

# Звіт про проєкт *TerrainGeneration*

Вікторія Кіберєва, Станіслав Коноваленко, Дарина Ничипорук, Анастасія Любенч

9 травня 2025 р.

## Анотація

Цей звіт описує розробку та реалізацію проєкту *TerrainGeneration* — додатку, натхненного Minecraft, для процедурної генерації ландшафту з використанням теорії клітинних автоматів на Python. Він призначений для тих, хто вперше знайомиться з проєктом, і містить детальний опис архітектури, алгоритмів, функціоналу та інструкції з використання.

## 1 Вступ

Процедурна генерація ландшафтів стала популярною в ігровій індустрії завдяки можливості створювати великі унікальні світи без ручного дизайну кожного елементу. Наш проєкт *TerrainGeneration* демонструє підхід до генерації таких світів із застосуванням клітинних автоматів та дискретної математики. Користувачі можуть задавати початкові параметри, отримуючи кожного разу свіжий та реалістичний результат.

## 2 Основні можливості

- **Процедурна генерація** ландшафту на основі клітинних автоматів.
- **Різноманітні біоми**: створення окремих екосистем із відмінними властивостями.
- **Сид-методика**: відтворюваність світу за фіксованим seed.
- **Налаштовувані параметри**: щільність заповнення, кількість ітерацій, розподіл текстур тощо.

## 3 Встановлення

1. Клонування репозиторію:

```
git clone https://github.com/vi-kibereva/TerrainGeneration.git
```

2. Перехід до каталогу:

```
cd TerrainGeneration
```

3. Встановлення залежностей:

```
pip install -r pyproject.toml
```

4. Запуск програми:

```
python3 -m src.main
```

## 4 Огляд архітектури

Проект складається з трьох основних модулів:

1. `grid.py`: керує мережею чанків розміром  $16 \times 16$  клітин та координує їх генерацію навколо вказаної позиції.
2. `chunk.py`: описує клас `Chunk`, який містить дані клітин і підтримує двоетапну генерацію: попередню та фінальну.
3. `evolution.py`: імплементує алгоритми еволюції ландшафту, формування біомів та накладання текстур з використанням згорток.

### 4.1 Модуль Grid

- **Ініціалізація**: параметр `density` задає вірогідність заповнення клітин на початку.
- `generate_around(pos, radius)`: створює чанки в трьох фазах:

1. `create_random_chunks : iiiradius+6`.

### 4.2 Модуль Chunk

Кожен `Chunk` має стан: `NOT_GENERATED`, `PRE_GENERATED`, `GENERATED` або `VOID`. Процес генерації:

1. `pre_generate_self`: згладжування даних з урахуванням сусідів.
2. `generate_self`: розширена зона  $20 \times 20$ , 100 ітерацій біомної еволюції з  $5 \times 5$  ядром та накладання текстур.

### 4.3 Алгоритми еволюції (`evolution.py`)

- `evolve`: згортка  $3 \times 3$  з таблицею переходів `CELL_LUT` для початкового згладжування.
- `generate_chunk_biome`: додавання двох бітів випадкового шуму та 100 проходів біомної еволюції з ядрами `BIOME_KERNEL`.
- `textures`: створення маски та рандомне накладання текстурних маркерів.

## 5 Висновки

Проект *TerrainGeneration* демонструє ефективність застосування клітинних автоматів для процедурної генерації ландшафту з різними біомами та текстурами. Завдяки відкритій архітектурі та налаштовуваним параметрам, користувачі можуть досліджувати безмежні варіанти світів та адаптувати систему під свої потреби.

## 6 Ролі розробників

- Вікторія Кіберєва (@vi-kibereva) — основна логіка та бекенд.
- Станіслав Коноваленко (@weqpro) — оптимізація алгоритмів, бекенд.
- Дарина Ничипорук (@dd8ria) — розробка інтерфейсу користувача.
- Анастасія Любенчук (@saudeawd) — візуалізація та UX.