

Wstęp do programowania w C - projekt

Konrad Werbliński

11 listopada 2017

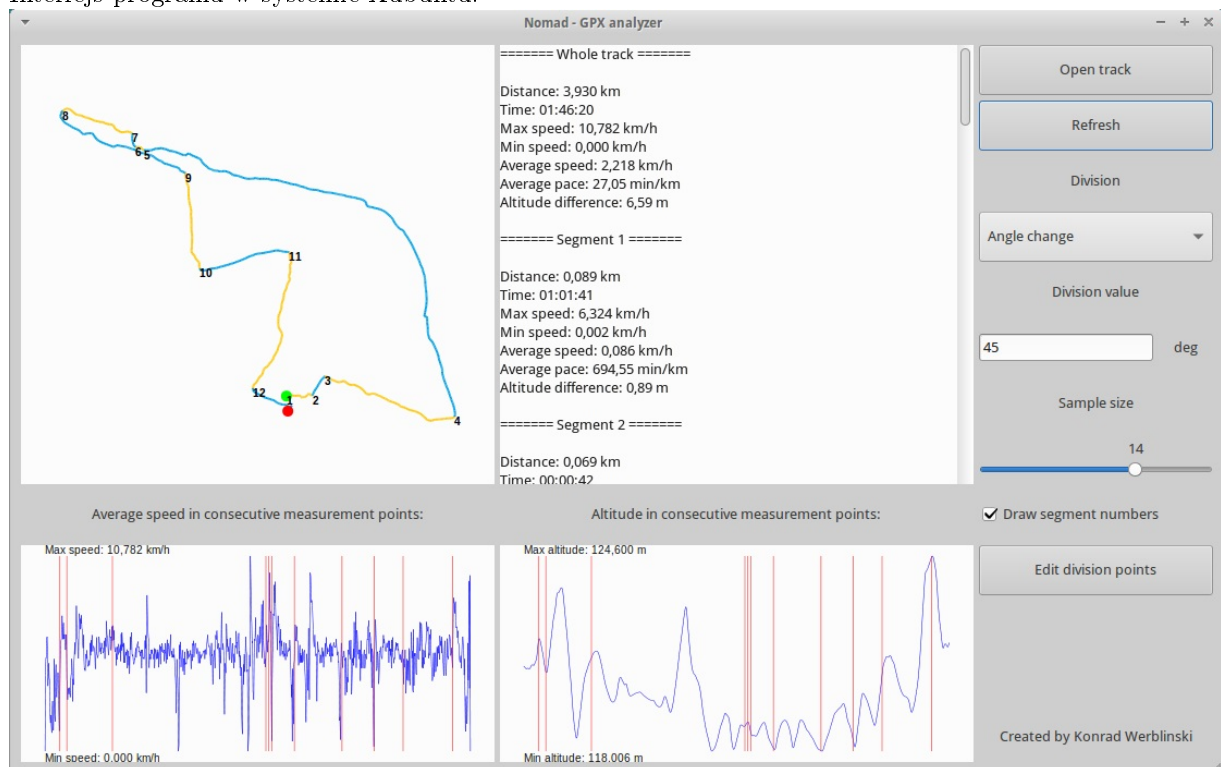
1 Ogólny opis

Stworzony przeze mnie, w ramach projektu, program o nazwie „Nomad - GPX analyzer” służy do analizy ścieżek zapisanych w plikach GPX. Potencjalnymi użytkownikami programu są osoby śledzące swoje treningi za pomocą GPS. Program oblicza parametry trasy (czas, prędkość, dystans, itp) oraz umożliwia podział trasy według kilku różnych kryteriów.

Interfejs programu w systemie Windows 10.



Interfejs programu w systemie Xubuntu.



2 Instrukcja obsługi

A. Parametry obliczane przez program:

1. Dystans
2. Czas
3. Maksymalna szybkość
4. Minimalna szybkość
5. Średnia szybkość
6. Średnie tempo
7. Różnica wysokości (wysokość maksymalna - wysokość minimalna)

Powyższe parametry są wyliczane dla całej trasy oraz dla poszczególnych segmentów.

B. Program umożliwia podział trasy według poniższych kryteriów:

1. Zmiana szybkości
2. Czas, np. podział trasy na segmenty odpowiadające kolejnym minutom
3. Dystans, np. podział trasy na segmenty odpowiadające kolejnym kilometrom
4. Zmiana tempa
5. Zakręt o kąt większy niż podany

Dla kryteriów zmiana szybkości, zmiana tempa oraz zakrętu program umożliwia wybór ilości punktów wykorzystywanych do wyliczenia wartości porównywanych w tym kryterium. Dla każdego powstałego w wyniku podziału segmentu zostają obliczone parametry wymienione w punkcie A.

C. Program rysuje dwa wykresy:

1. Szybkość w kolejnych punktach pomiaru

2. Wysokość nad poziomem morza w kolejnych punktach pomiaru

Na wykresach zaznaczone są punkty podziału. Znajdujący się po prawej stronie wykresów przycisk umożliwia edycje punktów podziału (dodawanie i usuwanie). Edycja polega na klikaniu w wybrane punkty na wykresie.

