**README — NpcProfile**

Этот документ — подробная спецификация **NpcProfile** (ScriptableObject) для гуманоидных NPC (солдат, командир, торговец и т. п.). Он описывает *что за персонаж*, с чем он появляется, как себя ведёт, и как его корректно подключать к лагерю и спавнеру.

Важно: **NpcProfile — только для NPC.** Фауну (волки и т. д.) описывает CreatureProfile.

**Назначение**

* Дать **единообразный источник правды** об одном типе NPC: префаб, фракция, роль, статы, восприятие, базовый бой, поведение в лагере, лоадаут и лут.
* Позволить дизайнерам **балансировать в инспекторе** без правок кода.
* Упростить спавн: NpcSpawner берёт NpcProfile, создаёт агента и применяет настройки.

**Где лежит и как создать**

* Файл класса: NpcProfile.cs.
* Создание ассета: Create → World/NPC/NpcProfile.
* Рекомендованная папка: Assets/Game/Actors/Instances/NPC/.

Именование ассетов: npc\_<роль>\_<фракция>[\_суффикс]  
Примеры: npc\_guard\_bandits, npc\_commander\_bandits\_t1.

**Поля и как их заполнять**

**1) Identity**

* **id** — *уникальный строковый идентификатор* (для сохранений/телеметрии).
  + Формат: npc.<faction>.<role>.<name>.v1 (пример: npc.bandits.guard.spear.v1).
  + Не меняется после релиза контента; при изменениях — увеличивайте суффикс .v2.
* **displayName** — имя для UI (локализуемая строка).
* **portrait** — (опционально) иконка в диалогах/окне взаимодействия.

**2) Prefab & Faction**

* **prefab** — префаб NPC с компонентами:
  + контроллер передвижения/навигации,
  + анимация (Animator или эквивалент),
  + боевой модуль (если нужен),
  + «мозг» (BT/FSM/скрипт, читающий brainId).
* **faction** — Neutral / Player / Bandits / Merchants / Animals (влияет на «свой–чужой»).
* **role** — Guard / Patrol / Trader / Healer / Worker / Leader.
* **npcTags** — флаги (например, Trader, Leader), для выборки и условий в AI/квестах.

**3) Perception (восприятие)**

* **viewDistance** — дальность зрения (м).
* **viewAngle** — угол зрения (°).
* **hearingRadius** — радиус «слышит шум».

Рекомендации:  
Guard — viewDistance 12–15, viewAngle 120–140.  
Trader/Worker — короче и уже.

**4) Stats/Combat (минимум боя)**

* **health** — базовое ХП.
* **moveSpeed** — скорость перемещения.
* **armor** — плоское или процентное ослабление (решается боевой системой).
* **attackDamage** — урон атаки.
* **attackRate** — атак/сек.
* **attackRange** — дальность удара/стрельбы.
* **projectileSpeed** — >0 для дальников (0 — ближник).

Альфа-1: простая ближняя атака у Soldier, у Leader — можно оставить нули (не драться).

**5) Brain (поведение)**

* **brainId** — ключ поведения (строка: ID BehaviorTree/FSM/сценария).
  + Примеры: bt.guard, fsm.trader, bt.patrol.simple.
* Скорости:
  + **patrolSpeedMul** — множитель скорости в патруле.
  + **chaseSpeedMul** — множитель скорости в погоне.
* Территория:
  + **leashRadius** — «домашний» радиус (не уходит дальше).
  + **wanderRadius** — радиус блуждания внутри дома.

**6) Поведение в лагере (якоря)**

* **preferredAnchor** — где спавниться/стоять относительно лагеря:
  + Center (центр), Fire (костёр), Gate (ворота), Tents (вокруг палаток), Perimeter (по окружности), Any (любой).
* NpcSpawner старается удовлетворить якорь; если нет нужной метки — использует фолбэк (Center/Any).

**7) Loadout (стартовая выдача)**

* **loadout[]** — список предметов, которые NPC получает при спавне:
  + itemId — ID предмета (совместимый с вашей инвентарной системой),
  + count — количество,
  + chance — вероятность выдачи (0..1).
* Советы:
  + Никогда не ставьте 100 разных слотов — объединяйте «стопки».
  + Для вариативности: несколько альтернативных записей с chance.

**8) Interactions (торговля/квесты/голос)**

* **dialogueRef** — ссылка на объект с диалогами (если есть).
* **isTrader** — галочка торговца.
  + **traderInventoryId** — ID ассортимента (совместим с вашим магазином).
* **isQuestGiver** — выдаёт квесты (командир).
* **voiceSet[]** — набор AudioClip для реплик (опционально).

