

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка игр в формате GameJam (2D / Яндекс.Игры / VK Play)»
по дисциплине «Проектный практикум»

Заказчик: Рубцов Павел Евгеньевич

Куратор: Рубцов Павел Евгеньевич

Студенты команды DeBuild 2.0:

Климов Олег Анатольевич

Нураев Тимофей Сергеевич

Овчинников Никита Сергеевич

Петрушина Надежда Дмитриевна

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Основная часть	5
1.1 Работа каждого члена команды	5
1.1.1 Участие в разработке игры «Puss in Boots: Memory Rescue»	5
1.1.2 Участие в разработке игры «FuelDash»	7
1.2 Анализ требований и план разработки	10
1.2.1 Анализ требований заказчика и конечного пользователя	10
1.2.2 План разработки (Backlog).....	11
1.3 Анализ аналогов	12
1.4 Обзор архитектуры и компонентов программного продукта.....	14
1.4.1 Структура.....	14
1.4.2 Архитектура игры Puss in Boots: Memory Rescue.....	15
1.4.3 Архитектура игры FuelDash.....	16
1.4.4 Особенности сборки	16
1.5 Методология и процесс разработки	17
1.5.1 Методология разработки.....	17
1.5.2 Процесс разработки игры «Puss in Boots: Memory Rescue»	17
1.5.3 Процесс разработки игры «FuelDash»	19
1.6 Планирование деятельности и распределение задач.....	20
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24
ПРИЛОЖЕНИЕ А (справочное) Концепт-документ игры «Puss in Boots: Memory Rescue».....	25
ПРИЛОЖЕНИЕ Б (Справочное) Отрисованные объекты игры «Puss in Boots: Memory Rescue».....	28
ПРИЛОЖЕНИЕ В (справочное) Концепт-документ игры «FuelDash».....	29
ПРИЛОЖЕНИЕ Г (справочное) Отрисованные объекты игры «FuelDash» ..	33

ПРИЛОЖЕНИЕ Д (справочное)	Опубликованные игры на платформе
Яндекс.Игры	34

ВВЕДЕНИЕ

Цель проекта — разработка и выпуск коммерчески готовых 2D-игр, созданных по правилам GameJam'a, с последующей публикацией на площадках Яндекс.Игры или VK Play. Основные задачи включают: создание концепт-документа и дизайн-документа, проектирование интерфейсов, реализацию игрового прототипа в Unity с применением отрисованной векторной графики, настройку монетизации и прохождение модерации на выбранной платформе. Команда также должна обеспечить техническую стабильность проекта, сбалансированный и увлекательный геймплей, подходящий для игровых сессий продолжительностью от 15 до 40 минут.

Актуальность проекта определяется стабильным интересом пользователей к коротким, ярким и атмосферным 2D-играм, особенно если они вдохновлены узнаваемыми культурными образами, мультфильмами или обладают оригинальной механикой. GameJam - формат позволяет участникам не только развить профессиональные навыки в сжатые сроки, но и подготовить релевантный продукт для дальнейшего продвижения.

Разрабатываемые игры ориентированы на публикацию в веб-формате (WebGL) и дистрибуцию через российские игровые платформы. Основная аудитория — пользователи, ищущие развлекательный контент, подходящий для коротких игровых сессий, с возможностью соревновательных элементов или сюжетного вовлечения.

Ожидаемые результаты включают: создание полноценных, протестированных 2D-игр с оригинальным визуальным стилем, интуитивно понятным интерфейсом и продуманной игровой механикой. Каждая из игр будет представлять собой завершённый продукт, готовый к размещению на выбранной площадке и соответствующий всем требованиям модерации. Успешная реализация проекта повысит компетенции участников в области гейм-дизайна, UI/UX, работы с Unity и командной разработки в рамках ограниченного времени.

1 Основная часть

1.1 Работа каждого члена команды

1.1.1 Участие в разработке игры «Puss in Boots: Memory Rescue»

Овчинников Никита Сергеевич – Unity-разработчик:

а) Реализация ключевых игровых механик:

- реализация логики переворота/совпадения карт (классы Card, CardManager);
- создание системы обратного отсчета с интеграцией в интерфейс (TimerSystem);
- реализация системы подсчета очков и сохранения рекордов (ScoreManager).

б) Оптимизация и техническая настройка:

- настройка рендеринга для WebGL-платформ (сборка под itch.io и Яндекс.Игры);
- исправление утечек памяти, в частности выгрузка неиспользуемых ассетов.

в) Интеграция и управление ресурсами:

- подключение звуковых эффектов и музыкального сопровождения (AudioManager);
- настройка управления мышью.

г) Озвучивание персонажей:

- озвучивание реплик Кота в сапогах.

д) Публикация игры:

- интеграция лидерборда и рекламы для монетизации на платформе Яндекс.Игры;

- сборка финальной версии под WebGL и подготовка к размещению на Яндекс.Игры и Vk.Play;
- сборка финальной версии под WebGL и публикация на itch.io.

Петрушина Надежда Дмитриевна – Дизайнер/2D-художник:

а) Визуальное оформление:

- разработка визуального стиля карт в эстетике «кибер-сказки» (рубашка, лицевые стороны);
- отрисовка игровых персонажей (Кот в сапогах, котята, Шалтай-Болтай);
- отрисовка наград, на прохождение уровня (карта-ключ, золотой ключ);
- создание фоновых сцен, поддерживающих атмосферу игры.