D. Program umożliwia włączenie lub wyłączenie numerowania segmentów na podglądzie trasy.

3 Implementacja

Kod programu jest podzielony na pięć modułów:

1. `main.c` - Moduł zawierający funkcję `main`.
2. `window.c` - Moduł z implementacją interfejsu programu. Znajdują się w nim funkcje wykorzystujące bibliotekę GTK min: tworzenie okna, obsługa zdarzeń, rysowanie wykresów oraz podglądu trasy, wybór pliku. Moduł ten stanowi serce programu, jest odpowiedzialny za wywoływanie odpowiednich funkcji z pozostałych modułów.
3. `GPX_parse.c` - Moduł z parserem plików GPX. Parser wyszukuje w pliku znaczniki `<trkseg>` a następnie parsuje wszystkie opisane w nich punkty. Parser omija komentarze oraz nieznane mu znaczniki. Jest zaprojektowany do radzenia sobie z wybrakowanymi plikami oraz plikami, które nie są GPX'ami. W takiej sytuacji parser usuwa punkty zwalniając przydzieloną pamięć oraz ustawia liczbę przeczytanych punktów na 0. Złożoność czasowa użytego algorytmu parsowania to $O(n)$. Za proces parsowania odpowiedzialna jest funkcja `GPX_read`. Zwraca ona wskaźnik na pierwszy element dynamicznie alokowanej tablicy struktur typu `GPX_entity`. Definicja typu `GPX_entity` znajdująca się w pliku nagłówkowym `GPX_parse.h`:

```
typedef struct
{
    double lat;
    double lon;
    double ele;
    long long time;
    bool lastInSegment;
} GPX_entity;
```

Trasa w pliku GPX może być podzielona na segmenty. Program wczytuje wszystkie segmenty, w ostatnich punktach ustawiając wartość pola `lastInSegment` na `true`. Program podczas liczenia dystansów uwzględnia tylko odległości pomiędzy punktami w tym samym segmencie, uznając że odległość przebyta pomiędzy końcem jednego segmentu, a początkiem drugiego nie zalicza się do treningu, ponieważ segmenty mogą być generowane np. wtedy gdy użytkownik zatrzyma i wznowi pomiar.

4. `analysis.c` - Moduł zawierający funkcje wyznaczające parametry oraz podział trasy. Znajdują się w nim 4 funkcje odpowiedzialne za podział: `divideByTime`, `divideBySpeedOrPace`, `divideByDistance`, `divideByAngle`.

Obliczanie tempa i szybkości jest analogiczne, dlatego podział na podstawie tych wartości jest zaimplementowany za pomocą jednej funkcji.

Funkcja `analize` jest odpowiedzialna za wyliczanie parametrów trasy. Tekst wygenerowanej analizy jest wyświetlany w polu tekstowym głównego okna programu. Aktualizuje ona pole tekstowe za pomocą funkcji `clearTextBox` oraz `appendTextBox` znajdujących się w module `window.c`.

Odległości pomiędzy punktami wykorzystywane do podziału i analizy wyznaczane są za pomocą wzoru:

$$scale = \cos\left(\frac{latitude1 + latitude2}{2}\right)$$
$$\Delta x = (longitude2 - longitude1) * scale * \frac{40075}{360}$$
$$\Delta y = (latitude2 - latitude1) * \frac{40030}{360}$$
$$d = \sqrt{\Delta x^2 + \Delta y^2}$$

5. `parse_double.c` - Moduł zawierający dwie funkcje: `parseDouble`, `isDouble`, które służą do wyliczania wartości liczby w napisie oraz sprawdzania czy dany napis jest liczbą. Powodem wykorzystania własnej funkcji `parseDouble` zamiast standardowego `atof` jest różnica pomiędzy różnymi kompilatorami języka C. Niektóre kompilatory traktują jako poprawny zapis liczby zmiennopozycyjnej z kropką, a inne z przecinkiem. Funkcja `parseDouble` obsługuje obie formy zapisu liczby zmiennopozycyjnej. Dzięki temu program jest niezależny od kompilatora oraz ułatwia użytkownikowi wprowadzenie danych, ponieważ nie musi się on zastawiać jaki zapis liczby będzie poprawny.

4 Kompilacja

Program można skompilować za pomocą środowiska Code Blocks, korzystając z załączonego pliku projektu. Kompilacja i działanie programu były testowane na komputerze z systemem Windows 10 oraz Xubuntu. Do kompilacji wymagana jest biblioteka gtk wersji 3.

Instalacja gtk (dla systemu windows i kompilacji w środowisku Code Blocks):

1. Pobrać gtk w wersji 3 i rozpakować pobrane archiwum do dowolnego folderu.
2. Dodać do zmiennej środowiskowej PATH ścieżkę do folderu z punktu 1.
3. Stworzyć zmienną środowiskową `PKG_CONFIG_PATH` oraz nadać jej wartość:
(ścieżka do folderu, w którym rozpakowaliśmy gtk)\lib\pkgconfig
4. W środowisku Code Blocks wejść w ustawienia kompilatora w zakładce Compiler settings, w polu other compiler options wpisać: `'pkg-config gtk+-3.0 -cflags'`. W zakładce linker settings w polu other linker options wpisać `'pkg-config gtk+-3.0 -libs'`

W systemie linux można skorzystać z załączonego pliku makefile.