

## Vintage Story 1.19–1.20+

- **Розмір чанків і областей:** Чанки мають розмір 32×32×32 блоки, колонка чанків (ChunkColumn) займає 8 таких чанків по висоті (за замовчуванням висота світу 256 блоків) <sup>1</sup>. У грі використовується ієрархія: *MapRegion* (16×16 колонок чанків), *MapChunk* (дані по одній колонці, наприклад карти висот), і *ChunkColumn* (безпосередньо стовпець чанків) <sup>1</sup>. Рівень моря задається як відносна величина (за замовчуванням ~0.43 від висоти світу, тобто приблизно у=110 при висоті 256).
- **Етапи генерації (passes):** Генерація розбита на послідовні етапи (EnumWorldGenPass 0–6) <sup>2</sup>. Pass 0 – None; 1 – Terrain Noise (базова топографія, шари порід, печери, первинні блок-слої); 2 – Terrain Features (рудні прожилки, структури, водойми); 3 – Vegetation (рослинність, кущі, дерева, килими трави, килими квітів, водойми); 4 – NeighbourSunLightFlood (розсівання снігу, оновлення освітлення); 5 – PreDone (окремі дії перед завершенням, спавн мобів); 6 – Done. На етапі **Terrain Noise** формується основна геологія (генерується GenRockStrataNew), на етапі **Terrain Features** – осаджуються копальні родовища.
- **Генерація шарів порід:** Використовується система GenRockStrataNew (клас у VSSurvivalMod). Шари групуються за трьома категоріями: осадові (sedimentary), метаморфічні (metamorphic) і ігнейні (igneous; включаючи підтип вулканічні/volcanic на поверхні). Ігнейні породи завжди формують базовий нижній масив (починаючи з мантиї внизу) та простягаються вгору поки не досягнуть метаморфічного/осадового шару або поверхні <sup>3</sup>. Базальт (Basalt) – ігнейна порода винятково, може генеруватися вище всіх інших шарів (включно з осадовим і метаморфічним) <sup>4</sup>. Мармур (Marble) і обсидіан генеруються «в жилах» у метаморфічному слані і слаті, а сувіт тільки в кратерах (класичні винятки) <sup>4</sup>.
- **Відомі бара (GenRockStrataNew):** Зауважено, що в GenRockStrataNew методі `genBlockColumn` ітерація по списку варіантів порід (rockstrata.json) відбувається в порядку запису, тож перші в списку породи заповнюють колонку першочергово <sup>5</sup>. Це призводить до того, що пізні породи (наприклад, крейда, боксит) зустрічаються набагато рідше, ніж вказано теоретично <sup>5</sup>. Особливий випадок – граніт (Granite): код спеціально „перескакує” суцільні ділянки граніту, але не оновлює лічильник поточної товщини, через що фактичні шари граніту виходять удвічі товщі <sup>6</sup>. Це може призводити до зникнення інших порід вище (наприклад, відсутність пісковиків під шаром подвоєного граніту) <sup>7</sup>.
- **Вертикальна дисторсія і інверсії:** Задача дисторсії шарів (горизонтальні зрушення вітром, ерозією тощо) реалізована шляхом налаштування шумових карт, тож осадовий шар може іноді опускатися під ігнейний або навіть впливати вище. Як наслідок, може виникати інверсія шарів (наприклад, два ігнейних шари над одним) <sup>8</sup>. Система це допускає: базальна таблиця «Max Thickness» по геологічних провінціях (Shield, Platform, Orogen, Basin, Large Igneous Province, Extended Crust) визначає максимально можливі товщини вулканічних, осадових та метаморфічних шарів <sup>9</sup>. Наприклад, для платформи (Platform) осадових – до 40 блоків, метаморфічних – до 10; для басейну (Basin) – до 50 та 15 відповідно; для великих ігнейних провінцій (LIP, Extended) з'являються вулканічні шари до 30 блоків <sup>9</sup>. Через це та геологічні викривлення одним регіоном можливі «штучні» перевороти та дублювати шарів <sup>8</sup>.

