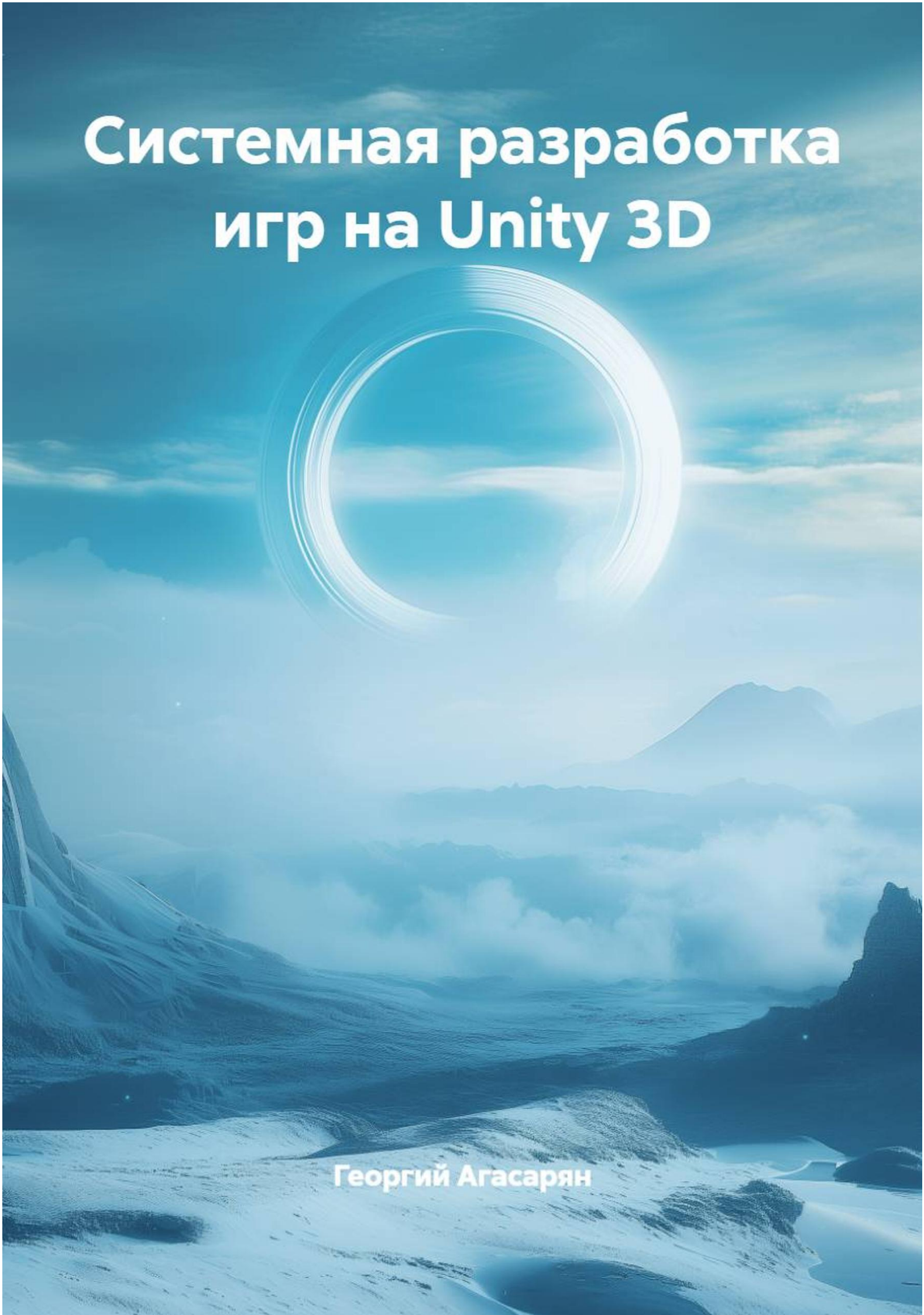
Системная разработка игр на Unity 3D



Георгий Агасарян

Георгий Агасарян

Системная разработка игр на Unity 3D

«Автор»
2026

Агасарян Г.

