**README — как управлять пуллингом/плотностью объектов по биомам**

Этот файл объясняет, **где и что менять**, чтобы регулировать количество и раскладку объектов в биомах, а также синхронность со стримингом/пулом.

**Быстрый рецепт (TL;DR)**

1. Открой нужный ассет **BiomeSpawnProfile** (например, BiomeSpawn\_Forest.asset).
2. Внутри списка **rules** найди нужный объект (Oak/Spruce/…):
   * **targetPerChunk** — главное колесо «сколько на чанк».
   * **mode** — стиль раскладки: Clustered / BlueNoise / Uniform.
   * **minDistanceSameType / minDistanceAny** — дистанции (могут мешать «влезть» в таргет).
   * **useNoiseGate + noiseThreshold** — шумовая маска; ниже threshold → гуще.
   * **clusterCountRange** — форма кластеров (только для Clustered).
3. Чтобы разрядить/сгустить **весь биом целиком**, крутите сверху **densityMultiplier**.
4. Если видите рассинхрон по дальности — проверьте objectsChunkSize/Radius в PoolManager и chunkSize в ObjectManager.

**Где что лежит**

* **Правила спавна по биомам:**  
  Assets/ScriptAssets/SpawnRules/BiomeSpawn\_\*.asset (тип: BiomeSpawnProfile)  
  Поля:
  + densityMultiplier — множитель для всех правил биома.
  + rules (List<BiomeObjectRule>):
    - objectType
    - targetPerChunk
    - mode (Clustered / BlueNoise / Uniform)
    - clusterCountRange (для Clustered)
    - minDistanceSameType, minDistanceAny, avoidOtherObjects
    - useNoiseGate, noiseThreshold, noiseScale
    - variants (обычно = числу спрайтов в ObjectData.spriteVariants)
* **Паспорт объекта (визуал/футпринт):**  
  Assets/Game/World/Objects/ObjectData\_\*.asset (тип: ObjectData)  
  Важно для спавна: footprint (большие следы сложнее «влезают»).
* **Генерация/стриминг объектов:**  
  ObjectManager.cs — генерация по чанкам (64×64), загрузка/выгрузка визуала из пула.  
  Ключевые поля:
  + worldSeed, chunkSize (= 64)
  + biomeProfiles (список профилей)
  + objectDatabase (список ObjectData\_\*)
  + objectPool (PoolManagerObjects)
  + cellSize, ySort, objectsSortingLayer
* **Стриминг вокруг игрока:**  
  PoolManager.cs — окно тайлов + окно **объектных чанков**.  
  Поля:
  + radius (в клетках) — окно тайлов
  + objectsChunkSize (== ObjectManager.chunkSize)
  + objectsChunkRadius (радиус в чанках, вычисляется из radius)
* **Пул префабов объектов:**  
  PoolManagerObjects.cs — Get(type)/Release(go).  
  Поля:
  + defaultSpritePrefab — базовый спрайтовый префаб
  + overrides — особые префабы на некоторые ObjectType
  + objectsRoot — контейнер для GO (желательно под тем же WorldRoot, что и тайлы)

**Как именно регулировать плотность**

**1) Главный регулятор — targetPerChunk**

* Где: BiomeSpawnProfile.rules[\*].targetPerChunk
* Что делает: целевое число инстансов **на чанк 64×64**.
* Если «не добирает до таргета», ограничивают **дистанции** и/или **шум** (см. ниже).

**2) Множитель всего биома — densityMultiplier**

* Где: BiomeSpawnProfile.densityMultiplier (верхний уровень)
* Что делает: умножает все targetPerChunk этого биома (удобно для быстрого баланса).

**3) Дистанции**

* Где: rules[\*].minDistanceSameType, rules[\*].minDistanceAny, rules[\*].avoidOtherObjects
* Что делают: минимальный зазор в клетках.
* Советы:
  + Уменьшайте по **0.2–0.5**, если таргет не «влезает».
  + Для **BlueNoise** шаг сетки ≈ minDistanceSameType.

**4) Шумовая маска (Perlin)**

* Где: rules[\*].useNoiseGate, noiseThreshold, noiseScale
* threshold: ниже → «гуще», выше → «реже». Крутите по **±0.05**.
* scale: размер пятен (не столько «сколько», сколько «как выглядит»).

**5) Режим раскладки**

* Где: rules[\*].mode = Clustered / BlueNoise / Uniform
* Влияет на рисунок. Если при Clustered «не влезает», уменьшите clusterCountRange — семян станет больше, спутников меньше → проще разместить.

**Частные моменты и «почему ничего не спавнится»**

* **Биом не совпадает:** профиль применяется **только** в своём биоме.
* **Слишком строгий шум/дистанции/footprint:** начните с:
  + threshold −0.05,
  + minDistance\* −0.2,
  + проверьте footprint в ObjectData.
* **Особые правила:** пальмы в пустыне — только рядом с Oasis/Savanna (проверка в ObjectManager.IsAllowed(...)).
* **Слишком ранняя генерация:** пока BiomeManager.IsBiomesReady == false, чанки **не** кэшируются (в коде уже учтено). При смене профилей удобно вызвать ObjectManager.ClearChunkCache() и затем PoolManager.ForceRefresh().

**Синхронизация с тайлами (видимость и радиусы)**

* ObjectManager.chunkSize == PoolManager.objectsChunkSize (обычно 64).
* Радиус чанков объектов считается из PoolManager.radius (в клетках); без «+1» поясов.
* **Sorting Layers:** тайлы — Ground, объекты — WorldObjects; sortingOrder по Y.
* Оба используют **один** cellSize (берите из одного источника: WorldManager или PoolManagerMainTile).

**Примеры быстрых правок**

* «В лесу слишком редкие дубы»  
  Forest → Oak:
  + targetPerChunk: 45 → **60**
  + если не помогает: noiseThreshold: 0.50 → **0.45**
  + крайний случай: minDistanceSameType: 1.8 → **1.6**
* «Кактусов слишком много в пустыне»  
  Desert → Cactus:
  + targetPerChunk: 35 → **24**
  + noiseThreshold: 0.60 → **0.65**
  + (опц.) minDistanceSameType: 3.0 → **3.2**

**FAQ**

* **Хотим разредить весь биом одинаково:** крутите **densityMultiplier** (например, 0.8).
* **Хотим сохранить рисунок, но изменить общее число:** крутите **targetPerChunk**; не трогайте dist/noise.
* **Хотим другой «рисунок»:** поменяйте mode и clusterCountRange/noiseScale.