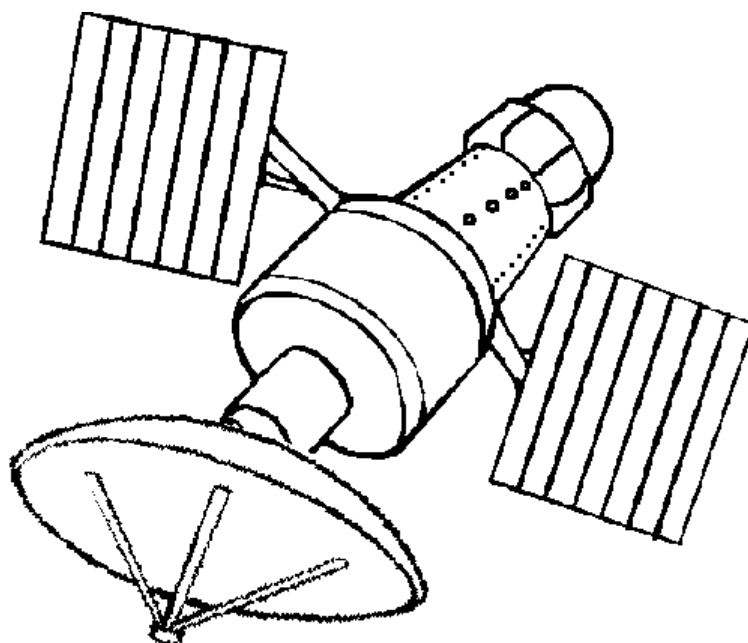


POLITECHNIKA WARSZAWSKA

WYDZIAŁ MECHANICZNY ENERGETYKI I LOTNICTWA

ZAKŁAD SILNIKÓW LOTNICZYCH

Mechanika Nieba
Satellite Ground Tracker



Prowadzący: dr inż. Mateusz Żbikowski

Wykonał: Kamil Macura

Warszawa STYCZEŃ 2018

Spis Treści

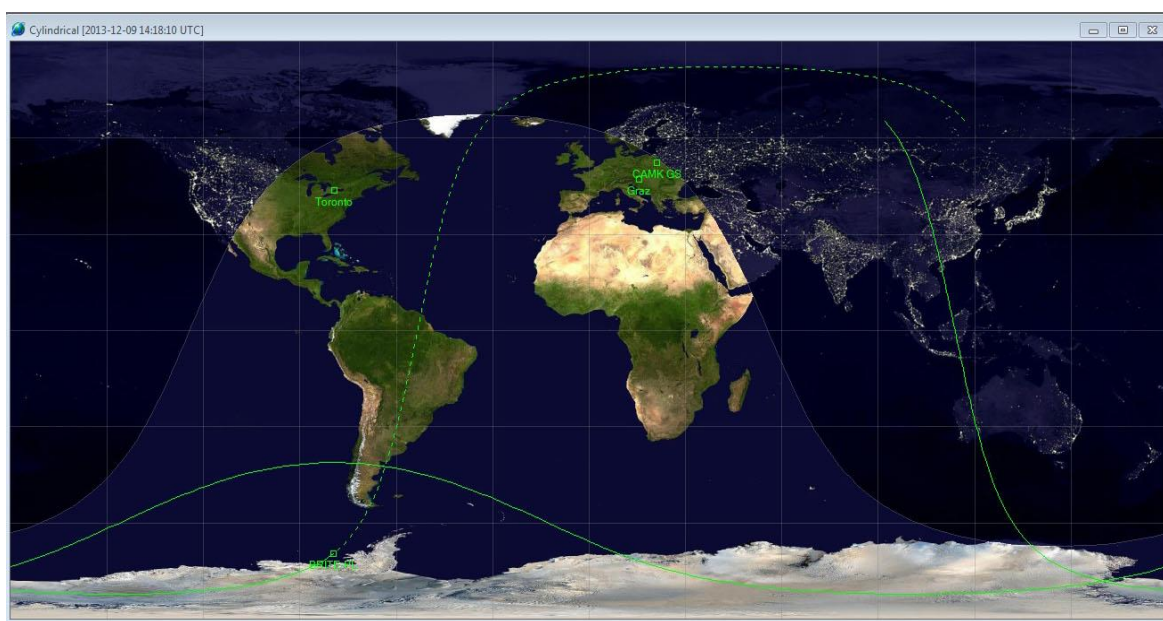
1. Wstęp	3
2. Satellite ground tracker.....	3
3. TLE	4
4. Rodzaje orbit okołoziemskich	5
5. Program	8
5.1 Opis kodu źródłowego	8
5.2 Ograniczenia programu.....	12
6. Wnioski	12
7. Literatura	14

