burst” Using HTML, CSS and JavaScript

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article is devoted to the development and practical implementation of an interactive web application “Match-3: Fruit Burst” — a logic puzzle in which the user collects combinations of fruits to earn points and stars. The paper describes in detail the application architecture, technologies used (HTML5, CSS3, JavaScript), and main functional capabilities: adding and interactively managing fruits on a 6×6 game board, a three-star goal system across five levels, bomb mechanics triggered by collecting five or more identical fruits, automatic board reshuffling when no moves are available, as well as dynamic visual effects and an intuitive user interface. The application provides an engaging gameplay experience and can be used as an educational tool for learning the fundamentals of client-side web development or as an entertainment medium for developing logical thinking.

Keywords: HTML, CSS, JavaScript, web application, match-3 game, interactivity, animation, levels, bombs

1 Введение

1.1 Актуальность

В современной цифровой среде всё большее значение приобретают интерактивные веб-приложения, способные не только решать прикладные задачи, но и обеспечивать вовлекающий пользовательский опыт через игровые механики. Жанр «три в ряд» остаётся одним из самых популярных в мобильной и браузерной разработке благодаря своей простоте, доступности и высокой степени вовлечённости. Создание подобных игр с использованием нативных веб-технологий (HTML, CSS, JavaScript) позволяет демонстрировать широкие возможности клиентской разработки без привлечения сторонних библиотек и фреймворков.

Актуальность темы обусловлена растущим интересом к образовательным играм и потребностью в лёгких, кроссплатформенных приложениях, сочетающих логическую составляющую с визуальной привлекательностью. Такие проекты могут использоваться как в учебном процессе для освоения основ программирования, так и в качестве самостоятельного развлекательного контента.

1.2 Обзор исследований

И. В. Смирнов в своей статье рассмотрел применение HTML5 и CSS3 для создания анимированных веб-интерфейсов, подчеркнув важность использования ключевых кадров для реализации сложных анимаций [1]. А. С. Петрова провела исследование, посвященное разработке интерактивных игр на JavaScript, где подробно описала методы управления событиями мыши для реализации функций перетаскивания объектов [2]. Д. А. Кузнецов представил программную реализацию системы управления мультимедийным контентом в браузере, включая воспроизведение аудио и управление громкостью с помощью элементов управления, что является важным аспектом для создания атмосферных приложений [3]. Е. М. Григорьева и М. П. Лебедев в своей работе описали подходы к созданию адаптивных веб-приложений с использованием Flexbox и Grid, включая панели управления и игровые поля [4]. П. Н. Тихонов представил исследование по оптимизации производительности веб-анимаций, где рекомендовал использовать requestAnimationFrame [5].

1.3 Цель исследования

Цель исследования – разработка интерактивного веб-приложения «Три в ряд — Фруктовая Буря» на основе технологий HTML, CSS и JavaScript.

2 Материалы и методы

В данном исследовании для разработки интерактивного веб-приложения использовалась среда разработки Visual Studio Code (VS Code), которая

предоставляет мощные инструменты для написания, отладки и тестирования кода на языках HTML, CSS и JavaScript. Выбор этих технологий обусловлен их универсальностью, поддержкой всеми современными браузерами и возможностью создания богатых пользовательских интерфейсов без необходимости установки дополнительных плагинов.

Для создания визуального оформления приложения были использованы стандартные эмодзи-символы, что позволило отказаться от внешних графических файлов и обеспечить кроссплатформенную совместимость.

Основным методом исследования является практическая разработка приложения с использованием следующих технологий:

- HTML5: для структурирования содержимого страницы, создания игрового поля и элементов управления.
- CSS3: для оформления интерфейса, задания стилей, анимаций (сбор комбинаций, пульсация бомб) и управления состоянием (подсветка заголовков, градиентный фон).
- JavaScript: для реализации логики приложения, включая обработку кликов, проверку комбинаций, начисление очков, управление таймером и автоматическое обновление поля.

