

Informatyka geodezyjna 2

Projekt 3

Aplikacja do analizy trasy

Maria Pszczolińska, grupa 3

rok akademicki 2018/2019

1 Opis zadania

Zadanie polega na napisaniu aplikacji z wykorzystaniem frameworka Kivy. Ma ona służyć do analizowania tras zapisanych w formacie gpx.

2 Wczytywanie pliku.gpx

Plik jest wczytywany za pomocą ścieżki umieszczonej w kodzie. Sprawdzenie, czy w pliku zapisane są informacje o wysokościach i czasie zostało wykonane za pomocą następującej funkcji:

```
def wczytaj_plik(filename):  
    el = []  
    dates = []  
    with open(filename, 'r') as gpx_file:  
        gpx_dane = gpx.parse(gpx_file)  
        for track in gpx_dane.tracks:  
            for seg in track.segments:  
                for point in seg.points:  
                    # jeżeli informacja o elewacji dostępna  
                    el.append(point.elevation)  
                    point.time = point.time.replace(tzinfo=None)  
                    dates.append(point.time)  
    return el, dates
```

3 Analiza pliku .gpx

Do wyznaczenia całkowitej długości trasy, pobrane z pliku .gpx dane zostały przekształcone na współrzędne X,Y,Z.

```

for fi ,la ,h in zip (lat ,lon ,ele):
a=6378245.000;
e2=0.006693421622;
N=a/math.sqrt(1-e2*((math.sin ( fi ))**2));
X=(N+h)*math.cos ( fi )*math.cos (la)
Y=(N+h)*math.cos ( fi )*math.sin (la)
Z=((1-e2)*N+h)*math.sin ( fi )

#FUNKCJA LICZACA ODLEGLOSC
def foo(it):
it = iter(it)
f = it.next()
for s in it:
yield f, s
f = s
odleglosc=math.sqrt([(x[1] - x[0] for x in foo(X))**2+[(y[1] - y[0]) for y in foo(Y))**2])
cala=sum(odleglosc)

```

```
def wczytaj_plik(filename):
    el = []
    dates = []
    with open(filename, 'r') as gpx_file:
        gpx_dane = gpx.parse(gpx_file)
        for track in gpx_dane.tracks:
            for seg in track.segments:
                for point in seg.points:
                    # jeżeli informacja o elewacji dostępna
                    el.append(point.elevation)
                    point.time = point.time.replace(tzinfo=None)
                    dates.append(point.time)
    return el, dates
```

```
plik2= open('odleglosc.txt','w')
szer=40
plik2.write('-' * szer)
plik2.write("\n|_{: ^10}|_\\n".format("odleglosc_ [m]"))
plik2.write('-' * szer)
...OBLICZENIA...
```

```
plik2.write( '\n|_{: ^10}| '.format( '%.3f' %cala ))
```

Trasa zostanie zaznaczona na mapie jako punkty wczytane z pliku gpx.



Rysunek 1: Przykład zaznaczonej trasy

Możliwe jest również wczytanie dowolnego obrazu, który będzie odpowiadał znacznikowi punktu na mapie.

```
def mark_point(self, lat, lon, layer=None, markerSource="dot.png"):
    if lat != None and lon != None:
        marker = MapMarker(lat=lat, lon=lon, source=markerSource)
        self.my_map.add_marker(marker, layer=layer)
```