1. Wstęp

W coraz większym stopniu uzależniamy się od satelitów, korzystamy dzięki nim z nawigacji satelitarnej, telewizji, internetu a w przyszłości i wielu więcej dobrodziejstw o których aktualnie nie zdajemy sobie sprawy. Stąd niezwykle istotną sprawą jest wyznaczenie aktualnej pozycji satelity aby możliwy był jak najlepszy odbiór nadawanego przez nią sygnału. Posiadanie aktualnej pozycji jest bardzo istotne ze względu na kontakt z satelitą, gdyż często nie mamy możliwości skontaktowania się z satelitą przez cały czas ruchu lecz tylko i wyłącznie gdy satelita jest nad danym obszarem anten nadawczych. W związku z tym jako cel projektu przyjęto stworzenie programu do śledzenia drogi satelitów (ang. satellite ground tracker).

2. Satellite ground tracker

Ground track jest to ścieżka na powierzchni ziemi bezpośrednio poniżej satelity. Jest to projekcja orbity satelity na powierzchni ziemi [10]. Śledzeniem drogi satelitów głównie zajmują się firmy i instytucje naukowe wysyłające satelity oraz pasjonaci. Programy tworzone są najczęściej przez pasjonatów i hobbystów, dlatego znaczna większość programów pozwalających śledzić położenie satelitów jest typu Open-Source, większe firmy bądź instytucje często korzystają z tego typu oprogramowania, przykładem jest centrum kontroli satelitów w Centrum Astronomicznym im. Mikołaja Kopernika które korzysta właśnie z oprogramowania Open-Source.



Rysunek 1 Zdjęcie programu wykorzystywanego w CAMK [15]

Program ten korzysta z plików TLE dostępnych na stronie[1], format TLE jest standardem dla rozmieszczenia elementów orbitalnych orbitujących wokół Ziemi. Pliki te są ogólnodostępne i często bywają uaktualniane. W związku z tym aby poprawnie określać pozycję należy aktualizować plik TLE. Tablice TLE są szeroko stosowane jako dane wejściowe do projektowania przyszłych śladów orbitalnych śmieci kosmicznych w celu scharakteryzowania przyszłych zdarzeń w celu wsparcia analizy ryzyka, analizy zbliżonego podejścia, manewrów unikania kolizji.

3. TLE [3]

Two-Line Element (TLE)- format danych parametrów orbity dla obiektów orbitujących wokół ziemi w danym czasie. Siły powietrzne USA śledzą wszystkie wykrywalne obiekty na orbicie Ziemi, tworząc TLE dla każdego z nich. Stosując odpowiednią formułę predykcji, stan (położenia i prędkości) w dowolnym momencie w przeszłości lub przyszłości można oszacować z pewną dokładnością. Plik TLE jest plikiem tekstowym zapisanym w dwóch siedemdziesięciokolumnowych wierszach ASCII. TLE może zawierać wiersz tytułu poprzedzający dane, więc każda lista może zajmować trzy wiersze pliku.

Plik TLE dla ISS (ZARYA):

ISS (ZARYA)

1 25544U 98067A 08264.51782528 -.00002182 00000-0 -11606-4 0 2927

2 25544 51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537

Znaczenie poszczególnych linii:

Pole	Kolumny	Zawartość	Przykład
1	01–24	Nazwa satelity	ISS (ZARYA)

Tabela 1 Linia tytułowa pliku TLE

[illegible]