Системная разработка игр на Unity 3D / Г. Агасарян — «Автор», 2026

Знать Unity — ещё не значит уметь выстроить разработку целой игры. Когда скрипты, механики и сцены должны стать единым проектом, на первый план выходят проектирование, архитектура и последовательность решений.«Системная разработка игр на Unity» — книга для тех, кто уже освоил движок и готов создавать целостные игровые проекты. Она проводит читателя от геймдизайн-документа и декомпозиции задач до реализации механик, создания локаций и объединения всех элементов в согласованную систему.В центре внимания — связь идей и кода: как превратить идею в план работы, определить связи и взаимодействие игровых систем, организовать структуру проекта и учесть ограничения. Вы узнаете не только как реализовать задуманное, но и как выбирать решения, проверять их на практике и закладывать возможности для развития игры.Это не знакомство с интерфейсом Unity и не сборник разрозненных рецептов, а системный подход к созданию игры, в которой механики, код и локации работают на общий результат.

© Агасарян Г., 2026
© Автор, 2026

Георгий Агасарян
Системная разработка игр на Unity 3D

Глава 1. Создание дизайн документа

Дизайн документ это...

Дизайн-документ — не сочинение о будущей игре, а рабочая система решений. Дисдок помогает команде или одному разработчику, одинаково понимать цель, правила, ограничения и критерии готовности. Книга построена от общего к частному: сначала вы формулируете видение, затем описываете игровой цикл и механики, после этого связываете дизайн с реализацией в Unity 3D и производственным планом.

Результат обучения

Сформулировать идею: аудитория, платформа, фантазия игрока, уникальность и ограничения.

Построить игровой цикл: понять, какие действия игрок повторяет и почему они остаются интересными.

Описывать проверяемые механики: через состояния, правила, параметры, исключения и критерии приёмки.

Наложить дизайн на Unity 3D: выделить данные, компоненты, ScriptableObjects, сцены, префабы и события.

Подготовить жанровый мини-GDD: для вашей игры, на примере Souls-like, стелса, песочницы, RPG, FPS или хоррора.

Рекомендуемый маршрут

Прочитайте страницы до раздела 2 и заполните одностраничный концепт.

Выберите один жанровый пример из изложенных в конце главы и замените учебные значения своими.

Соберите прототип одной петли длительностью 3–10 минут.

Проведите минимум пять наблюдаемых тестов и обновите документ, учитывая факты.

Перед производством проверьте чек-листы на последней странице главы.

Главный принцип - Документ должен уменьшать количество неоднозначных устных договорённостей. Если раздел не влияет на решение, оценку или тест — его можно сократить.

Условные обозначения

Решение — То, что команда обязуется реализовать в текущей версии.
Гипотеза — Предположение, которое нужно проверить прототипом или тестом.
Параметр — Число или настройка, вынесенная из кода и доступная для баланса.
Критерий — Наблюдаемое условие, по которому функция считается готовой.

1. Что такое GDD и из каких документов он состоит

GDD (Game Design Document) — центральное описание опыта игрока и правил игры. В маленькой команде это может быть один файл с таблицами. В большой — набор связанных страниц.

Минимальный комплект

One-pager — Что это за игра и для кого? 1 страница.

Vision / pillars — Какой опыт нельзя потерять? 1–2 страницы.

Core loop — Что игрок повторяет? 1 схема + пояснения

Спецификации механик — Как точно работает каждая система? 1–4 страницы на систему.

Контент-дизайн — Какие уровни, враги, предметы и задания нужны?Таблицы и шаблоны.

UX-спецификация — Как игрок управляет и получает обратную связь? Экраны, состояния, тексты.

Tech notes — Как это собирается в Unity?Компоненты, данные, события.

План тестирования — Какие гипотезы и метрики проверяем? Чек-лист + журнал.