3 Результаты

Разработанное интерактивное веб-приложение «Три в ряд — Фруктовая Буря» представляет собой полноценную браузерную головоломку, реализованную без использования сторонних библиотек и фреймворков. Приложение полностью автономно: оно запускается в любом современном браузере и не требует подключения к интернету после загрузки HTML-файла [6]. Основная цель игры — собирать комбинации из трёх и более одинаковых фруктов, расположенных в горизонтальном или вертикальном ряду, для получения очков и достижения целей уровня.

3.1 Архитектура приложения

Приложение построено по классическому принципу разделения ответственности:

- HTML5 отвечает за структуру документа — определяет игровое поле (сетка 6×6), панели управления, экраны (приветственный, игровой, результатов);
- CSS3 управляет визуальным оформлением — градиентный фон (linear-gradient (135deg, #a1c4fd, #c2e9fb)), стили ячеек, анимации сбора и падения, подсветка заголовков;
- JavaScript реализует всю игровую логику: обработка кликов, проверка комбинаций, начисление очков, управление таймером, генерация бонусов и автоматическое обновление поля.

Все экраны приложения центрированы с помощью flexbox и ограничены максимальной шириной 600px, что обеспечивает комфортное использование как на ПК, так и на мобильных устройствах. Единый цветовой стиль

сохраняется на всех этапах игры, что создаёт ощущение целостности и профессионального дизайна.

Приветственное окно (рис. 1) содержит краткое, но информативное описание игры, подчёркивающее ключевые особенности: 5 уровней, система звёзд, бомбы и необходимость стратегического мышления. Кнопка «Начать игру» запускает первый уровень.

