

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

на программный продукт

«Эхо Манёвра»

28 ноября 2025 г.

Содержание

1	ОБЩИЕ СВЕДЕНИЯ	2
1.1	Наименование программы	2
1.2	Назначение	2
1.3	Языки и средства разработки	2
2	АРХИТЕКТУРА ПРОГРАММЫ	2
2.1	Класс Game (Главный контроллер)	2
2.2	Механика ключей и дверей	3
2.3	Система «Эхо-Тени»	4
3	СТРУКТУРА УРОВНЕЙ	4
3.1	Формат уровней	4
3.2	Количество и тематика уровней	4
4	ФИЗИКА И УПРАВЛЕНИЕ	5
4.1	Физические параметры	5
4.2	Управление	5
5	СЮЖЕТНАЯ СИСТЕМА	5
6	ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ	5
6.1	Аппаратные требования	5
6.2	Программные требования	5
7	УСТАНОВКА И ЗАПУСК	6

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программы

Браузерная игра-платформер «Эхо Манёвра» (англ.: *Echo Maneuver*).

1.2 Назначение

Программа представляет собой головоломку-платформер, в которой игрок управляет роботом, избегая столкновения с собственной «эхо-тенью», повторяющей его прошлые действия. Игра включает систему ключей и запертых дверей для создания сложных головоломок.

1.3 Языки и средства разработки

- **Язык реализации:** JavaScript (ECMAScript 6+).
- **Язык разметки:** HTML5.
- **Графическая библиотека:** HTML5 Canvas API.
- **Шрифты:** Google Fonts (Orbitron).
- **Среда исполнения:** Любой современный веб-браузер.

2 АРХИТЕКТУРА ПРОГРАММЫ

Программа реализована по принципу «Single File Component» — весь код находится в одном HTML-файле.

2.1 Класс Game (Главный контроллер)

Центральный класс, управляющий всей логикой игры.

Основные поля:

levelIndex Текущий уровень (0-9).

tiles Массив игровых объектов уровня (стены, шипы, ключи, двери).

player Объект игрока с координатами, скоростью и состоянием.

hasKey Флаг: взял ли игрок ключ.

doorOpen Флаг: открыта ли дверь.

history Массив записей движений для тени.

frameCount Счётчик кадров.

Основные методы:

startLevel(idx) Инициализирует уровень, загружает карту, сбрасывает состояние игрока.

loadMap(mapData) Парсит ASCII-карту уровня в массив **tiles**.

update() Обрабатывает ввод, физику, коллизии и механику тени.

handleCollision(axis) Разрешает коллизии игрока с объектами мира.

winLevel() Вызывается при достижении открытой двери.

loop() Главный игровой цикл (вызывается через `requestAnimationFrame`).

2.2 Механика ключей и дверей

Ключ ('K'):

- Отображается как жёлтый круг на уровне.
- При касании игроком устанавливаются флаги `hasKey = true` и `doorOpen = true`.
- Ключ исчезает с карты (тип меняется на `'.'`).

Дверь ('D'):

- **Закрытая дверь (до взятия ключа):** Отображается как красный квадрат. Является физическим препятствием (работает как стена).
- **Открытая дверь (после ключа):** Отображается как зелёный портал. Проходима. При касании вызывается `winLevel()`.

Код обработки двери:

```
1 else if (t.type === 'D') {
2     if (this.doorOpen) {
3         this.winLevel();
4         return;
5     } else {
6         // -
7
8         if (axis === 'x') {
9             if (p.vx > 0) p.x = tx - p.w;
10            if (p.vx < 0) p.x = tx + TILE_SIZE;
11            p.vx = 0;
12        } else {
13            if (p.vy > 0) {
14                p.y = ty - p.h;
15                p.onGround = true;
16                p.vy = 0;
17            }
18            if (p.vy < 0) {
19                p.y = ty + TILE_SIZE;
20                p.vy = 0;
21            }
22        }
23    }
```

2.3 Система «Эхо-Тени»

Реализована через запись координат игрока в массив `history`.

Алгоритм:

1. Каждый второй кадр записывается текущая позиция игрока: `this.history.push({x, y})`.
2. После накопления более 90 записей начинается воспроизведение тени.
3. Тень отображается в позиции из `history[history.length - 90]` (задержка 90 записей = 180 кадров = 3 секунды при 60 FPS).
4. При пересечении хитбокса игрока и тени вызывается смерть: `this.player.dead = true`.

3 СТРУКТУРА УРОВНЕЙ

3.1 Формат уровней

Уровни описаны как массив строк (ASCII-карта). Каждый символ представляет тайл 40×40 пикселей.

Легенда символов:

Стена (непроходимый блок).

. Пустота (воздух).

P Точка спауна игрока.

K Ключ.

D Дверь (выход).

S Шип (смертельная ловушка).

3.2 Количество и тематика уровней

Игра содержит **10 уровней** с нарастающей сложностью:

1. **The Loop** — обучающий уровень.
2. **The Climb** — вертикальные платформы.
3. **Spike Floor** — введение шипов.
4. **The U-Turn** — обязательный возврат через пройденный путь.
5. **Cage Break** — игрок заперт в клетке.
6. **Zig Zag** — многоярусный лабиринт.
7. **Risk** — узкие проходы с шипами.
8. **Tight** — сложная навигация.
9. **Tower** — высокий вертикальный уровень.
10. **Final** — финальное испытание.

4 ФИЗИКА И УПРАВЛЕНИЕ

4.1 Физические параметры

Параметр	Значение
Гравитация (GRAVITY)	0.6 пикс/кадр ²
Сила прыжка (JUMP_FORCE)	-14 пикс/кадр
Скорость движения (SPEED)	5 пикс/кадр
Размер тайла (TILE_SIZE)	40 пикселей
Размер игрока	20×30 пикселей

Таблица 1: Физические константы

4.2 Управление

А / Стрелка влево Движение влево.

D / Стрелка вправо Движение вправо.

W / Стрелка вверх Прыжок (только на земле).

5 СЮЖЕТНАЯ СИСТЕМА

После завершения каждого уровня отображается текстовое сообщение из массива **STORY_TEXTS**. Сообщения хранятся на русском языке и раскрывают сюжет игры через диалог между «Системой» и «Неизвестным».

Пример:

```
1 const STORY_TEXTS = [
2     "                :                               .",
3     "                :                               .",
4     "                :                               .",
5 ];
```

6 ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ

6.1 Аппаратные требования

- Процессор: от 1.6 ГГц.
- ОЗУ: от 4 ГБ.

6.2 Программные требования

- ОС: Windows, macOS, Linux.
- Браузер: Chrome 90+, Firefox 88+, Safari 14+.
- JavaScript должен быть включён.

7 УСТАНОВКА И ЗАПУСК

Программа не требует установки. Для запуска необходимо открыть файл `index.html` в браузере. Игровой прогресс сохраняется автоматически в `localStorage`.