б) Озвучка персонажей:

- озвучивание реплик котят.

в) Подготовка графических ресурсов:

- экспорт ассетов в Unity-совместимых форматах.

г) Тестирование игры:

- проведение ручного тестирования на всех этапах разработки;
- выявление багов, включая визуальные артефакты, зависание, ошибки логики;
- финальное тестирование WebGL-сборки на itch.io и Яндекс.Игры (проверка загрузки, производительности, соответствия требованиям платформ).

Нураев Тимофей Сергеевич – Дизайнер/2D-художник:

а) Визуальное оформление:

- проектирование пользовательского интерфейса;
- разработка визуального стиля карт в эстетике «кибер-сказки» (лицевые стороны);
- создание окружения, поддерживающих атмосферу игры;

- отрисовка наград, на прохождение уровня (USB-флеш-накопитель);
- дизайн элементов пользовательского интерфейса: кнопки, таймер, экран победы.

б) Подготовка графических ресурсов:

- экспорт ассетов в Unity-совместимых форматах.

в) Тестирование игры:

- проверка стабильности работы основных механик: переворот и совпадение карт;
- проверка корректной работы звуковых эффектов и озвучки.

Климов Олег Анатольевич – Гейм-дизайнер:

а) Проектирование концепции и игрового процесса:

- формирование сюжетной идеи «Кибер-скизки»;
- настройка баланса сложности (количество карт, лимит времени на уровень).

б) Озвучка персонажей:

- озвучивание реплик Шалтая-Болтая, запись и монтаж голосов.

в) Документация:

- составление дизайн-документа;
- составление описания, инструкций;
- составление чек-листов для тестирования (например, корректная работа пропуска реплик).

1.1.2 Участие в разработке игры «FuelDash»

Овчинников Никита Сергеевич – Unity-разработчик:

а) Реализация ключевых игровых механик:

- программирование управления автомобилем (смена полос);
- реализация системы расхода топлива и его пополнения;

- реализация системы сбора монет;
- обработка столкновений с препятствиями и последствиями (потеря топлива);
- генерация уровней;
- реализация смены локации и автомобиля, исходя из пройденного расстояния.

б) Интеграция графики и интерфейсов:

- настройка изометрической камеры;
- синхронизация 2D-спрайтов с 3D-пространством;
- реализация элементов HUD: индикатор топлива, счетчик монет, прогресс-бар уровня.

в) Оптимизация и публикация:

- интеграция лидерборда и рекламы для монетизации на платформе Яндекс.Игры;
- сборка финальной версии под WebGL и подготовка к размещению на Яндекс.Играх и Vk.Play;
- сборка финальной версии под WebGL и публикация на itch.io.

Петрушина Надежда Дмитриевна – 2D-художник:

а) Визуальное оформление:

- отрисовка игровых объектов: три вида автомобилей, препятствия, монета, канистра, объекты для фона (для двух локаций);
- проработка фоновых сцен и локаций в мультяшном стиле;
- отрисовка дизайна экрана с результатами, кнопок;
- подбор фоновых текстур (например, текстура дороги, травы и т.д.).

б) Подготовка графических ресурсов:

- экспорт ассетов в Unity-совместимых форматах.

в) Поиск и подбор звуковых эффектов:

- поиск фоновой музыки для игры.

г) Тестирование игры:

- проведение ручного тестирования на всех этапах разработки;
- выявление багов, включая визуальные артефакты, зависание, ошибки логики;
- финальное тестирование WebGL-сборки на itch.io и Яндекс.Игры (проверка загрузки, производительности, соответствия требованиям платформ).

Нураев Тимофей Сергеевич – Дизайнер:

а) Визуальное оформление:

- проектирование пользовательского интерфейса;
- отрисовка элементов интерфейса (кнопки, иконки, полосы прогресса, панели HUD);
- отрисовка объектов для фона третьей локации.

б) Подготовка графических ресурсов:

- экспорт ассетов в Unity-совместимых форматах.

в) Поиск и подбор звуковых эффектов:

- поиск звуковых эффектов для столкновения, пополнения топлива, сбора монет.

г) Тестирование игры:

- проверка стабильности работы основных механик: переворот и совпадение карт;
- проверка корректной работы звуковых эффектов и озвучки.

Климов Олег Анатольевич – Гейм-дизайнер:

а) Проектирование концепции и игрового процесса:

- формирование идеи игры «Плоский мир»;
- дизайн мета-игры: покупка топлива за монеты.

б) Поиск и подбор звуковых эффектов:

- поиск звуковых эффектов для клика, для победы и поражения.

в) Документация:

- составление дизайн-документа;
- составление описания, правил (обучение в начале заезда);
- составление чек-листов для тестирования (например, корректная работа сбора монет и топлива в течение игры).

г) Анализ референсов:

- адаптация лучших практик из игр: Subway Surfers/Jetpack Joyride.

1.2 Анализ требований и план разработки

1.2.1 Анализ требований заказчика и конечного пользователя

Создание игры в рамках GameJam предполагает работу с двумя основными категориями требований: требованиями заказчика и ожиданиями конечного пользователя. Грамотный учёт обоих направлений позволяет выстроить эффективный процесс разработки и создать востребованный, конкурентоспособный продукт.