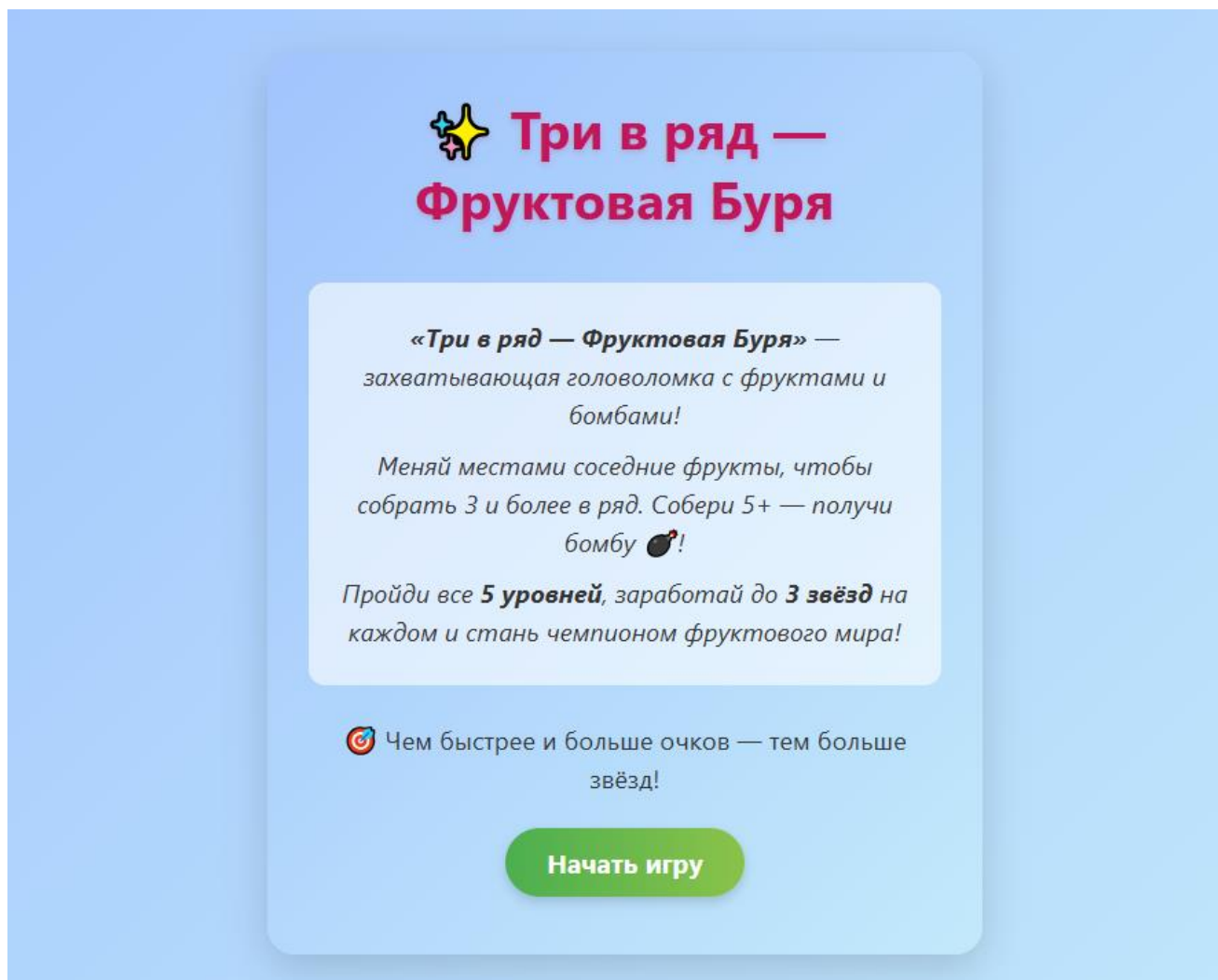


Рисунок 1 – Приветственное окно приложения

Игровой экран (рис. 2) включает:

- 1) панель счёта и уровня;
- 2) заголовок уровня с подсветкой (например, «Уровень 3: Цитрусовый шторм»);
- 3) информационную строку с целями для 1★, 2★ и 3★;
- 4) визуальную шкалу прогресса, заполняющуюся в зависимости от набранных очков;
- 5) таймер, отсчитывающий время до конца уровня;
- 6) игровое поле 6×6 с фруктами и бомбами;
- 7) невидимый блок для сообщений (используется только при отсутствии ходов).



Рисунок 2 – Игровой экран приложения

Экран результата появляется после завершения уровня или окончания времени (рис. 3). Он отображает:

- итоговый счёт;
- бонус за оставшееся время (1 очко за секунду);
- количество заработанных звёзд (от 0 до 3);
- кнопки «Следующий уровень» и «Играть сначала».