Rysunek 2 Rozmieszczenie kolumn linii tytułowej w pliku TLE

Pole	Kolumny	Zawartość	Przykład
1	01–01	Numer linii	1
2	03–07	Numer satelity	25544
3	08–08	Klasyfikacja (U= Niesklasyfikowany)	U
4	10–11	Międzynarodowy desygnator (Ostatnie dwie cyfry roku startu)	98
5	12–14	Międzynarodowy desygnator (Numer startu w roku)	067
6	15–17	Międzynarodowy desygnator	A
7	19–20	Rok epoki (Ostatnie dwie cyfry)	08
8	21–32	Epoka(Dzień roku I ułamkowa część dnia)	264.51782528
9	34–43	Pierwsza pochodna średniej prędkości obiegu podzielona przez 2 (przyjęty punkt dziesiętny)	-.00002182
10	45–52	Pierwsza pochodna średniej prędkości obiegu podzielona przez 6 (przyjęty punkt dziesiętny)	00000-0
11	54–61	BSTAR – opór (przyjęty punkt dziesiętny)	-11606-4
12	63–63	Numer 0 (pierwotnie powinien tu być typ efemerydy)	0
13	65–68	Numer zestawu elementów, zwiększany gdy generowana jest nowa tablica danych dla danego obiektu	292
14	69–69	Suma kontrolna (modulo 10)	7

Tabela 2 Linia pierwsza pliku TLE

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
1	2	5	5	4	4	4	U	9	8	0	6	7	A				0	8	2	6	4	.	5	1	7	8	2	5	2	8		-	.	0	0	0	0	2	1	8	2		0	0	0	0	0	-	0		-	1	1	6	0	6	-	4		0		2	9	2	7			
1	2			3		4			5		6			7			8										9							10							11							12	13			14																

Rysunek 3 Rozmieszczenie pól i kolumn pierwszej linii w pliku TLE

Pole	Kolumny	Zawartość	Przykład
1	01–01	Numer linii	2
2	03–07	Numer satelity	25544
3	09–16	Inklinacja (stopnie)	51.6416
4	18–25	Długość węzła wstępującego (stopnie)	247.4627
5	27–33	Ekscentryczność (przyjęty punkt dziesiętny)	0006703
6	35–42	Długość pericentrum (stopnie)	130.5360
7	44–51	Anomalia średnia (stopnie)	325.0288
8	53–63	Średnia prędkość obiegu (obiegów na dzień)	15.72125391
9	64–68	Liczba obiegów na epokę	56353
10	69–69	Suma kontrolna (modulo 10)	7

Tabela 3 Linia druga pliku TLE

[illegible]

Rysunek 4 Rozmieszczenie pól i kolumn drugiej linii w pliku TLE

Gdy przyjmowane są punkty dziesiętne, wartość z tabeli musi zostać poprzedzona przecinkiem a ostatnie dwa symbole to wykładnik potęgi w tym przypadku dziesiątki. W takim przypadku wartość w polu 11 z linii 1 (-11606-4) należy zapisać jako $(-0.11606 \times 10^{-4})$.

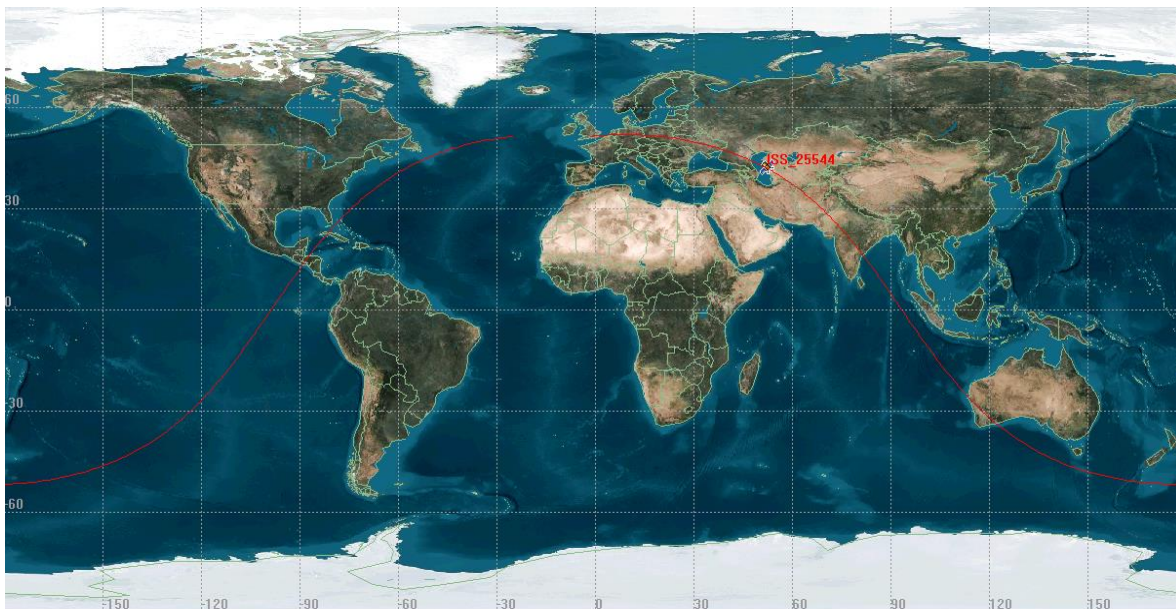
4. Rodzaje orbit okołoziemskich [2]

