

ИИ ДЛЯ РАЗРАБОТЧИКОВ

50 СПОСОБОВ УСКОРИТЬ
СОЗДАНИЕ ИГР



Владислав Рехтин

**ИИ ДЛЯ МОБИЛЬНЫХ
РАЗРАБОТЧИКОВ: 50
СПОСОБОВ УСКОРИТЬ
СОЗДАНИЕ ANDROID-ИГР**

«Автор»

2026

Рехтин В. О.

**ИИ ДЛЯ МОБИЛЬНЫХ РАЗРАБОТЧИКОВ: 50 СПОСОБОВ
УСКОРИТЬ СОЗДАНИЕ ANDROID-ИГР / В. О. Рехтин —
«Автор», 2026**

Создавать игры в одиночку — марафон на выживание. Вы одновременно кодер, художник и маркетолог. Пока вы тонете в рутине, конкуренты выпускают хиты. Но 80% задач можно делегировать ИИ! «ИИ для мобильных разработчиков» — жесткий гайд без воды. Только 50 методов, экономящих недели работы. Что внутри: Код: генерация скриптов на C#, отладка багов за минуты. Арт: концепты, UI-киты и тайлсеты без навыков рисования. Геймдизайн: баланс экономики, лор и уровни через промпты. Маркетинг: описание для Google Play и ASO-оптимизация. Заставьте нейросети работать на вас и ускорьте разработку в 3-5 раз! Книга для инди-разработчиков и всех, кто хочет делать игры быстрее.

© Рехтин В. О., 2026

© Автор, 2026

Владислав Рехтин

ИИ ДЛЯ МОБИЛЬНЫХ РАЗРАБОТЧИКОВ: 50 СПОСОБОВ УСКОРИТЬ СОЗДАНИЕ ANDROID-ИГР

ИИ ДЛЯ МОБИЛЬНЫХ РАЗРАБОТЧИКОВ: 50 СПОСОБОВ УСКОРИТЬ СОЗДАНИЕ ANDROID-ИГР

Практическое руководство для инди-разработчиков

ВВЕДЕНИЕ: ПОЧЕМУ ЭТА КНИГА ИЗМЕНИТ ВАШ ПОДХОД К РАЗРАБОТКЕ

Вы держите в руках не просто сборник советов. Это ваш персональный чит-код в мире мобильной разработки.

Давайте честно: создание Android-игры в одиночку — это марафон на выживание. Вы одновременно программист, художник, геймдизайнер, тестировщик и маркетолог. И пока вы боретесь с багами в физике персонажа, конкуренты уже выпустили три игры.

Но что если я скажу вам, что 80% рутины можно делегировать искусственному интеллекту?

Эта книга — результат двух лет экспериментов, ошибок и прорывов. Здесь нет теории ради теории. Каждый из 50 способов я лично тестировал на реальных проектах. Некоторые сэкономили мне недели работы. Другие провалились — и я расскажу об этом тоже, чтобы вы не повторяли моих ошибок.

Для кого эта книга:

- Инди-разработчики, которые делают игры в одиночку или малыми командами
- Начинающие геймдевы, которые хотят ускорить обучение
- Опытные программисты, которые чувствуют, что застряли в рутине

Как использовать эту книгу:

Не читайте её как роман. Откройте оглавление, найдите проблему, которая болит прямо сейчас, и примените решение. Возвращайтесь к другим главам по мере роста проекта.

Готовы ускорить разработку в 3-5 раз? Тогда начинаем.

ЧАСТЬ 1: ГЕНЕРАЦИЯ КОДА И СКРИПТОВ

Способ 1. Мгновенный прототип механики через промпт-инжиниринг

Проблема: Вы хотите проверить идею механики, но написание кода занимает часы.

Решение: Используйте ChatGPT или Claude для генерации базового скрипта за 5 минут.

Практический пример:

Вместо того чтобы писать с нуля систему инвентаря, используйте этот промпт:

Напиши скрипт инвентаря для Unity на C со следующими требованиями:

- Максимум 20 слотов
- Возможность перетаскивания предметов (drag & drop)
- Отображение иконки, названия и количества
- Сохранение в PlayerPrefs
- Методы AddItem, RemoveItem, GetItemCount

Добавь комментарии на русском языке

Важно: Никогда не копируйте код слепо. Всегда проверяйте:

- Работает ли логика
- Нет ли уязвимостей
- Соответствует ли вашему стилю кода

Результат: Экономия 2-4 часов на прототипирование.

Способ 2. Дебаггинг с ИИ-ассистентом

Проблема: Вы застряли на баге уже третий час.

Решение: Скопируйте ошибку и контекст в Claude или ChatGPT.

Правильный промпт:

У меня ошибка в Unity:

[текст ошибки]

Контекст:

- Версия Unity: 2022.3
- Я пишу 2D платформер
- Ошибка возникает когда игрок прыгает на движущуюся платформу
- Вот мой код контроллера персонажа: [код]

Что может быть причиной? Предложи 3 варианта решения.

Почему это работает: ИИ видит паттерны, которые вы упускаете из-за "замыленного глаза".

Результат: Сокращение времени дебаггинга на 60-70%.

Способ 3. Автоматическая генерация UI-скриптов

Проблема: Создание меню, настроек, экранов загрузки — рутина.

Решение: Попросите ИИ написать шаблонные скрипты.

Пример промпта:

Напиши скрипт главного меню для Android-игры на Unity:

- Кнопки: Играть, Настройки, Выход
- Анимация появления кнопок (fade in)
- Звук клика на кнопке
- Адаптация под разные разрешения экрана
- Сохранение выбранного языка

Результат: Готовое меню за 15 минут вместо 3 часов.

Способ 4. Рефакторинг кода через ИИ

Проблема: Ваш код работает, но он грязный и сложный для поддержки.

Решение: Попросите ИИ оптимизировать и очистить код.

Промпт:

Вот мой скрипт управления камерой. Он работает, но:

- Слишком много вложенных условий
- Магические числа
- Нет разделения ответственности

Перепиши его, сделав чище и производительнее. Объясни изменения.

[код]

Результат: Код становится поддерживаемым, производительность растёт на 10-20%.

Способ 5. Генерация системных скриптов (менеджеры, контроллеры)

Проблема: Создание GameManager, AudioManager, SaveSystem — скучно и долго.

Решение: ИИ отлично справляется с шаблонным кодом.

Промпт:

Напиши singleton GameManager для Unity со следующими функциями:

- Отслеживание очков игрока
- Управление состоянием игры (меню, игра, пауза, конец)
- Сохранение/загрузка прогресса
- События для UI (OnScoreChanged, OnGameStateChanged)

Используй паттерн Observer для событий.

Результат: Готовая архитектура за 20 минут.

Способ 6. Создание шейдеров через текстовое описание

Проблема: Вы не знаете Shader Graph или HLSL, но нужен визуальный эффект.

Решение: Опишите эффект словами — получите код шейдера.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.