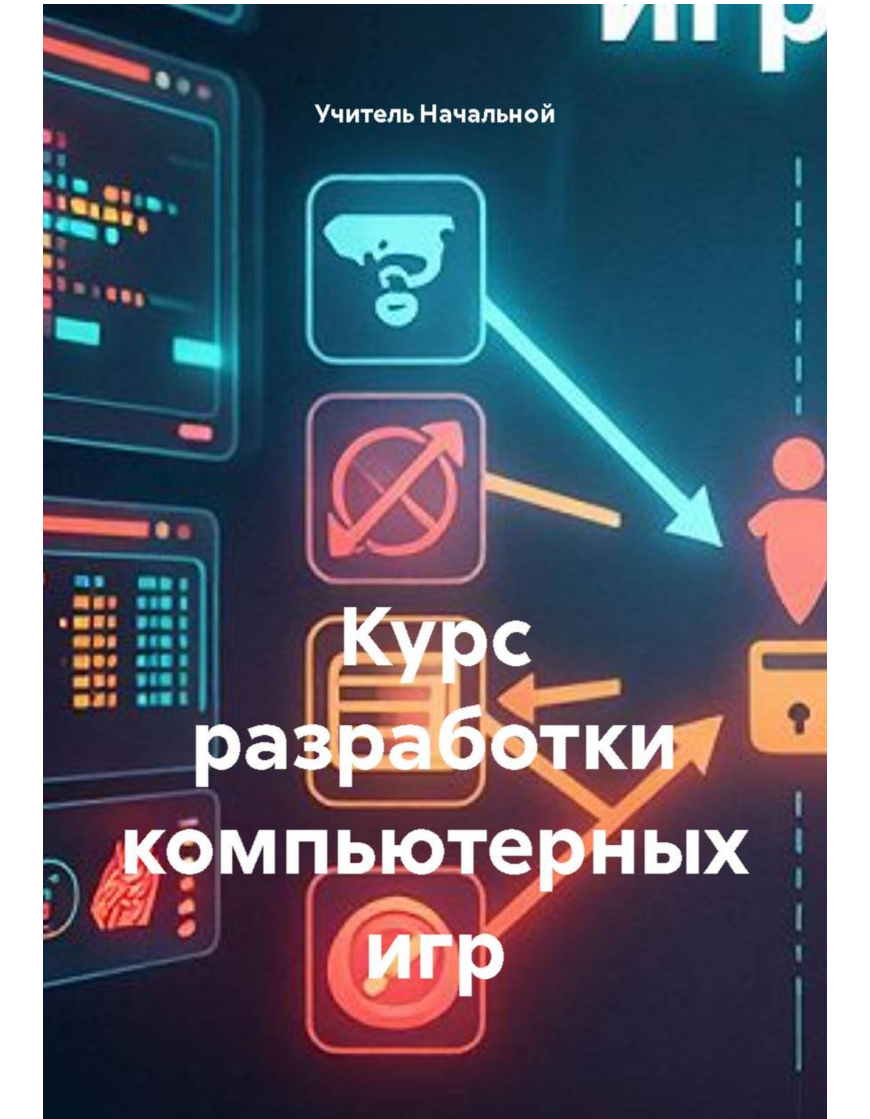


Учитель Начальной

The background is a dark blue gradient with various glowing icons and arrows. On the left, there are three computer monitors displaying colorful data or code. In the center, there are four square icons: a blue one with a camera, a red one with a crossed-out circle and an arrow, an orange one with a document, and a red one with a target. Arrows of different colors (blue, orange, red) point from these icons towards a stylized orange person icon on the right. The text 'Курс разработки компьютерных игр' is overlaid in the center in a large, white, sans-serif font.

Курс разработки компьютерных игр

Учитель Начальной Курс разработки компьютерных игр

<https://litres.ru/74566499>

SelfPub; 2026

Аннотация

Курс разработчика компьютерных игр» — понятное практическое руководство для тех, кто хочет научиться создавать игры с нуля. Книга шаг за шагом проводит читателя через весь путь: от идеи и основ геймдизайна до программирования, работы с движками, физики, анимации, интерфейса, искусственного интеллекта врагов, звука, сетевой игры, оптимизации и публикации готового проекта.