Krzywą na ziemi jaką zakreśla poruszający się obiekt zależy od orbity na jakiej się znajduje satelita.

Wyróżnia się kilka typów orbit:

Low Earth Orbit (LEO):

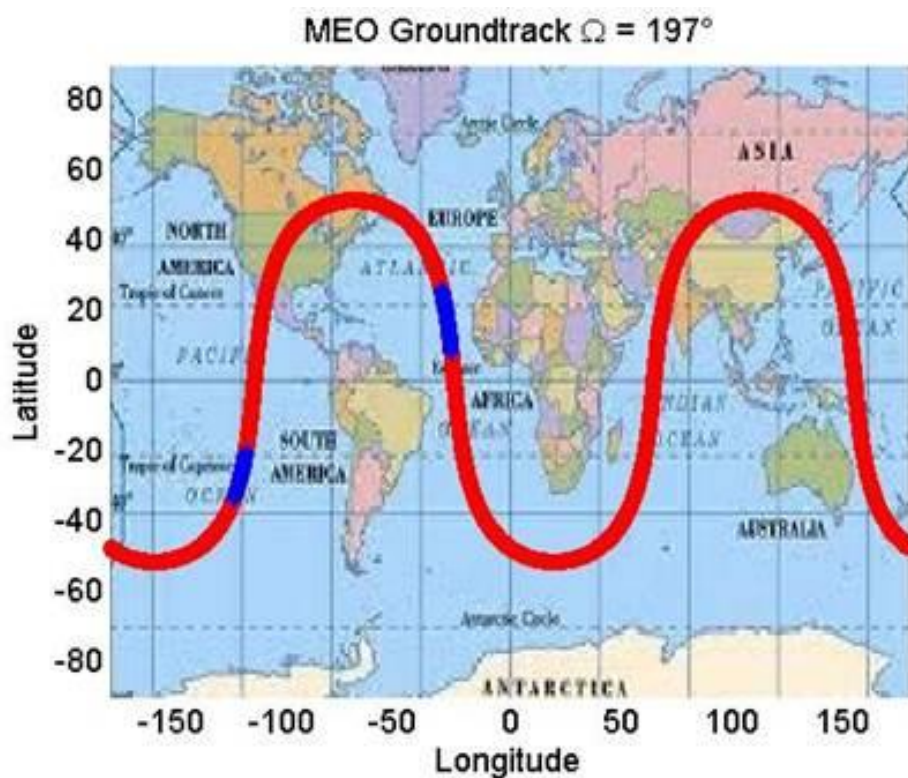
- Niska orbita okołozemska do wysokości $100 \div 1500$ km;
- Okres obiegu ok. 90 minut;
- Zasięg orbity znajduje się poza pasami Van Allena.



Rysunek 5 Ślad satelity na LEO [11]

Medium Earth Orbit (MEO):

- Wysokość orbity 8000 ÷ 20000 km;
- Orbita doskonale nadaje się do używania przez satelity telekomunikacyjne;
- Satelity są narażone na zwiększone promieniowanie ze strony pasów Van Allena.