В рамках геймджема заказчик предъявляет ряд обязательных условий:

а) Использование игрового движка Unity — стандарт де-факто в разработке 2D и 3D-игр, что обеспечивает кроссплатформенность и широкие возможности по реализации механик;

б) Создание 2D-графики в Figma — предполагается визуальное оформление интерфейсов и игровых элементов в рамках современного редактора с возможностью экспортирования в Unity;

в) Соответствие заданной теме GameJam'a — креативный концепт игры должен быть связан с объявленной темой (например, «Киберсказка», «Плоский мир»);

г) Публикация игры на целевых платформах: Яндекс.Игры, VK Play или itch.io. При этом важны требования к сборке, форматам, безопасности и наличию необходимых метаданных;

- д) Наличие элементов геймдизайна, интерфейса и системы монетизации (игра должна быть полноценным продуктом, а не технической демоверсией);
- е) Сборка проекта в формате WebGL, что обусловлено браузерной природой площадок;
- ж) Регулярная отчётность и участие в кураторских встречах — способствует постоянному контролю и корректировке хода разработки;
- з) Прохождение полного производственного цикла: от идеи до релиза, включая проектирование, реализацию, тестирование и публикацию.

Для привлечения и удержания аудитории необходимо ориентироваться на ожидания игроков, особенно в условиях коротких игровых сессий и высокой конкуренции в гиперказуальном сегменте:

- а) Простое и интуитивное управление — минимальное обучение, быстрый вход в игру;
- б) Визуально привлекательная графика — стилистическая целостность, приятная цветовая палитра, чёткое различение объектов;
- в) Звуковое сопровождение и озвучка — музыка и эффекты усиливают погружение и создают эмоциональный отклик;
- г) Доступный игровой процесс — отсутствие чрезмерной сложности, понятные цели и правила;
- д) Метаигровые элементы — система прогрессии (покупка топлива), повышающая вовлеченность.

1.2.2 План разработки (Backlog)

Разработка игр «Puss in Boots: Memory Rescue» и «FuelDash» велась согласно единому плану, разбитому на шесть логических этапов. Такой подход позволил грамотно распределить нагрузку между членами команды и обеспечить контроль над дедлайнами в условиях жёстких временных рамок GameJam. Бэклог продукты представлен в Таблица 1.

Таблица 1 - Бэклог продукта

Приоритет	Задача	Тип	Критерии выполнения
1	Изучить движок Unity	Исследование	Готов список tutorиалов
2	Придумать концепт игры на тему «кибер-сказка»; Концепт для игры с изометрической постановки камеры.	Дизайн	Есть описание и скетчи
3	Создать прототип механик	Разработка	Рабочая версия на ПК
4	Нарисовать арты для игры	Арт	Готовы спрайты
5	Озвучить персонажей/добавить аудио в игру	Арт	Готова озвучка/звуки встроены в игру
6	Выложить игру на целевую платформу	Релиз	Ссылка доступна

1.3 Анализ аналогов

Для построения собственных игровых механик и визуального стиля наша команда провела анализ популярных аналогов, близких по жанру, управлению и целевой аудитории.

Для игры «Puss in Boots: Memory Rescue» в качестве ориентиров были выбраны классические игры в жанре «поиск пар»: маджонг и настольная игра «Memory» (Рисунок 1).

Основной механикой в данных играх является нахождение и сопоставление двух одинаковых элементов с последующим их удалением с поля. Механика была адаптирована в цифровой формат и усилена сюжетной составляющей: игрок помогает Коту в сапогах спасти котят от Шалтая-Болтая, находя соответствующие пары и продвигаясь по сюжетной линии. Таким образом, простая, хорошо знакомая игроку основа сочетается с оригинальным визуальным оформлением и нарративом.



Рисунок 1 - Игра «Memory»

Для игры «FuelDash» были выбраны два узнаваемых примера в жанре аркадных раннеров и гонок с прогрессией: Subway Surfers и Hill Climb Racing (Рисунок 2, Рисунок 3).

От Subway Surfers была заимствована идея бесконечного движения по трассе с препятствиями и бонусами, от Hill Climb Racing — подход к управлению топливом и физикой движения транспорта. Разработанная версия комбинирует эти элементы, добавляя оригинальную механику сбора топлива. Такой синтез позволяет сделать игру динамичной и вовлекающей при сохранении узнаваемых для пользователя паттернов взаимодействия.

Проведённый анализ аналогов позволил выделить ключевые элементы успеха подобных игр, избежать типичных ошибок и создать продукты, сочетающие привычные механики с уникальным стилем и контекстом



Рисунок 2 - Игра «Subway surfers»



Рисунок 3 - Игра «Hill climb racing»

1.4 Обзор архитектуры и компонентов программного продукта

1.4.1 Структура

Обе игры («Puss in Boots: Memory Rescue» и «FuelDash») - построены по модульной архитектуре, что позволило разделить разработку на независимые

и легко тестируемые компоненты. Это значительно ускорило процесс итеративной разработки в рамках GameJam и облегчило внесение правок.

Каждый проект включает в себя три ключевых слоя:

- Игровая логика (Scripts): скрипты, реализующие поведение объектов и игровые механики;
- Интерфейс пользователя (UI): отдельный слой со всеми компонентами HUD (таймеры, кнопки, индикаторы);
- Ресурсы (Assets): ассеты графики, звуков и шрифтов, экспортированные из Figma и интегрированные через Unity.