Каждая из 20 глав объясняет тему простым языком, приводит примеры кода (в основном на C# для Unity) и предлагает конкретные задания с разбором решений. Читатель узнаёт, как устроена индустрия, как строить уровни, сохранять прогресс, не допускать типичных ошибок и доводить игры до конца.

Курс подходит новичкам и тем, кто уже пробовал делать игры, но хочет системных знаний. Цель — не только понять теорию, а собрать портфолио из законченных небольших проектов и уверенно двигаться дальше в инди-разработку или студию.

Содержание

Содержание	5
Введение	6
Глава 1. Введение в индустрию компьютерных игр	7
1.1. История и современное состояние индустрии	7
1.2. Роли в команде разработки	8
1.3. Жизненный цикл игрового проекта	9
1.4. Игровые движки и инструменты	10
Ключевые термины	11
Практические задания	12
Глава 2. Основы геймдизайна	13
2.1. Что такое геймдизайн	13
2.2. Модель MDA	14
2.3. Игровой цикл (Core Loop)	15
2.4. Баланс и сложность	16
2.5. Документация: GDD	17
Практические задания	18
Глава 3. Математика для разработчика игр	19
3.1. Зачем нужна математика	19
Конец ознакомительного фрагмента.	20

Учитель Начальной Курс разработки компьютерных игр

КУРС РАЗРАБОТЧИКА КОМПЬЮТЕРНЫХ ИГР

Практическое руководство от основ до профессионального уровня

20 глав • Расширенные практические задания • Без таблиц
Издание 2.0 • 2026

Содержание

Введение

Глава 1. Введение в индустрию компьютерных игр

Глава 2. Основы геймдизайна

Глава 3. Математика для разработчика игр

Глава 4. Основы программирования для игр

Глава 5. Знакомство с игровыми движками

Глава 6. Управление персонажем и ввод

Глава 7. Физика в играх

Глава 8. Анимация и скелетная система

Глава 9. Пользовательский интерфейс (UI/UX)

Глава 10. Искусственный интеллект врагов и NPC

Глава 11. Уровеньный дизайн и построение мира

Глава 12. Звук и музыка в играх

Глава 13. Системы сохранения и данных

Глава 14. Сетевая игра и мультиплеер

Глава 15. Оптимизация производительности

Глава 16. Процедурная генерация

Глава 17. Работа с графикой и шейдерами

Глава 18. Архитектура кода и паттерны

Глава 19. Тестирование, отладка и пайплайн

Глава 20. Публикация, маркетинг и карьера

Заключение

Приложение. Полезные ресурсы

Введение

Добро пожаловать во второе издание «Курса разработчика компьютерных игр». Это практическое руководство проведет вас от полного новичка до специалиста, способного самостоятельно создавать коммерческие игровые проекты.

Индустрия видеоигр — одна из самых динамичных отраслей ИТ. Спрос на квалифицированных разработчиков стабильно высок, а путь в профессию открыт каждому, кто готов системно учиться и много практиковаться.

В этом издании значительно расширены практические задания. Каждая глава содержит 6–8 заданий разной сложности: от простых упражнений до мини-проектов. Рекомендуется выполнять задания последовательно и сохранять результаты в портфолио.

Вы изучите весь цикл разработки: геймдизайн, математику, программирование, движки Unity и Unreal, физику, анимацию, UI, ИИ, уровни, звук, сеть, оптимизацию, процедурную генерацию, архитектуру кода и публикацию.

Рекомендуемый темп — одна-две главы в неделю с обязательным выполнением практических заданий. К концу курса у вас будет портфолио из нескольких законченных проектов.

Начинаем.

Глава 1. Введение в индустрию компьютерных игр

1.1. История и современное состояние индустрии

Компьютерные игры прошли путь от простых аркад 1970-х (Pong, Space Invaders) до сложных AAA-проектов с бюджетами в сотни миллионов долларов и инди-хитов, созданных небольшими командами.

1970–1980-е: аркады, домашние консоли (Atari, NES), появление персональных компьютеров

1990-е: 3D-графика, шутеры от первого лица (Doom, Quake), ролевые игры

2000-е: онлайн-игры, MMORPG, цифровая дистрибуция (Steam)

2010–2020-е: мобильные игры, free-to-play, live-service, VR/AR, инди-бум

Сегодня индустрия включает AAA-студии, mid-core и casual-сегменты, инди-разработчиков, мобильный рынок и новые платформы.

1.2. Роли в команде разработки

Современный игровой проект — это командная работа.

