**Как работает генератор динамических квестов** Динамические квесты подстраиваются под контекст, анализируя текущие условия игры (игроков, NPC, локации) и генерируя уникальные задачи. Например, если игрок часто отказывается от заданий, квест может включить мотивацию (дополнительную награду), а если NPC атакован, квест может измениться на "помощь в защите". Это похоже на адаптивные системы в RPG, таких как The Elder Scrolls, где квесты зависят от прогресса игрока.**Данные на входе (Input)**Для контекстной адаптации нужны данные о состоянии игры. Форматы: JSON, ScriptableObject или runtime данные в Unity.

* **Текущий контекст игрока**:
* Уровень, инвентарь, репутация. Пример JSON: { "игрок": { "уровень": 5, "предметы": ["меч"], "репутация": 10 } }.
* **Состояние NPC**:
* Имя, здоровье, отношение к игроку. Пример: { "npc": { "name": "Торговец", "здоровье": 80, "отношение": "нейтральное" } }.
* **Локация и события**:
* Место (лес, город), активные события (атака, мир). Пример XML: <location><name>Forest</name><event>monster\_attack</event></location>.
* **Шаблоны квестов**:
* Базовые цели (fetch, kill) с параметрами (уровень сложности). Пример ScriptableObject: QuestTemplate с полями objectiveType, трудность.

**Данные на выходе (Output)**Выход — адаптированный квест с учетом контекста. Форматы: JSON, ScriptableObject, или объекты времени выполнения.

* **Структура квеста**:
* Цели, шаги, награды. Пример JSON: { "quest": { "id": "Q002", "title": "Save Trader", "objectives": ["defend\_npc"], "rewards": ["gold:150"], "context": "monster\_attack" } }.
* **Диалоги с ветвями**:
* Текст с динамическими вставками. Пример XML: <dialog><node>Помогите мне, меня атаковали!</node><branch condition="refuse">Трейдер сердит.</branch></dialog>.
* **Состояния и реакции**:
* Active, Completed, с динамической логикой (например, NPC меняет отношение). Пример ScriptableObject: QuestState с полем npcРеакция.

**Реализация в Unity**Для создания адаптивного генератора в Unity используйте комбинацию ScriptableObjects, C# скриптов и, возможно, ML-Agents для динамики.**Шаги реализации**

1. **Создайте базу данных**:

* Используйте ScriptableObject QuestTemplate для шаблонов (например, QuestDatabase.asset). Храните входные данные (NPC, локации) в JSON или Resources.
* Пример: { "templates": [{ "type": "defend", "minLevel": 5, "npc": "Trader" }] }.

1. **Генератор квестов**:

* C# скрипт (QuestGenerator.cs):
* Анализирует контекст (player.level, npc.health).
* Генерирует квест. Пример:
* Шарп
* using UnityEngine;
* using System.Collections.Generic;
* [System.Serializable]
* public class QuestContext {
* public int playerLevel;
* public string npcState; *// "attacked", "neutral"*
* }
* [System.Serializable]
* public class QuestInstance {
* public string id;
* public string title;
* public List<string> objectives;
* public Dictionary<string, string> dialogs;
* }
* public class QuestGenerator : MonoBehaviour {
* public QuestContext context;
* public QuestInstance GenerateQuest() {
* QuestInstance quest = new QuestInstance();
* quest.id = "Q" + Random.Range(1, 1000);
* if (context.npcState == "attacked") {
* quest.title = "Save " + context.npcState + " NPC";
* quest.objectives = new List<string> { "defend\_npc" };
* quest.dialogs = new Dictionary<string, string> {
* {"start", "Help me, I’m under attack!"},
* {"refuse", "You’ll regret this!"}
* };
* } else {
* quest.title = "Fetch Item";
* quest.objectives = new List<string> { "fetch\_item" };
* quest.dialogs = new Dictionary<string, string> { {"start", "Bring me an artifact."} };
* }
* return quest;
* }
* }
* Интеграция LLM: Используйте LLMUnity или OpenAI API для динамических диалогов (промпт: "Generate quest dialog for used trader, player level 5").

1. **Адаптация под контекст**:

* Используйте ML-Agents (F4) для обучения NPC реагировать на действия игрока (например, отказ = -1 награда, помощь = +1 награда).
* Сохраняйте состояние в QuestState (ScriptableObject) с полем динамическаяреакция.

1. **Плагины и ассеты**:

* **Квест Система Про** ($50, Asset Store): Поддержка ветвей и состояний.
* **Диалоговая система для единства** ($65): Для диалогов с контекстными ветвями.
* **LLMUnity**: Бесплатно, для генерации текста.

1. **Выходные ассеты**:

* Экспортируйте квесты как JSON или ScriptableObjects для модов. Пример выхода: { "id": "Q002", "dialog": "Помогите мне!", "reaction": "angry\_if\_refuse" }.
* Подключите к Unity Asset Store или как мод (например, для игр вроде Factorio).

**Что добавить к квестовой системе**

* **Контекстные триггеры**: Реакция на события (атака, отказ) через события Unity (Event System).
* **Динамика NPC**: Интеграция с ML-Agents для адаптации поведения.
* **Привязка к игре**: Подключите к существующей системе (например, как мод для Diablo-like игры), экспортируя квесты как плагины.