Rysunek 6 Ślad satelity na MEO [12]

Geosynchroniczna (GEO):

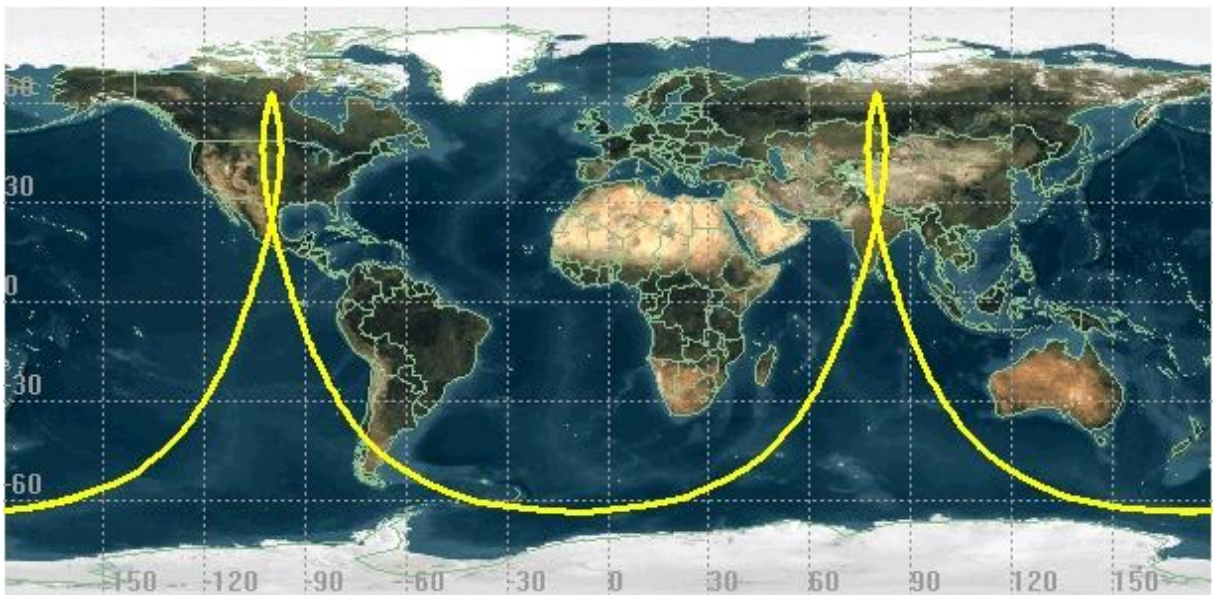
- Wysokość orbity 35786 km;
- Okres obiegu 24h, satelita znajduje się w stałym nad powierzchnią ziemi;
- Orbita jest używana przez satelity komercyjne i wojskowe.



Rysunek 7 Ślad satelity na GEO [13]

High Earth (Elliptical) Orbit (HEO):

- Rozwijana jako alternatywa dla Geosynchronicznej orbity, głównie przez Rosjan;
- Dogodna dla telekomunikacji głównie z północnymi szerokościami geograficznymi;
- Apogeum można „umieścić” w interesującym nas obszarze;
- Jednym z typów tej orbity jest Molniya.



Rysunek 8 Ślad satelity na HEO (Molniya) [14]

Synchronizowana słonecznie:

- „Wędrująca orbita, płaszczyzna orbity obraca się prawie takim samym okresem jak okres obrotu Ziemi wokół Słońca. Satelita przechodzi przez perigeum w tym samym słonecznym czasie lokalnym;
- Orbita ta nadaje się satelity mapującej powierzchnię planety.

Orbita polarna:

- Kąt inklinacji wynosi 90^0 , użyteczna dla mapowania planety lub obserwacji pola walki.

5. Program

Program został napisany w języku Python 3.6.3 z wykorzystaniem ogólnodostępnych bibliotek takich jak pyephem[4], orbitalPy[5] które posłużyły do pracy z plikiem typu TLE. Dzięki bibliotece geocoder[6] udało się uzyskać aktualną pozycję obserwatora wykorzystując lokalizację IP a stosując googlemaps[7] jest możliwość przeniesić obserwatora w dowolne miejsce na ziemi. Używając bibliotekę requests program jest w stanie przy każdym uruchomieniu programu pobierać dane ze strony [1], mapa generowana jest przy użyciu basemap[8] która w połączeniu z matplotlib umożliwiła stworzenie animowanego ruchu satelitów rysowanego na mapie w czasie rzeczywistym. Wartości otrzymane w programie były porównywane z trackerami dostępnymi w sieci [9] i ich wartości były zgodne. W związku z tym program można uznać za działający poprawnie i wykorzystać jego obliczenia do dalszych działań.

