

Projekt „Generator map kafelkowych”

Bartłomiej Piktel

1. Format danych:

Mapa kafelkowa wraz z danymi dotyczącymi kafelków przechowywane są w dedykowanych obiektach. W module ‘tile’ znajdują się definicje klasy Tile i TileTreeNode. Pierwsza z nich przechowuje informacje dotyczące pojedynczego kafelka, a druga opakuje je i pozwala na utworzenie wielostopniowej hierarchii kafelków rodziców i dzieci. Każdy kafelek składa się z unikalnego id, nazwy, koloru, oraz parametrów wypełnienia (określa jaka część kafelka wyżej w hierarchii zostanie wypełniona przez ten rodzaj kafelka) i ilości wysp danego rodzaju. Każdy element drzewa kafelków składa się z pojedynczego obiektu kafelka i dowolnej ilości dzieci – kolejnych elementów drzewa.

Mapa jest przechowywana jako obiekt klasy TileMap z modułu ‘tile_map’. Mapa składa się z dwuwymiarowej tablicy z modułu numpy oraz drzewa kafelków.

Wszystkie metody podziału generacji na zadania oraz samej generacji poszczególnych rodzajów kafelków zawarte są w module ‘generator’, gdzie klasa TileMapGenerator odpowiada za wyodrębnienie kolejnych kroków generacji, a BorderGeneration zawiera metody generowania pojedynczego rodzaju kafelka. Taki podział pozwala na łatwą zmianę metody generacji, ponieważ obiekty klasy TileMap są dzielone na zadania polegające na wygenerowaniu pojedynczego rodzaju id na prostej dwuwymiarowej tablicy id, zatem klasa generatora operuje tylko na wartościach id i nie musi operować na całym obiekcie mapy.

Dodatkowymi modułami są: ‘tile_map_io’ zawierające metody wczytywania i zapisywania obiektu mapy; ‘visualisation’ udostępniająca metody zamiany obiektu mapy na tekst do wypisania w terminalu lub obraz do wyświetlenia.

W celu zapewnienia wygody użytkownika klasa moduł ‘gui’ zawiera interfejs graficzny pozwalający na definiowanie, generowanie, wczytywanie i zapisywanie oraz wyświetlanie map.

2. Użyte algorytmy:

Generowanie mapy kafelkowej rozpoczyna się od stworzenia mapy o wybranym rozmiarze i wypełnienia jej przez id przypisane kafelkowi znajdującemu się najwyżej w hierarchii (tzw. root tile). Następnie wydzielone zostają zadania polegające na wygenerowaniu pojedynczego rodzaju kafelka na jego bezpośrednim rodzicu w hierarchii. Algorytm generacji polega na losowym

wyznaczeniu początkowego kafelka i generowaniu wokół niego wyspy. W pierwszym kroku wybierany jest jeden z kafelków brzegowych (przynajmniej jeden z kafelków sąsiadujących bokami zawiera id rodzica). Następnie wybierany jest jeden z kafelków sąsiadujących z nim (zawierający id rodzica) i zamieniany na id generowanego kafelka. Algorytm powtarzany jest do osiągnięcia zamierzonego rozmiaru wyspy lub gdy zabraknie miejsca do generacji (z powodu zbyt małej liczby kafelków rodzica, zbyt małego rozmiaru wyspy kafelków rodzica, na której generowana jest wyspa kafelka dziecka lub w celu zachowania ilości wysp – wyspy tego samego rodzaju kafelka nie mogą się łączyć).

Dodatkowo przed rozpoczęciem generowania wysp, ich rozmiar jest dobierany w następujący sposób: parametr wypełnienia danego kafelka dzielony jest na tyle części ile wysp ma zostać wygenerowanych w taki sposób, żeby największa wyspa była maksymalnie 5 razy większa niż najmniejsza.

3. Podsumowanie zakresu prac:

Generator map kafelkowych jest projektem kompletnym pod względem dostępności zaplanowanej funkcjonalności. Posiada on możliwość generowania map o określonym rozmiarze ze zdefiniowanych rodzajów kafelków, zamienia je na obrazy możliwe do zapisania oraz pozwala na zapisywanie i wczytywanie wcześniej stworzonych map. Największym atutem jest możliwość definiowania rodzajów i poziomów kafelków: każdy z nich może być skonfigurowany pod względem ilości zajmowanego miejsca, podziału na ilość wysp oraz dowolnej ilości kafelków – dzieci.

Dostęp do funkcji jest dosyć łatwy dzięki interfejsowi graficznego, który stanowił dosyć duże wyzwanie w zaprojektowaniu. Największym problemem było stworzenie widoku hierarchii kafelków łatwego w odczytywaniu i modyfikowaniu. Możliwe są dodatkowe usprawnienia działania interfejsu (np. podświetlanie błędnych danych kafelka w czasie edycji), jednak w celu uproszczenia programu prawdopodobnie wskazane byłoby użycie modułu innego niż tkinter.

Oprócz tego za problem może uznany być dosyć okrągły kształt wysp przy dużych rozmiarach map. Możliwością rozwiązania tego problemu byłoby zmodyfikowanie kroku wyboru jednego z sąsiadów kafelka o system wag pozwalający kontrolować kierunek wzrostu wyspy. Możliwe jest również dosyć proste dodanie nowego algorytmu sortującego, który musi obsługiwać jedynie generowanie id jednego rodzaju na id drugiego rodzaju w tabeli 2D.

Podsumowując, generator map kafelkowych to projekt kompletny, ale z łatwymi drogami rozwoju. Jest również przyjazny dla użytkownika, ponieważ nie wymaga używania poleceń tekstowych, posiada wygodny interfejs graficzny.