**9) Loot (дроп с NPC)**

* Наследуется из базового профиля: loot[] (ID предмета, countRange, chance).
* Простые настройки для Альфы-1: малый шанс на «фляга/еда/монета» у солдат.

**Интеграция в лагерь и спавн**

* **CampProfile** хранит ссылку на NpcSpawnList (пак NPC).
* **NpcSpawnList.Entry** ссылается на **NpcProfile** и задаёт count, anchor, spacing и т. д.
* **NpcSpawner**:
  1. берет center/radius лагеря,
  2. по каждой записи NpcSpawnList вычисляет позиции исходя из anchor и spacing,
  3. инстансит prefab,
  4. применяет faction, role, brainId, stats, perception,
  5. выдаёт loadout,
  6. регистрирует агента в системах (агро/комбат/диалоги).

NPC **не резервируют клетки** через ReservationService (он про запрет природы). Их позиции проверяются на **проходимость/коллизии**, а не на резервацию.

**Мини-примеры заполнения**

**Soldier (Bandits)**

* id: npc.bandits.guard.sword.v1
* displayName: «Бандит»
* prefab: pf\_npc\_bandit\_guard
* faction: Bandits
* role: Guard
* preferredAnchor: Perimeter
* viewDistance: 14, viewAngle: 130, hearingRadius: 8
* health: 100, moveSpeed: 3.5, attackDamage: 12, attackRate: 0.9, attackRange: 1.4
* brainId: bt.guard
* leashRadius: 15, wanderRadius: 3, patrolSpeedMul: 1.0, chaseSpeedMul: 1.2
* loadout: { (item: «меч»,1,1.0), (item: «еда»,1,0.4) }
* loot: { (item: «монеты», 1..3, chance 0.5) }

**Commander (Leader)**

* id: npc.bandits.leader.commander.v1
* role: Leader, npcTags: Leader
* preferredAnchor: Center
* isQuestGiver: ✅
* Combat можно опустить (не дерётся), brainId: bt.leader.idle

**Гайды и «подводные камни»**

1. **Уникальный id.** Не меняйте существующие ID — меняйте версию (…v2).
2. **Совместимость префаба.** В префабе должны быть все компоненты, на которые рассчитывает AI/боёвка (иначе NRE/поведение «слетит»).
3. **Роли и якоря.** Leader почти всегда Center. Guard — Perimeter. Trader — Fire/Tents.
4. **Баланс перцепции.** Слишком большие viewDistance + viewAngle → агро «через пол-лагеря».
5. **Loadout vs Loot.** Не дублируйте ключевой предмет и там и там (или снизьте шансы).
6. **Производительность.** Не делайте пачки с десятками дальников (снаряды/трассеры). Для Альфы-1 — ближник у солдата.
7. **Локализация.** displayName должен быть ключом или подхватываться вашей системой локализации.
8. **Тест якорей.** Если в лагере нет огня/ворот — NpcSpawner должен корректно фолбэкнуть (мы это предусмотрим).

**Чек-лист перед коммитом**

* id уникален, соответствует соглашению об именах.
* prefab задан и открывается без Missing Scripts.
* faction и role осмысленны; npcTags выставлены (если надо).
* perception и stats в разумных пределах (для роли).
* brainId указан и существует в AI-ресурсах.
* preferredAnchor валиден для целевого лагеря.
* loadout корректен (ID предметов существуют).
* loot корректен (min ≤ max, chance в [0..1]).
* Ассет сохранён в правильной папке и имеет осмысленное имя.

**FAQ**

**Q:** Можно ли один NpcProfile использовать в нескольких лагерях?  
**A:** Да. Поведение в лагере определяется якорем и NpcSpawnList. Один и тот же профиль может входить в разные паки.

**Q:** Чем role отличается от npcTags?  
**A:** role — «основная профессия» (одно значение). npcTags — «доп. свойства» (флаги) для фильтрации/условий.

**Q:** Где задаётся патрульный маршрут?  
**A:** В Альфе-1 — через brainId (например, bt.patrol.simple с автогенерацией точек вокруг якоря). Маршруты как набор точек — позже.

**Q:** Делать ли NPC дальниками?  
**A:** Для Альфы-1 — лучше ближники. Дальников можно добавить по одному (командир-стрелок) для проверки.