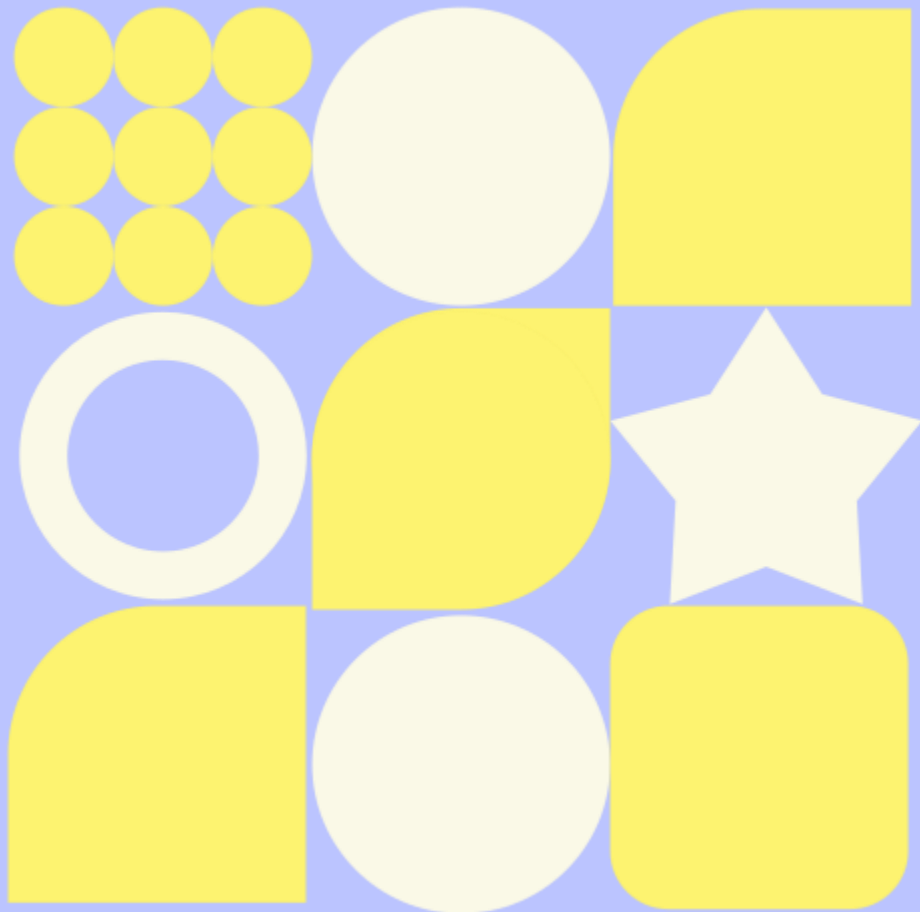


Обзор проекта Sokoban: браузерная версия классической логической игры

Создание версии Sokoban для браузера на
базе HTML, CSS и JavaScript.



Контекст и планирование разработки Sokoban

Проект создан в учебных целях для реализации логической игры в браузере с применением HTML5, CSS3 и JavaScript ES6. Включает проектирование карты, верстку, механику движения, а также функции подсчёта ходов и времени.



Изменения в процессе разработки

Добавление функции таймера, которая изначально не была запланирована, повысило информативность игры для пользователя и усложнило логику контроля времени.

Для управления состояниями ячеек применены CSS-классы, что упростило визуальное отображение и обновление игрового поля без вмешательства в логику.

Введён массив originalMap для учёта и правильного отображения целевых клеток, что значительно улучшило качество интерфейса и игровой механики.

Основные проблемы и методы их решения

Исправлена ошибка, позволяющая игроку толкать несколько ящиков подряд, благодаря введению проверки наличия препятствий за ящиком.



Чтобы различать ящики на целях, используется проверка массива `originalMap`, позволяющая корректно отображать такие ситуации визуально.



Для корректного завершения игры таймер теперь останавливается через `clearInterval` при достижении победного состояния, исключая продолжение отсчёта.



Достигнутые результаты проекта

Игра реализована с полной логикой перемещений, контроля ходов и победных условий, что обеспечивает стабильную и понятную механику.

Визуальное оформление дополнено тематическими текстурами, что улучшило восприятие и игровой опыт пользователей.

