

Dokumentacja projektu

Julia Majkowska

30 stycznia 2017

1 Krótki opis

Program umożliwiający zamianę mapy wysokości na kolorową mapę hipsometryczną.
Będzie umożliwiał:

1. wybór w nowym okienku pliku źródłowego tablicy wysokości (obsługiwane będą formaty DTED (Level 0, 1, 2), SRTM (pliki binarne HGT, TIFF, PNG))
2. stworzenie własnego zestawu poziomicy i odpowiadającej mu palety, przez dodawanie i usuwanie poziomicy i przypisywanie im wartości w metrach oraz koloru
3. zapis poziomicy i palety do istniejącego lub nowego pliku, wybranego lub wpisanego w odpowiednim okienku
4. wczytanie zestawu poziomicy i palety, zastępując stworzone w oknie poziomicy, z pliku wybranego w nowym okienku
5. wczytanie poziomicy dodając je do palety w okienku, z pliku wybranego w nowym okienku
6. skorzystanie z domyślnej palety kolorów
7. podgląd mapy dla aktualnej palety i poziomicy w nowym oknie po kliknięciu odpowiedniego przycisku
8. zapis nowej mapy do pliku png

2 Wymagane biblioteki

1. Gtk 3.0
2. GDK-PixBuf 2.0
3. Gdal 1.9

3 Kompilacja

1. Linux

Zainstalować biblioteki można następującymi komendami

```
sudo apt-get install gdal-bin
sudo apt-get install libgtk-3-dev
sudo apt-get install build-essential libgdal-dev
sudo apt-get install libgtk-3-dev
```

Aby skompilować można dodać do folderu z plikami programu plik Makefile.

Następnie wejść z konsoli do folderu zawierającego pliki programu i wpisać komendę 'make'.

Przykładowy Makefile

```
compile = cc 'pkg-config --cflags gtk+-3.0 gdal '
```

```
all: interfejs.o paleta.o obrazek.o
```

```
$(compile) interfejs.o paleta.o obrazek.o -o caly_program 'pkg-config --libs g
```

```
interfejs.o: interfejs.c paetalib.h mapa.h
```

```
$(compile) interfejs.c -c -o interfejs.o
```

```

paleta.o: paleta.c paletalib.h
$(compile) paleta.c -c -o paleta.o

obrazek.o: obrazek.c paletalib.h mapa.h
$(compile) obrazek.c -c -o obrazek.o

```

2. Windows

Można stworzyć projekt w CodeBlocksie dopisując w opcjach kompilatora 'pkg-config -cflags gtk+-3.0 gdal', oraz 'pkg-config -libs gtk+-3.0 gdal' w opcjach linkera

4 Wymagania systemowe

- Program powinien działać zarówno na Windowsie i Linuxie.
- Testowano na Ubuntu 15.10.
- Nie wymaga dostępu do internetu.
- Wymagana ilość pamięci RAM zależy od rozmiaru wczytywanych plików. W większości przypadków wystarczy 500 MB