- **Провінції та товщини шарів:** Геологічні провінції (задано у конфігах `GeologicProvinces.json`) визначають базові статистики: наприклад, величину колонок (blocks) для кожного типу порід. В таблиці «Max Thickness» видно: Shield не дає осадових/ метаморфічних (0/0), залишаючи тільки ігнейні (255); Platform – 40/10/255; Orogen – 20/10/255; Basin – 50/15/255; LIP/Extended – 20/0/255 та 40/0/255 з 30 блоками вулканічної породи <sup>9</sup>. Ці значення призначені максимальними, далі реалізуються шумовими картами (див. нижче).
- **Поклади (Deposits):** Генерація рудних покладів і інших утворень виконується класом `GenDeposits`. Спершу завантажуються конфігураційні файли `worldgen/deposits/` (описують форму і вид покладів). При ініціалізації створюється **карта деформації форми** (deposit shape distortion map) для уніфікованої побудови форм покладів <sup>10</sup>. Для кожного виду покладу створюються власні **орієнтовані карти руди** (ore maps), а також для похідних (child) покладів <sup>10</sup>. При генерації кожного регіону скликається `GenDeposits.OnMapRegionGen`: перш за все оновлюються вертикальні карти дисторсій (vertical distortion ore maps) для регіону <sup>11</sup>, потім кожен поклад «малює» свій шар руди в колонках згідно з використанням алгоритмом (напр., диск/горизонтальний прошарок тощо). Оскільки конфіг файли детально описують форму та розподіл покладу, результат формується з урахуванням шумів поверхні і орієнтації.
- **Шуми і мапи шуму:** На ранньому етапі генеруються численні шумові карти: клімату, квітів, кущів, лісу, пляжу, геологічних провінцій і ландформ (загалом 7 типів шуму) <sup>12</sup>. Для цих шумів VS використовує три основні алгоритми: **звичайний октавний (weighted octave) шум**, **класичний Перлін** і **нормалізований шімплекс** <sup>12</sup>. Кожна карта має власні параметри октав, амплітуд і частот (зазвичай задані у JSON-файлах конфігурації ландформ чи провінцій). Карти шуму впливають на висоту ґрунту, вологість, вершинні види рельєфу тощо. Ландформи (Landforms) самі визначають форму місцевості – вони теж описані в асетах (JSON) з параметрами октав, порогів (thresholds) і ключових висот (Terrain Y key) <sup>13</sup>.
- **Ландформи і варіанти:** Система ландформ визначає загальний рельєф (гори, долини, рівнини тощо) на основі шумів і клімату. Кожна ландформа у конфігах задає набір параметрів (октав, порогів, позицій Y), які змінюють інтенсивність шуму по висоті <sup>13</sup>. Наприклад, для гірських ландформ піднімаються ключові висоти так, щоб над певним значенням формувалися гори. Варіанти ландформ дають змогу міксувати шуми – технічно це просто інші записи у тому ж JSON-файлі.
- **Верхні шари ґрунту:** Після формування рельєфу надгенеровані шари ґрунту (soil), трави (grass) та снігу. Цей етап виконується у пізніх пасах (Vegetation/NeighbourSunLightFlood). Функціонал відповідає генераторам блокових шарів (`GenBlockLayers`): накладається покривний шар ґрунту, зверху – шар трави/моху, над високими горами та у холодних біомах – шар снігу. У документації згадується, що саме на етапі «Block layers» створюються саме ці шари <sup>14</sup>.
- **IWorldGenBlockAccessor та BeginColumn():** Для безпечної роботи з блоками у процесі генерації використовується спеціальний об'єкт `IWorldGenBlockAccessor`, який дає можливість змінювати блоки у колонках, навіть якщо сусідні чанк не згенеровані. Особливість: цей аксесор кешує останню колонку. Через це при повторному виклику генерації того ж стовпчика потрібно очистити кеш викликом `BeginColumn()` <sup>15</sup>. Тобто після початку обробки кожної нової колонки перед першим записом слід викликати `BeginColumn()`, аби уникнути попадання в кеш старого стовпчика <sup>15</sup>.