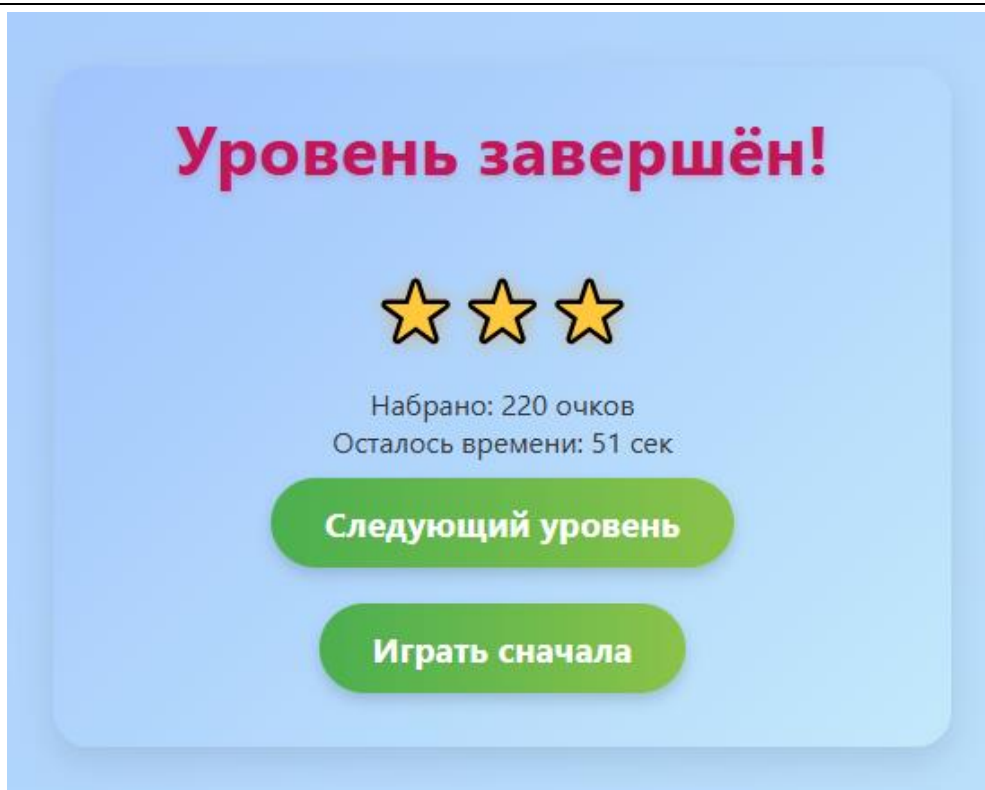


Рисунок 3 – Экран результата после прохождения уровня

Если игрок проходит пятый (финальный) уровень, дополнительно появляется сообщение « ФИНАЛЬНАЯ ПОБЕДА!», а вместо кнопки «Следующий уровень» — «В главное меню», сигнализируя об окончании игры (рис. 4).