5 Obsługiwane formaty map wysokości

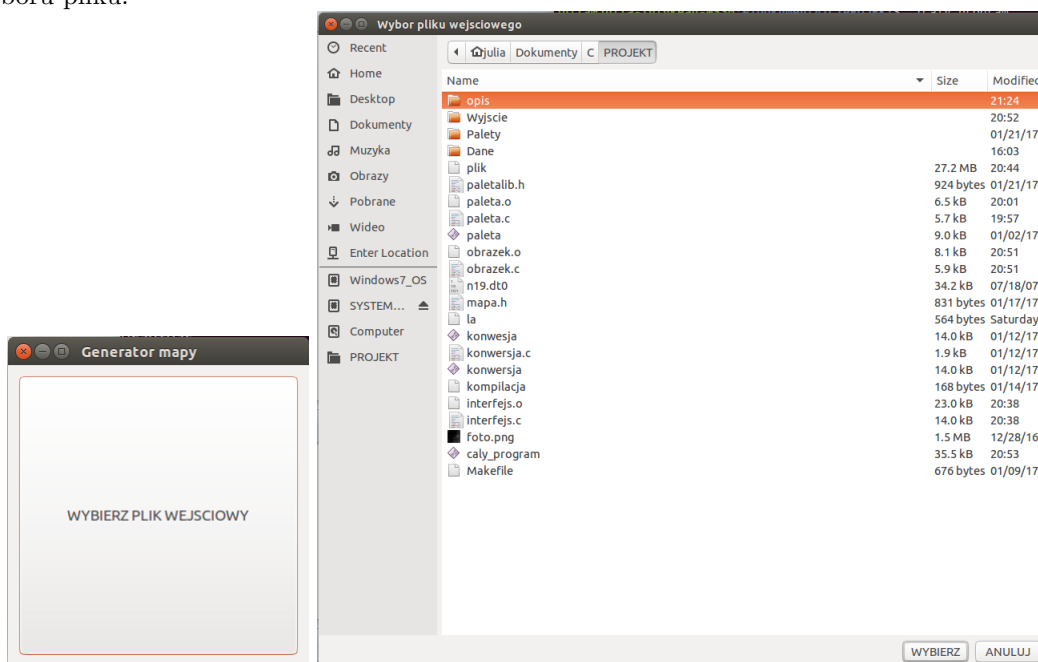
1. DTED (Level 0, 1, 2)
2. SRTM (PNG, HGT, TIFF)

6 Opis formatu zewnętrznej palety

Plik taki powinien być listą poziomich składających się z całkowitoliczbowej wysokości oraz 4 liczb rzeczywistych z przedziału [0,1] oznaczających kolejno składowe czerwoną, zieloną i niebieską oraz przezroczystość koloru. Niezależnie od wpisanej przezroczystości poziomicy na mapie wszystkie kolory będą nieprzezroczyste.

7 Interfejs

Korzystanie z programu rozpoczynamy od wybrania pliku wejściowego z mapą, wyskoczy dodatkowe okienko wyboru pliku.



Po wybraniu pliku pojawią się dodatkowe elementy:

1. Przycisk WCZYTAJ PALETĘ

Przycisk umożliwia wybranie w osobnym okienku pliku z paletą, którą następnie będzie można zaaplikować do mapy wysokości, lub zmodyfikować.

2. Przycisk ZAPISZ PALETE

Przycisk umożliwia zapisanie stanu aktualnie modyfikowanej palety do nowego lub istniejącego pliku. Plik docelowy wybiera się w osobnym okienku.

3. Przycisk ZASTOSUJ PALETE DOMYŚLNĄ

Po naciśnięciu przycisku generuje się domyślną paletę, z kolorami typowymi dla mapy hipsometrycznej. Poziomice są dostosowane do pliku z mapą w taki sposób, aby uwzględnić zakres wysokości.

4. Przycisk ODŚWIEŻ PODGLĄD

Przycisk aktualizuje podgląd o wprowadzone zmiany w paletcie. Przy plikach o rozmiarze powyżej 15 MB podgląd może się nie ładować natychmiastowo

5. Przycisk ZAPISZ MAPE

Przycisk umożliwia zapisanie stanu aktualnie modyfikowanej mapy do nowego lub istniejącego pliku. Plik docelowy wybiera się w osobnym okienku. Mapa zapisuje się w formacie PNG.

6. Przycisk NOWA POZIOMICA

Po naciśnięciu przycisku pojawi się nowa poziomica do zmodyfikowania.

7. Przycisk WYCZYŚĆ POZIOMICE

Przycisk usuwa wszystkie (!) poziomice z aktualnie modyfikowanej palety.

8. Pole podglądu

Pojawia się również czarne pole z podglądem oraz, po jego prawej stronie, przyciski do powiększania i pomniejszania podglądu. Jeżeli wielkość podglądu przekroczy rozmiary pola można poruszać się po podglądzie przy pomocy suwaków z prawej strony i od dołu pola podglądu.