Основные роли:

Геймдизайнер — механики, баланс, прогрессия, GDD

Программист (gameplay, engine, tools, network, AI)

Художник (2D/3D, концепт, UI, VFX, анимация)

Саунд-дизайнер и композитор

Продюсер / менеджер проекта

QA-тестер

Технический художник (Tech Artist)

В инди-командах один человек часто совмещает несколько ролей.

1.3. Жизненный цикл игрового проекта

Концепт и пре-продакшн: идея, прототип, GDD, оценка

Продакшн: вертикальный срез → альфа → бета

Полировка и тестирование

Релиз: билды, магазины, маркетинг

Пост-релиз / live-ops

1.4. Игровые движки и инструменты

Unity — C#, кроссплатформенный, отличен для 2D/3D и мобильных

Unreal Engine — C++ и Blueprints, мощная графика, AAA

Godot — открытый, лёгкий, GDScript/C#

Git, Jira/Trello, FMOD/Wwise, Asset Store

Ключевые термины

AAA — крупные высокобюджетные проекты

GDD (Game Design Document) — документ геймдизайна

Vertical Slice — полностью проработанный кусок игры

MVP — минимально жизнеспособный продукт

Практические задания

Составьте список из 15 игр разных жанров и эпох. Для каждой укажите год выхода, платформу и одну ключевую инновацию.

Выберите 3 любимые игры. Для каждой опишите, какие роли (программист, художник, геймдизайнер и т.д.) внесли наибольший вклад и почему.

Нарисуйте схему жизненного цикла гипотетической инди-игры на 6 месяцев (этапы + примерные сроки).

Установите Unity Hub и создайте пустой 3D-проект. Изучите интерфейс: Hierarchy, Inspector, Project, Scene, Game.

Установите Unreal Engine (или Epic Games Launcher) и создайте пустой проект Third Person. Сравните интерфейс с Unity.

Зарегистрируйтесь на itch.io и Steamworks (или хотя бы изучите требования публикации). Запишите 5 ключевых отличий платформ.

Найдите 3 вакансии junior game developer / gameplay programmer. Выпишите требуемые навыки и сравните с тем, что уже знаете.

Мини-проект: напишите одностраничный «pitch» своей будущей игры (название, жанр, целевая аудитория, уникальная особенность, платформа).

Глава 2. Основы геймдизайна

2.1. Что такое геймдизайн

Геймдизайн — проектирование правил, систем и опыта игрока. Хороший геймдизайнер думает о решениях игрока, эмоциях и мотивации продолжать играть.

2.2. Модель MDA

Mechanics — правила, системы, доступные действия

Dynamics — поведение систем во время игры

Aesthetics — эмоции и ощущения игрока

Дизайнер начинает с желаемой эстетики и подбирает механики.

2.3. Игровой цикл (Core Loop)

Повторяющийся цикл: действие → награда → развитие → новое действие. Примеры: Match-3, шутер, стратегия, roguelike.

2.4. Баланс и сложность

Кривая сложности

Риск / награда

Доминирующие стратегии и контр-пики

Онбординг и обучение через геймплей

2.5. Документация: GDD

Обзор и high concept

Целевая аудитория и платформы

Core gameplay и системы

Прогрессия и мета-игра

Нарратив и мир

Технические ограничения

Практические задания

Разберите любимую игру по модели MDA: выпишите 5 механик, 3 динамики и 4 эстетики.

Опишите core loop своей будущей игры на 1 страницу (схема + текст).

Напишите one-pager (high concept) игры: название, жанр, платформа, аудитория, уникальная фишка, референсы.

Придумайте систему прогрессии для RPG или шутера (уровни, навыки, оружие). Укажите, как избегаете power creep.

Составьте таблицу баланса для 4 видов оружия (урон, скорострельность, магазин, отдача, цена) — но оформите её списком, без таблицы.

Напишите 10 правил хорошего онбординга на примере любой известной игры.

Создайте краткий GDD (3–5 страниц) для небольшой 2D-игры: платформер или top-down шутер.

Мини-проект: проведите «бумажный прототип» механики (карты, кубики, настолка) и опишите, что узнали о балансе.

Глава 3. Математика для разработчика игр

3.1. Зачем нужна математика

Движение, столкновения, камеры, ИИ, физика, графика
— всё это математика в реальном времени.

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.