1.4.2 Архитектура игры Puss in Boots: Memory Rescue

Игра реализована на основе классической механики "Memory Match". Архитектура включает следующие основные компоненты:

- CardManager — управляет созданием, размещением и состоянием карточек;
- Card — отдельный компонент каждой карточки, содержащий методы переворота и проверки на совпадение;
- TimerSystem — отвечает за обратный отсчёт времени, проигрыш при нулевом значении;
- ScoreManager — подсчёт очков и управление рекордами;
- AudioManager — воспроизведение звуковых эффектов и реплик персонажей;
- DialogueSystem — отображение диалогов перед началом уровня и в конце (в зависимости от результата);
- SceneController — переключение сцен и управление переходами.

UI построен с помощью Canvas и включает таймер, очки, кнопку «Пропустить диалог», экран победы и поражения. Все сцены оптимизированы под WebGL и адаптированы к разрешению браузера.

1.4.3 Архитектура игры FuelDash

FuelDash представляет собой бесконечный аркадный раннер, использующий псевдо-3D в изометрическом пространстве. Основные компоненты архитектуры:

- CarController — управление автомобилем, переключение полос, обработка столкновений;
- FuelSystem — контроль за уровнем топлива, визуализация шкалы, реакция на сбор канистр и потери;
- LevelGenerator — динамическая генерация препятствий, канистр и монет;
- HUDManager — отображение интерфейса: топливо, монеты, расстояние, прогресс;
- GameManager — общая логика старта/завершения игры, возрождение после проигрыша;
- LeaderboardManager — реализация лидерборда для сохранения и отображения рекордов;
- AdManager — интеграция рекламы для монетизации на Яндекс.Играх.

Изометрическая камера (IsometricCameraController) настроена вручную под работу с 2D-спрайтами, создавая эффект объёма. Все игровые объекты (машины, препятствия, фон) реализованы как 2D-спрайты в 3D-пространстве с ручной установкой порядка отрисовки.

1.4.4 Особенности сборки

Обе игры собираются в формате WebGL, что обеспечивает совместимость с Яндекс.Играми, VK Play и itch.io.

Используется одноуровневая сцена с подгружаемыми префабами, что упрощает тестирование и снижает нагрузку на память.

Все ассеты оптимизированы вручную: понижены разрешения фонов, удалены неиспользуемые ресурсы.

Такая архитектура позволила обеспечить стабильную работу, быстрое время загрузки и высокую производительность даже на слабых устройствах.

1.5 Методология и процесс разработки

1.5.1 Методология разработки

В условиях ограниченного времени (4 дня на GameJam и параллельная работа над проектной практикой) команда выбрала гибридную методологию на стыке Scrum и Kanban:

- От Scrum были заимствованы: ежедневные/еженедельные стендапы, чёткие сроки выполнения этапов, распределение ролей;
- От Kanban — гибкость в постановке и перераспределении задач по мере появления непредвиденных ошибок или новых идей;
- Инструмент трекинга: YouGile, с доской, включающей колонки To Do, In Progress, Testing и Done.

1.5.2 Процесс разработки игры «Puss in Boots: Memory Rescue»

Разработка проекта велась в условиях жёсткого дедлайна GameJam'a (72 часа) и была разбита на три ключевых этапа, каждый из которых решал определённый набор задач, направленных на быстрое создание минимально жизнеспособного, но завершённого продукта:

- а) Начало разработки, составление концепции (1 день):
 - Подбор референсов;
 - Создание визуального концепта в Figma – начало отрисовки карточек, общего стиля персонажей. Эти элементы экспортировались в Unity для использования в сценах;

- Составление дизайн-документа – описание жанра, основные игровые механики, пользовательский сценарий и т. д. (приложение А);

- Реализация базовой механики – написание прототипа логики переворота карточек, работа над обработкой таймера.

б) Продолжение разработки и озвучка (1 день):

- Доработка механик – реализация логики совпадений, удалений карт, проигрыш по таймеру, обновление UI во время игры;

- Озвучка персонажей – подготовлены и записаны реплики главных персонажей, сопровождающие игровые события;

- Отрисовка объектов – Figma-ассеты окончательно перенесены в Unity, привязаны к игровым объектам и сценам (приложение Б).

в) Доработка игры, сборка проекта (1 день):

- Добавление анимаций и эффектов – настроены анимации переворота карточек, появление/исчезновения элементов;

- Финальное тестирование – проверены все сцены, устранены критические баги;

- Создание и загрузка сборки WebGL – проект собран под WebGL, адаптирован под браузерные требования и опубликован на платформе itch.io с кратким описанием и сопроводительными материалами [6].

г) Доработка игры после GameJam`а (1 неделя):

- Реализация лидерборда, внедрение рекламы для монетизации;

- Реализация пропуска всех реплик;

- Повторное тестирование на доработанной сборке – было выявлено наложение реплик друг на друга при быстро пропуске. Для устранения ошибки и улучшения геймплея был реализован пропуск всех реплик сразу;

- Создание и загрузка сборки WebGL – проект собран под WebGL, адаптирован под браузерные требования и опубликован на платформе Яндекс.Игры с кратким описанием и сопроводительными материалами.

1.5.3 Процесс разработки игры «FuelDash»

Разработка игры «FuelDash» началась за 4 недели до старта GameJam`а в рамках учебной подготовки команды. Целью было опробовать полноценный производственный цикл игры: от идеи и базовой механики до визуального оформления и публикации. После завершения GameJam`а проект продолжал дорабатываться – в него вносятся новые этапы.