9. Poziomice

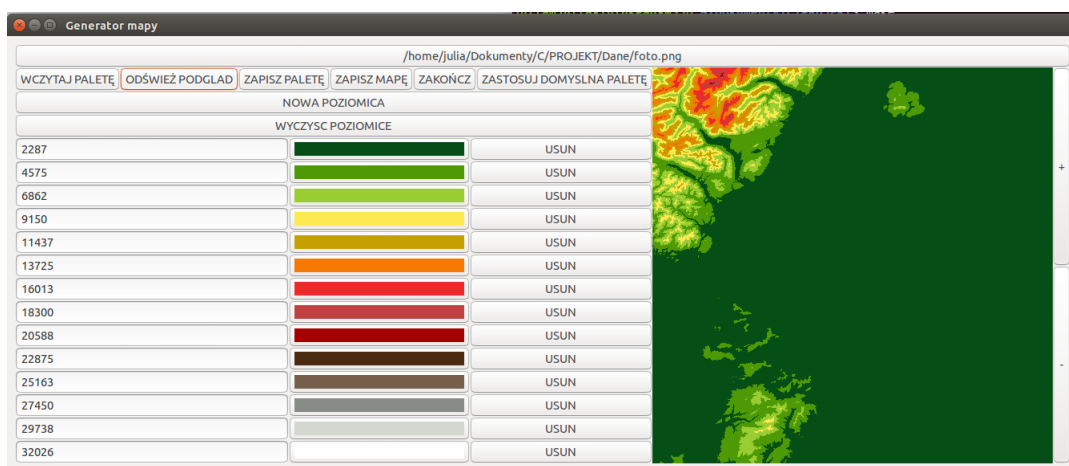
Pojawiają się po wczytaniu palety, lub po dodaniu poziomicy. Na każdej znajdują się :

- (a) Pole tekstowe do wpisania wartości liczbowej poziomicy w metrach. Wartość poziomicy się zmienia po wpisaniu - nie trzeba wciskać dodatkowo Enter. Do momentu wpisania wartości liczbowej poziomica nie jest aktywna. Wpisanie czegoś innego niż liczba powoduje przypisanie poziomicy wartości domyślnej 0. Wartość poziomicy stanowi górne ograniczenie wartości punktów pokolorowany na dany kolor.

Uwaga! W niektórych plikach wartości 1 w pliku mapy wysokości jest przypisana jest inna wartość niż 1 metr. Wtedy ciężar przeliczenia wartości poziomicy spoczywa na użytkowniku.

- (b) Przycisk zaznaczony kolorem (domyślnie seledynowym) służący do ustawiania poziomicy koloru . Po naciśnięciu go wyskakuje okienku wyboru barwy, w którym można wybrać z pośród istniejących kolorów lub mieszać własny. Po wybraniu koloru przycisk zmienia kolor na wybrany.

- (c) Przycisk usuń trwale usuwa poziomice z palety



Wszelkie komunikaty o błędach wypisywane są na standardowe wyjście

8 Plik paleta.c

Plik korzysta z funkcji bibliotek zewnętrznych GTK oraz stdio oraz stdlib oraz funkcji ustaw_poziomice z modułu interfejs

Biblioteka opiera się na dwóch strukturach:

1. poziom

```
struct poziom{
    int  wysokosc;
    GdkRGBA kolor;
    struct poziom* nastepny;
    struct poziom* poprzedni;
};
```

Struktura reprezentuje element listy poziomic i zawiera całkowitoliczbową wysokość, kolor w postaci struktury GdkRGBA oraz wskaźniki na poprzedni oraz następny element listy.