- **Генерація печер:** Типовий процес вирізання печер здійснюється на пасі Terrain Noise за допомогою алгоритмів карвінгу (можливо, схожих на Minecraft). За замовчуванням система використовує лінійний конгруентний генератор випадкових чисел (LCG) для конфігурації печер. (Мод CaveTweaks зазначає, що базовий LCG може бути замінений на Mersenne Twister для покращення різноманітності) <sup>16</sup>. Поклади печерних просторів формують «тунелі» під землею, часто з бічними відгалуженнями і вертикальними шахтами.
- **Баги генерації:** Серед відомих проблем генерації – **кусочність (chunky terrain)** при різкому зіткненні старих і нових чанків, **синусоїдальні тераси** (плавні хвилі рельєфу, спричинені шумом з низькою частотою, характерно для TFC); а також **зміщення шарів** у GenRockStrata (як описано вище про гібридні шари) <sup>6</sup>. Окрім цього, баг double-granite (подвійний гранит) є наслідком помилки в GenRockStrataNew <sup>6</sup>.

## TFC 1.7.10 (старий TerraFirmaCraft)

TerraFirmaCraft 1.7.10 мав власний рушій генерації (клас TFCChunkProviderGenerate), що працював за ванільною 16×16×256 системою. Тут висота світу = 256 блоків, рівень моря приблизно у≈142 (значно вище, ніж у Minecraft) <sup>17</sup>. Світ поділяється на геологічні провінції (Shield, Plateau, Basin тощо) з фіксованими товщинами осадових/метаморфічних/ігнейних шарів. Генератор створює три основні шари порід (осадові над метаморфічними над ігнейними), зокрема враховує метаморфічні інверсії і базальтові шари зверху. Руди розподіляються горизонтально в дискових покладах (смітські алгоритми) і змішуються шарами. Відомі особливості: «синусоїдальне терасування» рельєфу, дуплікація граниту (як і у VS), а також проблеми вирівнювання шарів при зміні версії світу.

## TFC 1.20+ (нова, JSON-датапак worldgen)

У версіях TFC 1.18+ (і 1.20+) увесь генератор світу перенесено у JSON-датапаки (аналогічно до сучасних механік Minecraft) <sup>18</sup>. При створенні світу обирається спеціальний level-type чи preset TFC, який використовує власний генератор чанків і біом. Розмір чанків залишається 16×16 (висота – 384 в MC 1.20, збережено властивості Java 1.18+), а деталі (геологічні шари, провінції, поклади, структури) описуються JSON-файлами у каталозі data/tfc/worldgen. Наприклад, конфігурація rock\_layer\_settings задає товщини і порядок шарів, а файли у deposits/ – параметри рудних родовищ. Система підтримує повністю модифіковані шляхи генерування (бекенди Nether/Minecraft), оскільки рушій співпрацює з новою архітектурою ванілі.

## TerraFirma2 (hex+polygonal)

Про TerraFirma2 (гри, що описують світ у гексових/полігональних вокселях) відомо небагато. На відміну від TFC, вона використовує гексагональні вокселі замість кубічних. Генерація світу за аналогією опирається на параметри геології та клімату, але структура даних іншого типу (ймовірно, підхід «voxel mesh»). Світ може бути більш плавним завдяки полігональним поверхням. Інформації з офіційних джерел (геймдев) мало: назви конкретних класів/алгоритмів невідомі. Загалом використовується система розбиття території на «провінції» з визначенням товщини шарів порід (осадові, метаморфічні, ігнейні) подібно до TFC, але сітка – не квадратна, а гексагональна. Підтримуються круті схили та еволюційні процеси (ерозія, ріст рельєфу тощо) завдяки внутрішнім симуляціям. (Оскільки це окрема гра, технічні деталі не документовані у відкритих джерелах.)