В отличие от компактного проекта «Puss in Boots: Memory Rescue», игра «FuelDash» имеет более масштабную структуру и ориентирован на длинные игровые сессии, включающие перемещение по трассе, сбор ресурсов и избегание препятствий. Процесс разработки делился на следующие этапы:

а) Прохождение обучения, начало работы (2 недели):

- Прохождение вводных курсов: «Основы Unity», курс от Яндекс.Игр, обучение на Stepik [1];

- Изучение статей по гейм-дизайну, UX и проектированию интерфейсов [2,3,4];

- Подбор визуальных референсов по игровым объектам: автомобили, препятствия, канистры, окружение;

- Создание визуального концепта в Figma – собраны референсы, начата работа над визуальным стилем трассы, объектов и интерфейса;

- Составление концепт-документа – описание игрового процесса, управление, цели игрока, примерный список сцен и экранов (приложение В).

б) Реализация MVP-игры (2 недели):

- Актуализация концепт-документа: дополнены разделы с описанием внутриигровых фиш;

- Добавление звукового сопровождения: интеграция предварительно подобранных звуков эффектов;

- Отрисовка интерфейса, элементов управления и объектов – меню, кнопки, индикаторы топлива и монет, машины и т. д. (приложение Г);

- Реализация базовой экономики - добавлен функционал покупки топлива при его окончании;
- Тестирование MVP-версии – проведено тестирование основных механик (движение автомобиля, сбор топлива и монет, срабатывание проигрыша при отсутствии топлива) по чек-листу. При данном тестировании ошибок не было обнаружено.
- Подготовка к публикации – отрисовка обложек, реализация отключения звука, сборка проекта.
- в) Доработка игры до финального релиза (3 недели):
 - Добавление контента - новые локации, новые машины;
 - Внедрение системы рекордов – внедрение лидерборда с рекордами пройденного расстояния;
 - Внедрение рекламы для монетизации [7];
 - Финальное тестирование – комплексная проверка UI, звуков, рекламных интеграций, логики покупки и лидерборда. При данном тестировании ошибок не было обнаружено;
 - Сборка и публикация финальной версии игры на различные платформы [5, 8].

1.6 Планирование деятельности и распределение задач

Планирование деятельности осуществлялось с учётом ограниченного времени и необходимости поэтапного выполнения ключевых задач. В начале проекта был составлен ориентировочный план, включающий разработку концепции, визуального стиля, реализацию игровых механик, тестирование и подготовку проекта к публикации. Для отслеживания выполнения задач использовались таблицы и ежедневные созвоны в командном чате, где обсуждались прогресс и текущие сложности.

Работа велась в распределённой команде, состоящей из четырёх участников, каждый из которых взял на себя отдельную роль:

- Программист отвечал за реализацию механик и сборку проекта в Unity;
- Художник занимался созданием всех графических элементов — от фонов до анимаций;
- Проектировщик интерфейса разрабатывал структуру и внешний вид экранов, обеспечивая удобство взаимодействия;
- Геймдизайнер разрабатывал сценарий, игровые правила, систему прогрессии и логическое наполнение уровней.

При необходимости задачи перераспределялись: например, художник участвовал в оформлении интерфейсов, а геймдизайнер помогал в финальном тестировании. Благодаря чёткой координации и гибкому подходу к управлению задачами, проект был успешно реализован в срок.

ЗАКЛЮЧЕНИЕ

Разработанные в рамках проекта игры (приложение Д) соответствуют изначально сформулированным целям и требованиям как с точки зрения технической реализации, так и с точки зрения пользовательского восприятия. Игровые механики были полностью реализованы в пределах жанра, а интерфейс — адаптирован под аудиторию, ориентированную на короткие и динамичные сессии. Использование WebGL позволило обеспечить совместимость с браузерными платформами, что соответствует критерию доступности, заявленному в начале проекта.

Результаты тестирования показали достаточную стабильность обеих игр при запуске в разных браузерах и на различных устройствах. Были выявлены отдельные баги — такие как некорректное срабатывание триггеров или зависание UI-элементов — однако они не носили системного характера и были устранены до релизной сборки. Это говорит о высоком качестве архитектуры проекта и внимательности при реализации ключевых компонентов. Тестирование также подтвердило интуитивную понятность игрового процесса: пользователи без инструкций успешно осваивали механику, что является важным критерием качества для казуальных игр.

Несмотря на общую завершённость и успешное прохождение модерации, у проекта сохраняется значительный потенциал для дальнейшего развития. В частности, возможно:

- расширение контента (новые уровни, персонажи, карты, сюжетные сцены);
- улучшение системы монетизации (внедрение аналитики, балансировка рекламы);
- оптимизация под мобильные устройства с низкой производительностью;
- интеграция системы достижений и социальных функций (например, обмен результатами между игроками).

Опыт, полученный в рамках проекта, подтвердил, что команда способна эффективно работать в условиях жёсткого дедлайна, выстраивать рабочие процессы и создавать продукт, сочетающий оригинальный визуальный стиль, продуманный геймдизайн и стабильную техническую основу. Проект показал, что формат GameJam не только способствует прокачке профессиональных навыков, но и может служить стартовой точкой для создания коммерчески перспективных продуктов.

В дальнейшем подобный подход можно масштабировать: как в рамках внутренних учебных инициатив, так и для работы с реальными заказами на рынок браузерных или мобильных 2D-игр.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Unity | Yandex Games – документация для разработчиков [Электронный ресурс] / Notion. – URL: <https://yandexgames-unity.notion.site/Unity-959b137b6d644630a6acc598a8945168> (дата обращения: 29.03.2025).