Korzystam z następujących funkcji do tej struktury:

- (a) poziom* nowy_poziom(paleta *pal, int w, GdkRGBA kol) — funkcja alokująca pamięć pod strukturę poziom i zwracająca wskaźnik na nią. Ustawia wartość wysokości na w i wartość kolor na kol oraz wstawia ją w odpowiednie miejsce w paletce pal (kolejność w strukturze jest zgodna z wysokościami).
- (b) void usun_poziom(poziom *p, paleta *pal) — funkcja zwalniania pamięć struktury poziom oraz usuwająca go ze struktury pal.
- (c) poziom* przenies(poziom* nowy, paleta* pal); - funkcja aktualizująca położenie danego poziomu w paletce zgodnie z wpisaną wysokością.

2. paleta

```
typedef struct {
    poziom* p;
    poziom* k;
} paleta;
```

Struktura reprezentuje listę poziomic i trzyma jej pierwszy i ostatni element.

Korzystam z następujących funkcji do tej struktury:

- (a) paleta* nowa__paleta() — funkcja alokująca pamięć pod strukturę paleta i zwracająca wskaźnik na nią.
- (b) void usun__palette(paleta *pal) — funkcja zwalniania pamięć struktury paleta.
- (c) void wyczyszc__palette(paleta *pal) — funkcja niszcząca wszystkie poziomicę w strukturze paleta.
- (d) paleta* wczytaj__palette(char* nazwa__pliku, GtkWidget* pudelko, paleta* p) — funkcja wczytuje paletę z pliku o ścieżce zapisanej w nazwa__pliku do palety p (jeśli p to NULL to tworzy nową paletę) oraz dodaje reprezentację poziomic do kontenera gtk pudelko.
- (e) void zapisz__palette(paleta* p, char* nazwa__pliku) — funkcja zapisująca paletę p do pliku o ścieżce zapisanej w nazwa__pliku.
- (f) paleta* paleta__domyslna(int min, int max, GtkWidget *pudelko, paleta* pal) — funkcja tworząca domyślną paletę kolorów w paletce pal dla danej wartości minimalnej i maksymalnej występującej na mapie. Funkcja tworzy również reprezentację graficzną palety w kontenerze GTK pudelko.

Dodatkowo została zaimplementowana funkcja pomocnicza void zniszcz(GtkWidget* widget, gpointer dane) służąca do niszczenia widgetu podanego jako pierwszy argument

9 Plik obrazek.c

W tym pliku znajdują się funkcje związane z operacjami na mapie wysokości.

Struktura mapa

```
typedef struct{
    int h, w, maxval, minval;
    int ** obraz;
}mapa;
```

Funkcje

1. `mapa* stworz__nowa__mape(int h, int w)`
Funkcja alokuje pamięć na strukturę `mapa` o wysokości `h` i szerokości `h`.
2. `void zniszcz__mape(mapa *m)`
Funkcja zwalnia pamięć struktury pod wskaźnikiem `m`. Zwalnia też wszystkie zawarte w niej wskaźniki.
3. `mapa* wczytaj__mape__z__formatu(char* nazwa)`
kod zaczerpnięty z http://www.gdal.org/gdal_tutorial.html
Funkcja wczytuje plik o ścieżce zapisanej w `nazwa` przy pomocy biblioteki `gdal`. Wczytuje jedynie pierwszą warstwę barwną zgodnie ze standardową specyfikacją pilku mapy wysokości. Zapisuje dane w nowo utworzonej strukturze typu `mapa`. Dane są wczytywane jako 16-bitowy integer ze znakiem, dane wykraczające poza zakres zostaną zaminione na maksymalną wartość 16-bitowego inta. Korzystam z funkcji biblioteki GDAL aby obliczyć minimalną i maksymalną wysokość na mapie.
4. `void zapisz__mape__do__pliku(mapa* m, paleta *pal, const char* nazwa)`
Funkcja zapisuje mapę kolorową będącą mapą wysokości `m` z zaaplikowaną paletą `pal` do pliku o ścieżce zapisanej w `nazwa`.
5. `GdkPixbuf* zamien__na__kolorki(paleta* pal, mapa* m)`
Funkcja zamieniającą mapę wysokości na nową strukturę `GdkPixbuf` wraz z nadaniem pixelom kolorów zgodnie z paletą.
6. `void zmien__pixel(GdkPixbuf* imageptr, int x, int y, GdkRGBA* kol)`
kod pochodzi z tej strony <http://stackoverflow.com/questions/16785886/get-pixel-value-on-gdkpixbuf-set-pixel-value-with-gdkcairo>
Funkcja ustawiająca kolor pixela o współrzędnych `x, y` w `GdkPixbuf` na `kol`.
7. `GdkRGBA* znajdz__kolor(int wysokosc, paleta* pal)`
Funkcja znajdującą kolor odpowiadający wysokości dla danej palety.
8. `short get__green(GdkRGBA* kolor)`
Funkcja zamienia zmiennoprzecinkową składową zieloną na wartość 8-bitowego inta tak jak w strukturze `GdkPixbuf`.
9. `short get__red(GdkRGBA* kolor)`
Funkcja zamienia zmiennoprzecinkową składową czerwoną na wartość 8-bitowego inta tak jak w strukturze `GdkPixbuf`.
10. `short get__blue(GdkRGBA* kolor)`
Funkcja zamienia zmiennoprzecinkową składową niebieską na wartość 8-bitowego inta tak jak w strukturze `GdkPixbuf`.