【\*\*】

【embed】 На наведеному скріншоті показано мікровоксельний рельєф з дуже дрібною деталізацією (0.25–0.5 м). Для такого світу ділять звичайний метричний блок на 2–4 маленьких субблоки. **Структура:** зберігають чанк подібно до кубічної системи, але з підрозділенням вдвічі/вчетверо по кожній осі. Наприклад, якщо звичайний чанк 32×32×32 блоків по 1 м, то мікровоксельний може бути 64×64×64 по 0.5 м. **Шуми:** використовується комбінація великих і малих масштабів шуму. Спершу задають грубий контур рельєфу (низькочастотний фрактальний шум), потім додають високочастотні шуми для дрібних деталей (щільніші зміни висоти). Наприклад, використовують перлин-шум або симплекс з високою частотою (багато октав) для текстуровання поверхні. **Алгоритм побудови сітки (mesh):** оскільки мікровокселі дозволяють доволі плавний рельєф, можна застосувати алгоритми генерування полігональної поверхні (Marching Cubes, Dual Contouring або Surface Nets). Вони перетворюють масив мікровокселів на полігональну сітку, зберігаючи круті схили та гострі кути. Surface Nets, наприклад, створює вершини в центрі кожного кубічного елемента і плавно зв'язує їх, тоді як Dual Contouring відомий збереженням чітких контурів при апроксимації висот. У випадку, коли потрібно просте кубічне відображення (як на скріншоті), можна просто розташовувати маленькі куби без додаткового сглажування. **Підтримка крутих схилів:** дрібніший розмір блоків сам собою дозволяє набагато крутіші підйоми (майже вертикальні), адже перепад 1–2 блоків мікровокселя не такі величезні метричні висоти. Якщо ж використовувати плавні алгоритми (Marching Cubes тощо), вертикальні стіни можуть бути представлені як вузькі різко оконтурені полігони. Важливо забезпечити високу щільність mesh (великий бюджет полігонів) для збереження рельєфу на малій висоті.

**Коротко:** у мікровоксельному світі з 0.25–0.5 м блоками зберігають розбиття на чанки (збільшену деталізацію) і генерують висотний шум високої частоти. Потім будують полігональну поверхню або просто ставлять віскілетові куби. Такі методи дозволяють досягати високої деталізації та крутих форм, як на цьому прикладі.

---

1 2 15 Modding:WorldGen Concept - Vintage Story Wiki

[https://wiki.vintagestory.at/Modding:WorldGen\\_Concept](https://wiki.vintagestory.at/Modding:WorldGen_Concept)

3 4 10 11 12 13 14 Terrain Generation - Vintage Story Wiki

[https://wiki.vintagestory.at/Special:MyLanguage/Terrain\\_Generation](https://wiki.vintagestory.at/Special:MyLanguage/Terrain_Generation)

5 GitHub · Where software is built

<https://github.com/anegostudios/VintageStory-Issues/issues/3790>

6 7 GenRockStrataNew.genBlockColumn() creates granite strata that are twice as thick as they should be · Issue #3856 · anegostudios/VintageStory-Issues · GitHub

<https://github.com/anegostudios/VintageStory-Issues/issues/3856>

8 9 Rock - Vintage Story Wiki

<https://wiki.vintagestory.at/Rock>

16 CaveTweaks - Vintage Story Mod DB

<https://mods.vintagestory.at/cavetweaks>

17 Server RAM - Support - TerraFirmaCraft Forums

<https://terrafirmacraft.com/f/topic/5076-server-ram/>

18 World Generation

<https://terrafirmacraft.github.io/Documentation/1.18.x/worldgen/>