2. aviGames, Проектирование UX в Figma [Электронный ресурс] / aviGames / Telegraph. - 2023. - URL: <https://telegra.ph/Proektirovanie-UX-v-Figma-11-21> (дата обращения: 28.03.2025).

3. aviGames, Написание концепт-документа [Электронный ресурс] / aviGames / Telegraph. – 2023. – URL: <https://telegra.ph/1-Napisanie-koncept-dokumenta-10-17> (дата обращения: 28.03.2025).

4. aviGames, Проектирование UX в Figma [Электронный ресурс] / aviGames / Telegraph. - 2023. - URL: <https://telegra.ph/Proektirovanie-UX-v-Figma-11-21> (дата обращения: 28.03.2025).

5. Siberian Game Jam. Руководство по загрузке материалов [Электронный ресурс] / Оргкомитет Siberian Game Jam. – 2023. – URL: <https://sibgamejam.com/uploadguide> (дата обращения: 3.05.2025).

6. Unity Documentation [Электронный ресурс] / Unity Technologies. - URL: <https://docs.unity.com/> (дата обращения: 29.03.2025).

7. Документация Яндекс.Игры [Электронный ресурс] / Яндекс. - URL: <https://yandex.ru/dev/games/doc/ru/> (дата обращения: 22.04.2025).

8. Документация VK Play [Электронный ресурс] / VK. - URL: <https://documentation.vkplay.ru/> (дата обращения: 23.05.2025).

ПРИЛОЖЕНИЕ А

(справочное)

Концепт-документ игры «Puss in Boots: Memory Rescue»

Описание продукта: Puss in Boots: Memory Rescue — это браузерная головоломка в стиле "киберсказки", где игрок помогает Коту в сапогах спасти котят, попавших под влияние Шалтая-Болтая. Игроку предстоит решать простые, но увлекательные задачи на запоминание и скорость: находить пары одинаковых карточек за ограниченное время. Сюжет подаётся через озвученные диалоги персонажей, а игровой процесс дополнен интерактивными реакциями героев в зависимости от успеха или неудачи игрока.

Ключевые особенности:

- а) Геймплей: Классический механизм "Memory Match" с элементами таймера и сюжетной подачи;
- б) Сеттинг: Стилизация под киберпанк-сказку — смесь технологий и сказочных мотивов;
- в) Интерактивность: Персонажи реагируют на действия игрока (успех/неудача) уникальными репликами;
- г) Озвучка: Полностью озвученные диалоги для погружения в историю;
- д) Простота и доступность: Игра рассчитана на короткие сессии и широкую аудиторию.

Технические детали:

Таблица 1 – Технические детали

Параметр	Описание
Платформа:	Браузер (Яндекс.Игры, HTML5)
Жанр:	Головоломка / Казуальная игра
Целевая аудитория:	Дети 12+ лет, поклонники сказок и простых головоломок

Продолжение таблицы 1

Модель монетизации:	Бесплатно
Движок:	Unity (WebGL)
Арт-стиль:	Яркий мультяшный стиль с элементами киберпанка (неоновые элементы)

Игровой процесс:

а) Старт уровня:

– Игрок видит короткую сцену с диалогом между Котом, Шалтаем-Болтаем и котятами;

– После диалога открывается игровое поле с закрытыми карточками.

б) Механика:

– Игрок открывает по две карточки за ход, пытаясь найти пары;

– На уровень даётся ограниченное время (например, 60 секунд);

– При успешном завершении уровня игрок видит реакцию персонажей (положительную);

– При неудаче — альтернативную реакцию (отрицательную).

в) Уровни:

– Уровень 1: "Ключ-карта";

– Уровень 2: "USD-флеш-накопитель";

– Уровень 3: "Золотой ключ";

г) Интерактив:

– Во время уровня Шалтай-Болтай и Кот комментируют прогресс игрока;

– При успехе/неудаче запускается уникальная анимация и реплика.

д) Визуал и звук

– Персонажи в стиле "кибер-сказки" (Кот в сапогах с технологичными аксессуарами, Шалтай-Болтай в кибер-броню);

– Карточки оформлены в неоновых тонах;

– Фоновая музыка в стиле мультфильма «Кот в сапогах».

е) User Flow (UX)

Старт → Диалоговая сцена → Игровой процесс → Реакция персонажей .

ж) Референсы игр

– "Memory" (классическая настольная игра) — базовая механика поиска парных карточек;

– "Bejeweled Blitz" — динамика ограниченного времени и простых правил;

– "Peek-a-Zoo" (Dora the Explorer) — детские головоломки с запоминанием и анимацией персонажей.