5.1 Opis kodu źródłowego

Kod źródłowy znajduje się na github [17]. Kolejność działania programu przedstawia się następująco:

Pierwszym etapem pracy programu jest wybór dla jakiego typu obiektów orbitujących wokół ziemi chcemy wyznaczyć położenia. Do wyboru jest aktualnie 20 rodzajów i są to:

- Stacje kosmiczne
- Satelity wystrzelone w ostatnich 30 dniach
- Satelity nawigacyjne – GPS
- Satelity nawigacyjne – GLONASS
- Satelity nawigacyjne – GALILEO
- Satelity nawigacyjne – BEIDOU
- Satelity systemu wspomagającego - WAAS/EGNOS/MSAS
- Satelity nawigacyjne - NAVY(NNSS)
- Satelity nawigacyjne Rosyjskie na LEO
- Satelity pogodowe
- Satelity geostacjonarne
- Satelity naukowe
- Satelity geodezyjne

- Satelity inżynierskie
- Satelity edukacyjne
- Satelity amatorskie
- Satelity eksperymentalne
- Satelity militarne
- Satelity typu cubesat
- Satelity Molniya

Rodzaje obiektów orbitujących wokół ziemi zostały dobrane subiektywnie i nie są to wszystkie dostępne typy na stronie[1] ale jest ich znaczna większość. Rodzaje te są zapisane w formie słownika tak więc aby dodać kolejny wystarczy dodać jako klucz nazwę grupy satelitów a wartość to link do pliku tekstowego z danymi TLE. Po wybraniu typu obiektu program łączy się z odpowiednim linkiem dla wybranego typu i pobiera dane TLE zapisując je w pliku tekstowym pod nazwą wybranego rodzaju satelitów. Można pobrać i zapisać plik z danymi na dysku i odczytywać dane bez konieczności łączenia się z siecią, jednak bardzo ważne jest aktualizowanie danych dlatego dla każdego uruchomienia programu pobierane są aktualne dane. Następnym krokiem jest wyodrębnienie ze stworzonego pliku txt nazw satelitów i wypisanie ich po czym należy wybrać dla jakich satelitów określone ma być położenie. Po wybraniu odpowiadających obiektów program tworzy listę list danych satelitów np. w przypadku gdy wybrany zostanie jeden obiekt lista wygląda następująco:

[Linia tytułowa, Linia 1, Linia 2]

gdzie linie odpowiadają linią pliku TLE w przypadku większej ilości satelitów lista ta wygląda następująco:

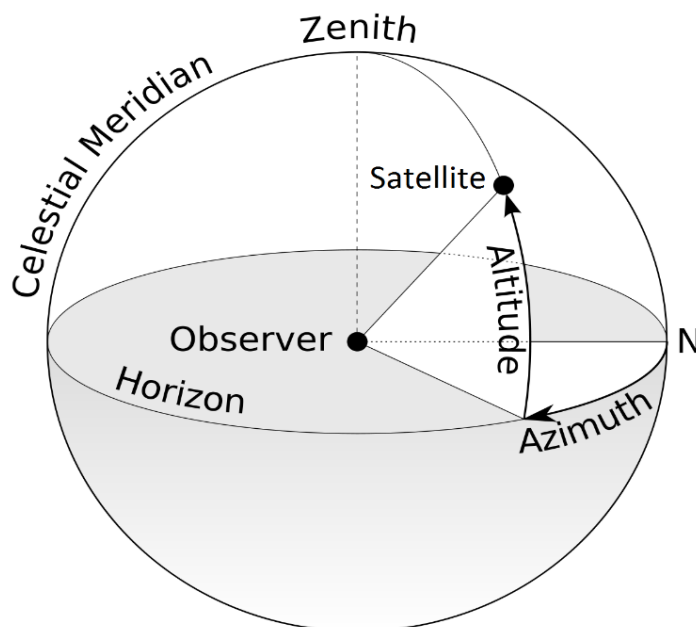
Satelita 1	Satelita 2
[[Linia tytułowa, Linia 1, Linia 2],	[Linia tytułowa, Linia 1, Linia 2],...]