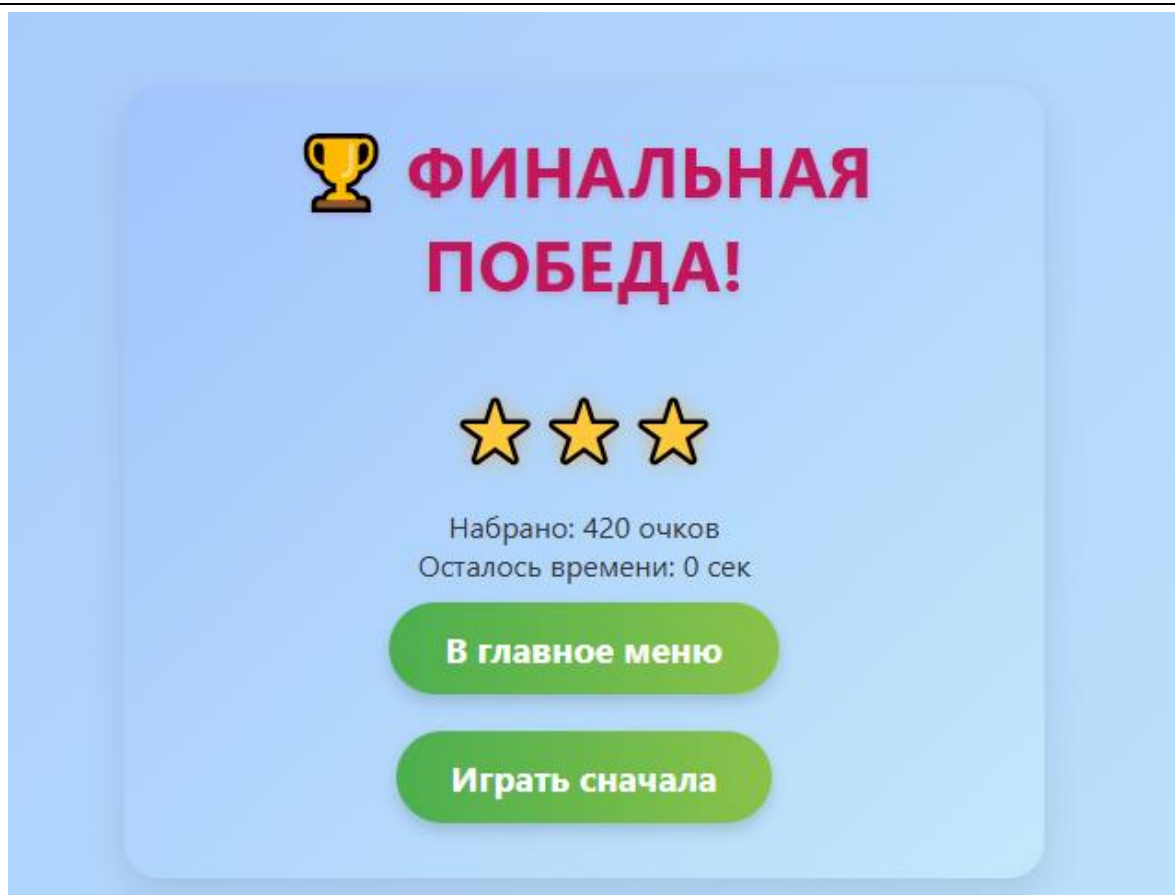


Рисунок 4 – Экран результата после прохождения последнего уровня

3.2 Основные функции и их реализация

Инициализация уровня и автоматический сбор стартовых комбинаций

При запуске любого из пяти уровней вызывается функция `startLevel(level)`. Она сбрасывает счёт и таймер, генерирует новое игровое поле с помощью `generateCleanBoard()`, и сразу проверяет наличие готовых комбинаций. Если таковые присутствуют, они мгновенно собираются, а очки начисляются игроку без его участия:

```
function startLevel(level) {
  currentLevel = level;
  const cfg = levelConfig[level - 1];
  score = 0;
  timeLeft = cfg.time;
  isProcessing = false;
  selectedCell = null;
  welcomeScreen.style.display = 'none';
  gameScreen.style.display = 'flex';
  resultScreen.style.display = 'none';
  levelDisplayEl.textContent = `Уровень ${level}: ${cfg.title}`;
  updateStats();
  startTimer();
  generateCleanBoard();
  renderBoard();
  setTimeout(async () => {
    let matches = findMatches();
```

```
while (matches.length > 0) {
  await processMatches(matches);
  matches = findMatches();}
checkForMoves();}, 300); }
```

Это решение устраняет ситуацию, когда игрок оказывается перед полем, в котором уже есть комбинации — игра сразу переходит в активное состояние.

Система уровней и звёзд

Приложение содержит 5 уровней с возрастающей сложностью (рис. 5).

Уровень	Название	Цель (1★)	Время	2★ (130%)	3★ (170%)
1	Фруктовый старт	120	60 с	156	204
2	Сладкий вызов	180	55 с	234	306
3	Цитрусовый шторм	240	50 с	312	408
4	Ягодный апокалипсис	300	45 с	390	510
5	Фруктовый чемпион	360	40 с	468	612

Рисунок 5 – Таблица уровней

Если игрок набирает $\geq 170\%$ цели, уровень завершается мгновенно с отображением экрана победы:

```
const cfg = levelConfig[currentLevel - 1];
if (score >= cfg.target * 1.7) {
  showResult();
  return; }
```

Функция showResult() определяет, является ли текущий уровень последним:

```
function showResult() {
  clearInterval(timerInterval);
  const remainingTime = Math.max(0, timeLeft);
  finalScoreEl.textContent = score;
  finalTimeEl.textContent = remainingTime;
  if (currentLevel === 5) {
    resultTitleEl.textContent = '🏆 ФИНАЛЬНАЯ ПОБЕДА!';
    nextLevelBtn.textContent = 'В главное меню';
  } else {
    resultTitleEl.textContent = 'Уровень завершён!';
    nextLevelBtn.textContent = 'Следующий уровень';
  }
  gameScreen.style.display = 'none';
  resultScreen.style.display = 'flex';}
```

Механика бомб:

При сборе 5 и более одинаковых фруктов создаётся бомба:

```
const groups = groupMatches(matches);
matches.forEach(m => board[m.row][m.col] = null);
groups.forEach(g => {
  if (g.length >= 5) board[g[0].row][g[0].col] = '💣'; });
```