В проект успешно внедрены счётчик ходов и таймер, обеспечивающие информативность и мотивацию игроков.

Создана гибкая система обновления игрового поля, позволяющая легко вносить изменения и поддерживать код в дальнейшем.

Бюджет, сроки и состав команды

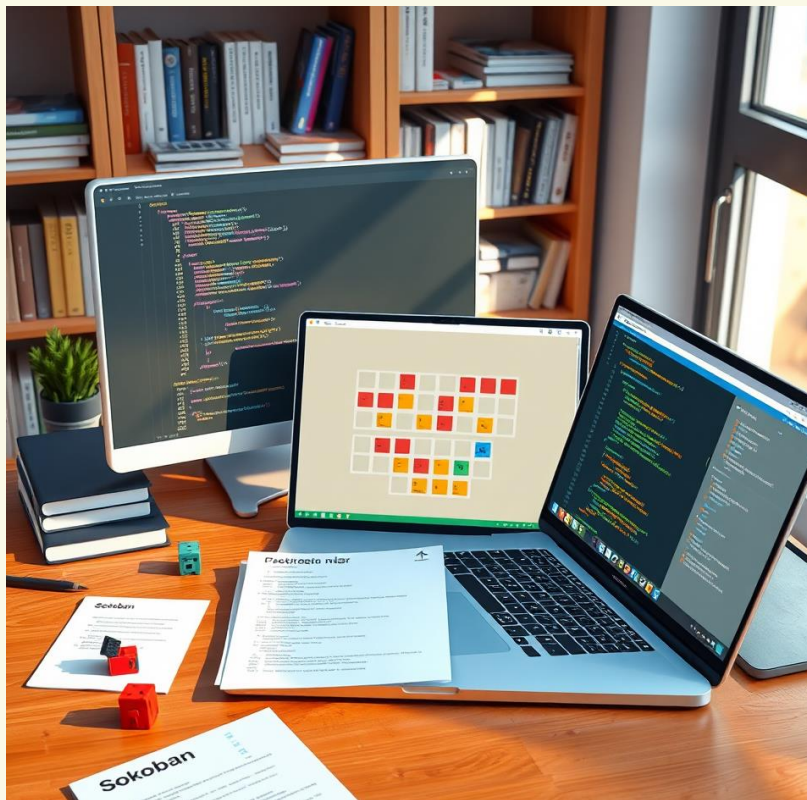
Показатель	Значение
Сроки	По плану
Бюджет	Бесплатно
Команда	1 разработчик

Проект выполнен в указанные сроки, с нулевым бюджетом и минимальным составом участников.

Все ключевые показатели соответствуют плановым, что подтверждает успешное выполнение задач в рамках доступных ресурсов.

Внутренний отчёт проекта

Факторы, влияющие на процесс разработки



Положительные моменты

Чёткое разделение логики и интерфейса облегчило разработку. Использование констант для типов клеток упростило код. Готовые спрайты из Minecraft ускорили визуальное оформление.

Основные трудности

Отсутствие опыта с таймерами в JavaScript усложнило реализацию. Также возникли сложности с отладкой движения ящиков, требующей тщательного контроля состояний.

Реализация целей проекта

Игра функционирует стабильно, реализована полноценная логика перемещений и взаимодействия элементов.

Интерфейс выполнен интуитивно понятным, что обеспечивает удобство управления и привлекательность для пользователей.

Дополнительно внедрены функции таймера и счётчика ходов, что изначально не планировалось, но значительно повысило качество продукта.

Планы по улучшению и развитию

1 Планируется добавить новые уровни для расширения игрового контента и увеличения реиграбельности.

3 Будут внедрены звуковые эффекты для улучшения погружения и ощущения игрового процесса.

2 Реализация функции отмены хода (Ctrl+Z) повысит удобство и позволит исправлять ошибки игроков.

4 Адаптивная вёрстка и сохранение прогресса обеспечат комфортную игру на мобильных устройствах и сохранение достижений.

Проверка состояния `map[boxNewX]?.[boxNewY]` на 0 или 4

позволяет эффективно предотвратить некорректное движение ящиков, сохраняя целостность игрового процесса.

Исследования и тестирование игрового движка

Для исключения возможности толкать ящик через стену или за пределы карты реализована строгая проверка состояния целевой клетки.