з) Уникальное торговое предложение

– Сказка + киберпанк: Необычный сеттинг, сочетающий классических персонажей и технологичный антураж;

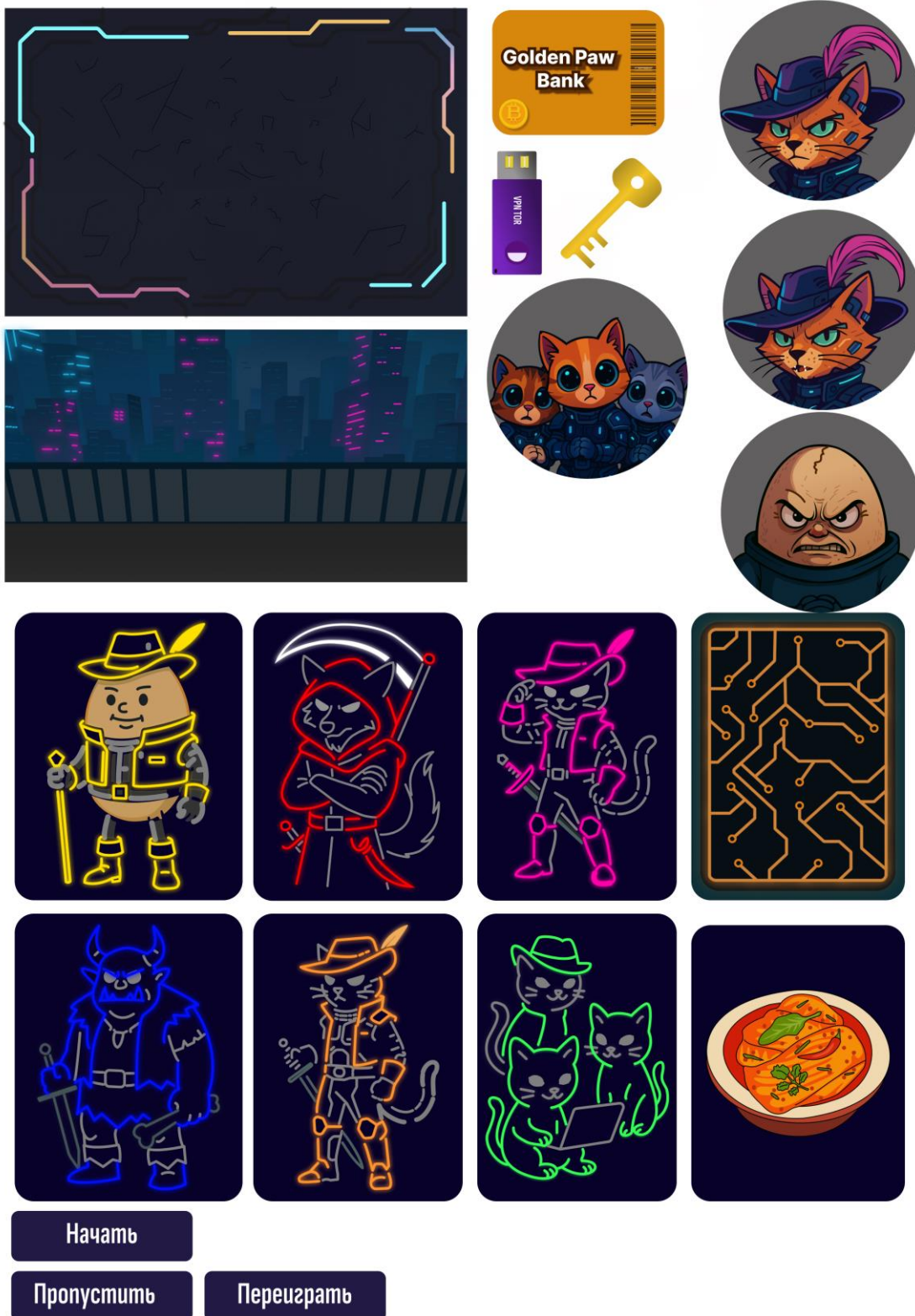
– Эмоциональный интерактив: Персонажи реагируют на действия игрока, создавая эффект сопричастности;

– Простота и глубина: Лёгкие правила, но увлекательный сюжет и мотивация играть дальше.

ПРИЛОЖЕНИЕ Б

(Справочное)

Отрисованные объекты игры «Puss in Boots: Memory Rescue»



ПРИЛОЖЕНИЕ В

(справочное)

Концепт-документ игры «FuelDash»

Описание продукта: FuelDash — это аркадная 3D-гонка. Игрок управляет автомобилем, проходя трассу с ограниченным запасом топлива, собирая канистры и монеты, избегая препятствий. Игра акцентирует внимание на управлении ресурсом (топливом) и точности манёвров.

Тема проекта: Плоский мир реализована через визуальный стиль: все объекты (включая машину, препятствия, канистры и трассу) выполнены в виде 2D-спрайтов, расположенных в 3D-пространстве. Камера располагается сбоку под углом (изометрия), создавая эффект "трёхмерного мира из плоских элементов".

Таблица 2 – Технические детали

Платформа:	Яндекс.Игры (только ПК, WebGL)
Сегмент:	Гиперказуальные аркады / казуальные гонки
Модель монетизации:	Отсутствует. Игра бесплатная, без рекламы и покупок.
Жанр:	Аркадный раннер с топливной механикой
Сеттинг:	Плоский 3D-мир
Движок:	Unity 6 (WebGL билд)
Арт-стиль:	Плоские 2D-спрайты в 3D-пространстве ("плоский мир"). Яркая палитра, мультяшный стиль. Толщина задаётся вручную (визуально), без реальной 3D-модели.

Целевая аудитория:

Таблица 3 – Целевая аудитория

Сегмент ЦА:	Поклонники лёгких аркадных игр и раннеров
Пол / возраст:	12–35 лет, преимущественно школьники и студенты

Продолжение таблицы 3

Поведение:	Играют в браузере между делами, ищут короткие игровые сессии и понятные правила
Референсы по ЦА:	Subway Surfers, Hill Climb Racing, Jetpack Joyride
Целевой рынок:	Яндекс.Игры, аудитория Рунета
USP игры: (Unique Selling Point)	Управление ресурсом (топливом) в сочетании с рефлексами и точностью вождения на фиксированных уровнях. Визуальная фишка — эффект псевдо-3D в плоском мире.