Три уровня подробности

Уровень 1 — зачем: цель, эмоция, аудитория, дизайнерский принцип.

Уровень 2 — что: действия, состояния, правила, контент, интерфейс.

Уровень 3 — как проверить: числа, сценарии, телеметрия, критерии приёмки.

Ошибка новичка. Написать сюжет, список оружия и лор раньше, чем определены действия игрока и условия выбора. Начинайте с глаголов: исследовать, скрываться, целиться, обменивать, рисковать.

Правило актуальности

У каждого раздела дисдока, должны быть ответственное лицо, текущий статус и дата изменения. Устаревшее решение либо обновляют, либо явно помечают как архивное. Команда не должна угадывать, какой из двух противоречащих абзацев верен.

2. Одностраничный концепт и дизайн-столпы

Шаблон one-pager

Рабочее название — Как коротко ссылаться на проект? Например «Пепельный Предел» или «Spectre»

Жанр и камера — Как игрок видит и читает мир? Action RPG, вид от третьего лица.

Платформа — Где и каким вводом играют? ПК, клавиатура/мышь и геймпад.

Аудитория — Какой опыт у игрока? Знаком с экшен-RPG, любит имме мир.

Фантазия — Кем игрок себя ощущает? Охотник, побеждающий знанием.

Крючок — Чем игра отличается? Следы прошлых попыток меняют арену.

Сессия — Сколько длится типичный заход? 20–40 минут

Объём MVP — Что доказывает жизнеспособность? 1 хаб, 1 уровень, 3 врага, босс

Не делаем — Что сознательно исключено? Открытый мир, PvP, транспорт

Дизайн-столпы

Столп — короткое правило выбора. Он помогает решить спор между двумя хорошими идеями. Обычно достаточно трёх или четырёх.

Например:

Читаемый риск — опасность заметна заранее; наказание следует за решением, а не за скрытой информацией.

Знание сильнее числа — понимание паттернов важнее бесконечного роста характеристик.

Короткий путь после открытия — исследование мира регулярно создаёт полезные сокращения.

Проверка столпа

Хороший столп влияет на конкретный выбор. Вопрос «добавлять ли случайные мгновенные ловушки?» получает ответ: нет, если они нарушают читаемый риск.

Фраза «красивая атмосфера» слишком общая: она не задаёт критерия решения в отличие от сформулированного столпа.

Практика на 20 минут. Напишите по одному предложению, состоящему из крючка и трёх столпов. Затем удалите все прилагательные, которые невозможно проверить наблюдением игрока.

Это значит что нужно убрать из описаний предметов, локаций или персонажей все субъективные, оценочные и абстрактные слова, которые игрок не может подтвердить своими глазами или ушами прямо в игре.

Что считается «непроверяемым»?

Слова, которые описывают чувства, историю или оценку автора, но не дают конкретной визуальной или физической информации.

Субъективные оценки: «красивый», «ужасающий», «великолепный», «уютный». (То, что автору кажется уютным, игроку может показаться тесным).

Абстрактные понятия: «древний», «магический», «проклятый», «забытый». (Если на мече просто написано «древний», но он выглядит как новый, игрок не может это проверить наблюдением).

Скрытые свойства: «тяжелый» (если игрок не видит, как персонажу тяжело), «острый» (игрок не видит как им что то разрезается), «громкий» (если громкость такая же как у окружения).

2. Что считается «проверяемым»?

Это факты, которые очевидны из графики, звука или механики.

Цвет и форма: красный, квадратный, зазубренный.

Состояние материала: ржавый, влажный, треснувший, горящий.

Материал: железный, деревянный, шерстяной.

3. Игровой цикл и решения

Игровой цикл показывает повторяющуюся последовательность: действие обратная связь изменение состояния новое решение. Он должен объяснять, почему игрок хочет сделать следующий шаг.

Три масштаба цикла

Секунды — заметить, прицелиться, выстрелить, оценить попадание. сокращая : ввод, тайминг, feedback.

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.