AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do generowania map

z przedmiotu

Języki programowania obiektowego

EIT rok 3

Karol Klapa

Środa 8:00

prowadzący: mgr. Inż. Jakub Zimnol

13.01.2026

1. Opis i cel projektu

Mój projekt został wykonany w celu stworzenia biblioteki do generowania map, którą można potem wykorzystać na przykład jako mapa do gry komputerowej. Mapa generuje się w określonych wymiarach i z określonym seedem, dlatego możemy wygenerować tą samą mape wielokrotnie, jeżeli tylko będziemy mieli odpowiedni seed.

2. Opis klas

W moim projekcie wykorzystałem podejście obiektowe, wykorzystuje dziedziczenie I polimorfizm oraz klasy abstrakcyjne.

Głównym elementem obiektowym jest jednak polimorfizm oparty na smart pointerach, dzięki któremu klasa Map przechowuje kolekcje tile'ów czyli wskaźników na Biome, co pozwala na łatwą rozbudowe programu o kolejne biomy, ponieważ dodajemy tylko klasę kolejnego biomu I sposób w jaki może się ona pojawiać w Mapgenerator.

Biome – klasa bazowa po której dziedziczą wszystkie konkretne biomy, odpowiada ona w głównym stopniu za wygląd biomu. Przetrzymuje ona także liczbę całkowitych dostępnych biomów która równa jest wartości podstawowo występujących biomów.

Tile – reprezentuje jeden kafelek mapy, przechowuje wskaźnik na biome oraz liczbę służącą do dodawania detali do mapy

PerlinNoiseGenerator – klasa obsługująca generacje mapy przy użyciu szumu Perlina, wykorzystuje wartość szumu I przypisuje do kafelka odpowiedni biom bazując na podanych warunkach. Pozwala uzyskać mapę z gładkimi przejściami idealna do generowania wysp.

VoronoiGenerator – klasa służąca również do generacji mapy oparta na diagramach Woronoja, losuje pare startowych punktów a następnie przypisuje kafelkom leżącym najbliżej taki sam typ biomu.

MapDetails – klasa odpowiadająca za dodawanie do mapy detali, takich jak głębiny, ciemne lasy, polany czy plaże oraz dodatkowe elementy takie jak krzaki czy drzewa.

3. Kompilacja I uruchomienie

W celu kompilacji został wykorzystany CMake

Przed uruchomieniem kompilacji należy upewnić się że w folderze libs znajduje się biblioteka FastNoiseLite, jeżeli nie należy ją pobrać z tego miejsca:

https://github.com/Auburn/FastNoiseLite/tree/f510216263f0bc6231d8c5b8b0c9db8d62ff38d9

Potem należy wykonać następujące komendy w głównym katalogu projektu

mkdir build

cd build

cmake ..

make

Po udanej kompilacji powstanie plik wykonywalny ./generatedmap

Po uruchomieniu programu będziemy mogli wybrać jeden z dostępnych dwóch generatorów.

W pliku testowym możliwa jest opcja odkomentowania linii 20 – 32 oraz zakomentowania linijek 16 – 19,

pozwoli to na wprowadzenie własnych wymiarów mapy oraz seeda.