Игровой опыт: Игрок управляет автомобилем, который постепенно тратит топливо. Нужно суметь доехать как можно дальше, собирая канистры и монеты, избегая столкновений. Управление осуществляется вручную. Каждое столкновение и промах с канистрой уменьшает максимальное расстояние, которое игрок сможет пройти за заезд.

Игровая цель: Проехать, как можно дальше, собирая канистры с топливом и монеты.

Игровой цикл (core):

- а) Игрок начинает заезд (нажатие кнопки "Старт");
- б) Краткое обучение (в начале заезда);
- в) Игрок управляет машиной, которая автоматически едет вперёд;
- г) Топливо убывает каждый кадр;
- д) Игрок:
 - Перемещается между тремя полосами с помощью клавиш A и D (или стрелки влево/вправо);
 - Собирает красные канистры (+20% топлива);
 - Избегает препятствий (камни, бетонные блоки), которые отнимают 10% топлива при столкновении.
- е) Собирает монеты (используются для покупки топлива);
- ж) Поездка заканчивается при полном исчерпании топлива;

з) Показ экрана с результатом (кол-во собранных монет, пройденное расстояние);

и) Игрок возвращается в меню или запускает заезд заново.

Игровые механики:

а) Физические механики:

– Движение автомобиля: Имитация инерции и управления машиной (А, D или стрелки);

– Столкновения: Физика взаимодействия с препятствиями;

– Изменение локации: при преодолении расстояния, меняется локация.

б) Экономические механики:

– Топливо: Основной ресурс, который тратится со временем и восполняется через канистры;

– Монеты: Вторичный ресурс для покупки возрождения;

– Штрафы: Потеря топлива при столкновении с препятствиями.

в) Ментальные механики:

– Реакция: Быстрое принятие решений при объезде препятствий;

– Планирование: Оптимизация маршрута для сбора канистр и монет.

Игровые сущности:

а) Игровые объекты:

– Автомобиль: Основной объект управления, который визуально меняется с пройденным расстоянием;

– Красная канистры: +20% топлива;

– Препятствия: Камни, барьеры (отнимают топливо);

– Монеты: Для покупки возрождения.

б) Игровые ресурсы:

– Топливо: Основной ограничивающий фактор;

– Монеты: Накопление для разблокировки контента;

Core-game:

а) Основные механики:

- Управление машиной (A, D);
- Сбор топлива и монет;
- Избегание препятствий.

б) Игровой цикл:

Старт заезда → Движение и сбор ресурсов → Завершение заезда → Награда → Возврат в меню/новый заезд.

Meta-game:

Вторичные механики: Лидерборды (сравнение результатов с другими игроками).

Модель монетизации: Публикация на Яндекс.Игры.

Нарратив и визуал:

а) Нарратив: Минималистичный сюжет: "Гонщик путешествует по плоскому миру, чтобы найти легендарный источник топлива".

б) 2. Референсы:

- Арт: Subway Surfers (яркие цвета), Jetpack Joyride (мультяшный стиль).
- UI: Простое меню с интуитивными иконками на фоне игры (нажмите любую кнопку, чтобы начать игру).

User Flow (UX):

а) 1. Сценарий игрока:

Главное меню → Игра → Результаты.

б) 2. Wireframe:

- Главный экран: кнопка "Настройка звука".
- Игровой HUD: шкала топлива, счетчик монет, кнопка звука, прогресс бар.


ПРИЛОЖЕНИЕ Г (справочное)




Отрисованные объекты игры «FuelDash»



ПРИЛОЖЕНИЕ Д (справочное)

Опубликованные игры на платформе Яндекс.Игры

**Игра Puss in Boots: Memory Rescue** 12+
DeBuild · Головоломки · Карточные



Об игре


Puss in Boots: Memory Rescue — это браузерная головоломка в стиле "киберсказки", где игрок помогает Коту в сапогах спасти котят, попавших под влияние Шалтая-Болтая. Игроку предстоит решать простые, но увлекательные задачи на запоминание и скорость: находить пары одинаковых карточек за ограниченное время. Сюжет подаётся через озвученные диалоги персонажей, а игровой процесс дополнен интерактивными реакциями героев в зависимости от успеха или неудачи игрока.




Как играть

- Кликни по карточке, чтобы открыть её.
- Кликни по второй карточке — если пара совпадает, они исчезнут.
- Найди все пары до окончания времени, чтобы пройти уровень.

без регистрации · без скачивания · бесплатные · браузерные · десктоп

для компьютера · для развития мозга · логические · на внимание

**Игра Fuel Dash** 6+
DeBuild · Аркады



Об игре

Fuel Dash - Аркадная 3D-гонка. Игрок управляет автомобилем, проходя трассу с ограниченным запасом топлива, собирая канистры и монеты, избегая препятствий. Игра акцентирует внимание на управлении ресурсом (топливом) и точности манёвров.

Как играть

Авто движется вперёд автоматически. Чтобы повернуть направо - нажми "D", налево - нажми "A". Чтобы купить бонус с дополнительным топливом - нажми "R"

без регистрации · без скачивания · бесконечные · бесплатные · браузерные

в хорошем качестве · десктоп · для компьютера · мобильные · на внимание

на ловкость