Альтернативные решения технической проблемы

В таблице представлены пять вариантов решения проблемы ограничения движения ящиков с оценкой их преимуществ и недостатков по критериям надёжности, сложности реализации, пользовательского опыта и ресурсозатрат.



Каждый вариант предлагает компромисс между надёжностью, сложностью и удобством для пользователя, требуя выбора подходящего баланса.

Вариант	Плюсы	Минусы
Проверка соседних клеток	Высокая надёжность	Избыточная сложность
Предрасчёт ходов	Быстрое выполнение	Сложен в реализации
Карта препятствий	Гибкость настроек	Потребляет память
Визуальная подсказка	Улучшает UX	Не решает логику
Ограничение по границам	Простота реализации	Мало гибкости

Анализ проектных решений и опыт разработки Sokoban

Выводы и опыт, полученный в ходе разработки

Проект успешно завершён, что позволило приобрести ценный опыт в создании браузерных игр с использованием современных технологий JavaScript, HTML и CSS. Это подтверждает мои навыки разработки и понимание игровых механизмов.

Созданный продукт отвечает изначальным целям и содержит возможности для будущего расширения функционала. На практике показана эффективность выбранных подходов и структурирования кода для дальнейших улучшений.

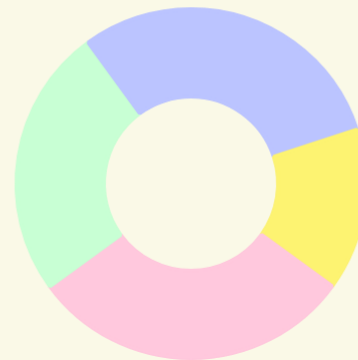
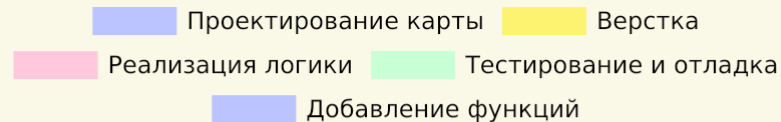


Диаграмма распределения времени по этапам разработки

Больше всего времени было выделено на реализацию логики и тестирование, что важно для стабильности игры.



Оптимальное распределение времени позволило успешно пройти ключевые этапы разработки без превышения сроков.



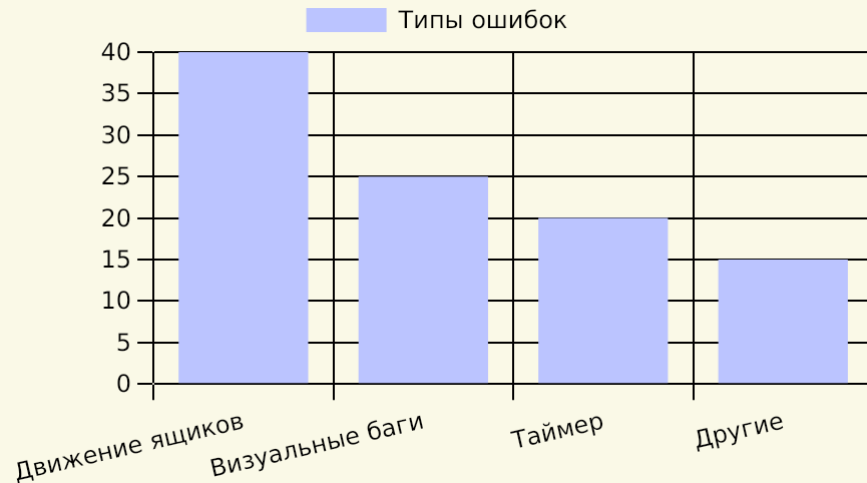
Личный учёт времени разработки, 2024 год

Статистика пользовательских действий и ошибок


Проблемы с движением ящиков требуют приоритетного исправления для улучшения игрового процесса.



Анализ ошибок помогает сфокусировать ресурсы на устранении критических багов, повышая качество и стабильность игры.



Результаты тестирования игры Sokoban, 2024



Обобщение и перспективы развития проекта Sokoban

Проект достиг основных целей и подтвердил выбранную архитектуру и технологии. В будущем планируется расширение функционала и адаптация под новые платформы для улучшения пользовательского опыта.