10 Plik `interfejs.c`

Moduł główny programu. Implementuje interfejs oraz wszystkie funkcje programu.

1. Operacje na plikach
Do wczytywania i zapisu plików wykorzystany został widget `GtkFileChooserDialog`. Kod obsługi tego widgetu został zaczerpnięty ze strony <https://developer.gnome.org/gtk3/stable/GtkFileChooserDialog.html>.
2. Wybór koloru
Do wyboru koloru dla poziomicy wykorzystano widget `GtkColorChooserButton`.
3. Wprowadzanie wysokości
Do wprowadzania wysokości służy widget `GtkEntry`. Wprowadzony tam tekst zostanie przekształcony przez funkcję `atoi` na liczbę.
Dla poziomicy stworzonej w oknie początkowo ustawiony jest tekst startowy funkcją `gtk__set__placeholder__text`.
Dla wczytanych poziomicy tekst jest ustawiany przy pomocy `gtk__set__text` na wartość poziomicy.
Wartość poziomicy jest aktualizowana po każdej zmianie w tym polu.
4. Podgląd
Podgląd jest tworzony poprzez konwersję do widgetu `GdkPixbuf` a następnie wyświetlany jako `GtkImage`.
Do poruszania się po podglądzie został użyty widget `GtkScrolledWindow`.

Powiększenie (analogicznie pomniejszenie) podglądu odbywa się przez pomnożenie przez 2 globalnej zmiennej typu `double enlargement`, a następnie przeskalowaniu widgetu `GdkPixbuf`, zawierającego obraz w początkowej rozdzielczości, o tą zmienną. Wykorzystana została funkcja `gdk__pixbuf__scale()` z opcją `GDK__INTERP__BILINEAR`.

Po odświeżeniu obraz się skaluje do szerokości 500 pixeli i proporcjonalnej wysokości.

Podgląd jest ładowany 5 sekund po ostatniej modyfikacji w palecie, lub od razu po zmianie pliku wejściowego, wczytaniu nowej palety lub zastosowaniu palety domyślnej. Przy plikach o rozmiarze powyżej 15mb podgląd może się nie ładować natychmiastowo.

5. Kolejność poziomicy

5 sekund po wprowadzeniu ostatniej zmiany w palecie poziomice sortują się automatycznie rosnąco względem wysokości.

11 Źródła map

Mapy zachodniego wybrzeża Stanów Zjednoczonych w formacie PNG

Wyszukiwarka map w formacie HGT

Wyszukiwarka map w formacie TIFF

Wyszukiwarka map w formacie DTED

12 Źródła używane przy pisaniu programu

Centrum programistyczne GNOME

Strona główna biblioteki GDAL