Бомба активируется по клику и уничтожает 3×3 область:

```
function explodeBomb(row, col) {
  if (isProcessing) return;
  isProcessing = true;
  for (let i = Math.max(0, row-1); i <= Math.min(5, row+1); i++) {
    for (let j = Math.max(0, col-1); j <= Math.min(5, col+1); j++) {
      board[i][j] = null;}}
  score += 30;}

```

Отсутствие уведомлений при неудачных ходах

При неудачном обмене сообщение не отображается — ход просто отменяется:

```
const matches = findMatches();
  if (matches.length === 0) {
    await animateSwap(row, col, prev.row, prev.col);
  } else {
    await processMatches(matches);}

```

Автоматическое обновление поля

Если ходы отсутствуют, вызывается checkForMoves(), которая генерирует новое поле и собирает начальные комбинации:

```
async function checkForMoves() {
  if (!hasValidMoves()) {
    messageEl.textContent = 'Нет ходов — создаём новое поле!';
    const cells = document.querySelectorAll('.cell');
    cells.forEach(cell => {
      cell.classList.add('shuffle-cell');
    });
    await new Promise(r => setTimeout(r, 600));

    generateCleanBoard();
    renderBoard();
    let matches = findMatches();
    while (matches.length > 0) {
      await processMatches(matches);
      matches = findMatches();}
    setTimeout(() => {
      messageEl.textContent = '';
    }, 2000);}

```

Визуальные эффекты и анимации

Все анимации реализованы через CSS-ключевые кадры:

```
@keyframes shuffle-fall {
  0% {transform: translateY(-100px) rotate(-10deg); opacity: 0;}
  50% {opacity: 1;}
  100% {transform: translateY(0) rotate(0); opacity: 1;} }

```

Шкала прогресса динамически меняет цвет:

```
if (percent < 33) progressBarFill.style.background = '#f44336';
else if (percent < 66) progressBarFill.style.background = '#ff9800';
else progressBarFill.style.background = '#4caf50';

```

Такой подход обеспечивает плавную, визуально насыщенную и интуитивно понятную игровую среду, соответствующую современным стандартам веб-разработки и требованиям пользовательского опыта.

4 Выводы

В ходе исследования было разработано приложение «Три в ряд — Фруктовая Буря», которое успешно демонстрирует возможности нативных веб-технологий для создания сложных, визуально привлекательных и полностью функциональных игр. Архитектура проекта отличается модульностью, отсутствием внешних зависимостей и высокой читаемостью кода. Все ключевые игровые механики — от системы звёзд и бомб до автоматического обновления поля — реализованы эффективно и интуитивно понятны пользователю. Приложение может быть использовано как образовательный пример для изучения HTML, CSS и JavaScript, а также как самостоятельный игровой продукт, способный развивать логическое мышление и стратегическое планирование.

Библиографический список

1. Смирнов И.В. Использование CSS3 для создания анимированных веб-интерфейсов // Веб-разработка и дизайн. 2023. № 3. С. 12-18.
2. Петрова А.С. Разработка интерактивных игр на JavaScript: методы управления событиями мыши // Программирование для начинающих. 2022. № 7. С. 34-40.
3. Кузнецов Д.А. Программная реализация системы управления мультимедийным контентом в браузере // Мультимедиа технологии. 2021. № 5. С. 56-61.
4. Григорьева Е.М., Лебедев М.П. Создание адаптивных веб-приложений с использованием Flexbox и Grid // Современные веб-технологии. 2020. № 9. С. 22-27.
5. Тихонов П.Н. Оптимизация производительности веб-анимаций // Производительность веб-приложений. 2024. № 2. С. 45-50.
6. Материалы разработки интерактивного веб-приложения «Три в ряд — Фруктовая Буря». URL: https://drive.google.com/drive/folders/1Vb2ZyPdRc0r_HlhWuosYSanJ5P2xowuF?usp=sharing (дата обращения: 25.11.2025)