Документация проекта: Unity — ThirdPersonShooter

1. Общая информация

Название: Unity — ThirdPersonShooter
Платформа: Unity (6.3.8f1 LTS)
Жанр: Third-Person Shooter (TPS)
Статус: Прототип, учебный проект
Цель: Демонстрация фундаментальных механик TPS-игры, структурированное обучение разработке на Unity. (GitHub)

2. Стратегическая цель

Проект создан как **пошаговое руководство** по созданию TPS-игры:

- посвящён изучению архитектурных подходов,
- реализации ввода, анимаций, стрельбы, инвентаря, ИИ врагов,
- акцентируется внимание на **модульности** и **расширяемости**.

Проект подходит для:

- портфолио разработчика,
- open-source репозитория,
- учебных целей (самообразование, преподавание).

3. Структура репозитория

/Assets
/Docs
/Packages
/ProjectSettings
README.md

Assets/ — исходники Unity (сцены, скрипты, ассеты).
Docs/ — документация (если присутствует — см. внутри).
ProjectSettings/ — настройки Unity-проекта.
README.md — базовое описание.

4. Описание по разделам (сценам)

Проект разделён на логические **сцены-уроки**, каждая демонстрирует конкретные механики.

4.1 Сцена 1 — Базовый контроллер персонажа

Задача:
Реализовать управление персонажем и камерой.
Результат:

- игрок способен перемещаться в 3D пространстве,
- базовый TPS-контроллер,
- настройка чувствительности ввода.

Метрики: корректный отклик WASD + мышь.

4.2 Сцена 2 — Аудио шагов по поверхности

Задача: добавить контекстный звук шагов.

Решение:

- Raycast вниз для определения поверхности,
- звуки шагов привязаны к типу поверхности.

Принцип: логика аудио вынесена из контроллера движения.

4.3 Сцена 3 — Сбор предметов и инвентарь

Задача: показать взаимодействие с предметами.

Особенности:

- ScriptableObject как модель данных,
- инвентарь управляет предметами,
- UI-подсказки: F для сбора.

Выделение: отделение визуального объекта от данных.

4.4 Сцена 4 — Механика стрельбы

Задача: реализовать оружие.

Подход:

- Hitscan (Raycast)
- Projectile (физический снаряд)

Архитектурное решение:
интерфейсы + наследование для абстракции оружия.

4.5 Сцена 5 — Квестовая система

Задача: создать простой квест (ключ + дверь).

Реализация:

- проверка инвентаря до открытия двери,
- анимация двери,
- UI-подсказки.

4.6 Сцена 6 — ИИ врагов (FSM)

Задача: показать простую систему AI.

Подход: конечный автомат (FSM)
Баланс:

- патрулирование,
- преследование,
- атака.

Архитектура: FSM может быть реализован как внутри компонента, так и через отдельные состояния. (GitHub)

5. Архитектурные принципы

5.1 Модульность

Проект разделён по механикам — облегчает тестирование и реюз.
Преимущества:

- меньшая связанность,
- независимое тестирование сцен.

5.2 Чистота кода

Скрипты ориентированы на:

- SOLID-принципы,
- разделение визуального представления и логики,
- использование ScriptableObject для данных.

5.3 Расширяемость

Лёгкое добавление новых механик:

- новые типы оружия,

- дополнительные AI-состояния,
- расширение инвентаря.

6. Технологии

Компонент	Инструмент/Технология
Игровой движок	Unity3D
Язык программирования	C#
Управление вводом	Unity Input System / Custom
Сцены	Unity Scenes
Скрипты	C# Scripts
Данные	ScriptableObject
AI	FSM (Finite State Machine)

7. Требования и сборка

Минимальные требования:

- Unity Editor 6.x или новее
- Платформа: Windows / Mac / Linux (зависит от билд-настройки)

Сборка:

- Открыть проект в Unity, убедиться, что **Assets** и **ProjectSettings** загружены, построить сборку через Build Settings.

8. Руководство разработчикам

8.1 Добавление новой механики

- Создайте новую сцену или расширьте существующую.
- Реализуйте логику в отдельном классе/скрипте.
- Обеспечьте слабую связанность через интерфейсы.
- Добавьте конфигурацию через ScriptableObjects.

8.2 Стандарты кодирования

- PascalCase для классов и методов.
- camelCase для полей.
- Комментарии XML для публичных API.
- Избегать «магических констант», использовать конфиги.

9. Лицензия

Убедитесь в указании лицензии в репозитории. Если лицензия отсутствует — рекомендуем добавить файлы **LICENSE** (MIT / Apache 2.0) для open-source.

10. Перспективы развития проекта

Возможные улучшения

- Система анимаций через Animator Controller с переходами и Blend Trees.
- Продвинутая система ИИ (NavMesh, Steering).
- Инвентарь с экипировкой и экипом.
- UI-панель прогресса здоровья и боеприпасов.
- Сетевой режим (Multiplayer).