Mając tablicę danych dla wybranych obiektów następnym krokiem jest określenie pozycji obserwatora. Do wyboru mamy dwie opcje:

- Określenie pozycji używając adresu IP;
- Wykorzystując Google maps.

W pierwszym przypadku z pomocą przychodzi biblioteka geocoder która automatycznie wykorzystując IP zwraca miejsce, długość i szerokość geograficzną. Jeżeli wybierzemy drugą opcję należy wpisać miejsce gdzie ma być obserwator w taki sam sposób jak w google maps i również otrzymamy szerokość i długość geograficzną. Mając już te dane jesteśmy w stanie wyznaczyć położenie satelity. Następnie dla wyznaczonego położenia, program korzystając z google Elevation API wyznacza przewyższenie które

jest niezbędne do wyznaczenia azymutu (ang.azimuth) i wyniesienia (ang.altitude) satelity.



Rysunek 9 Przedstawienie azymutu i wyniesienia [16]

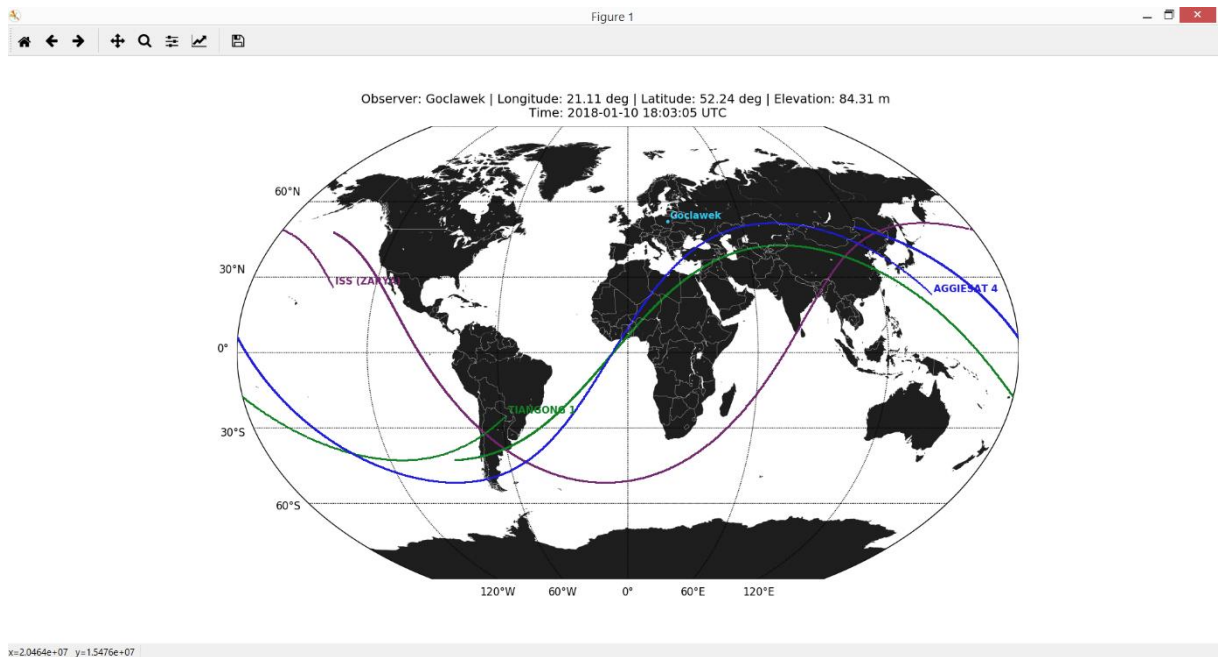
Mając już wszystkie potrzebne dane dla obiektu orbitującego oraz obserwatora można przystąpić do obliczania pozycji satelity w czasie rzeczywistym oraz parametrów jej orbity.

Wykorzystując bibliotekę orbitalpy jesteśmy w stanie wyznaczyć parametry orbity bezpośrednio wykorzystując plik TLE. Parametry orbity satelity wyznaczone w programie dla Polskiego satelity BRITE-PL 2:

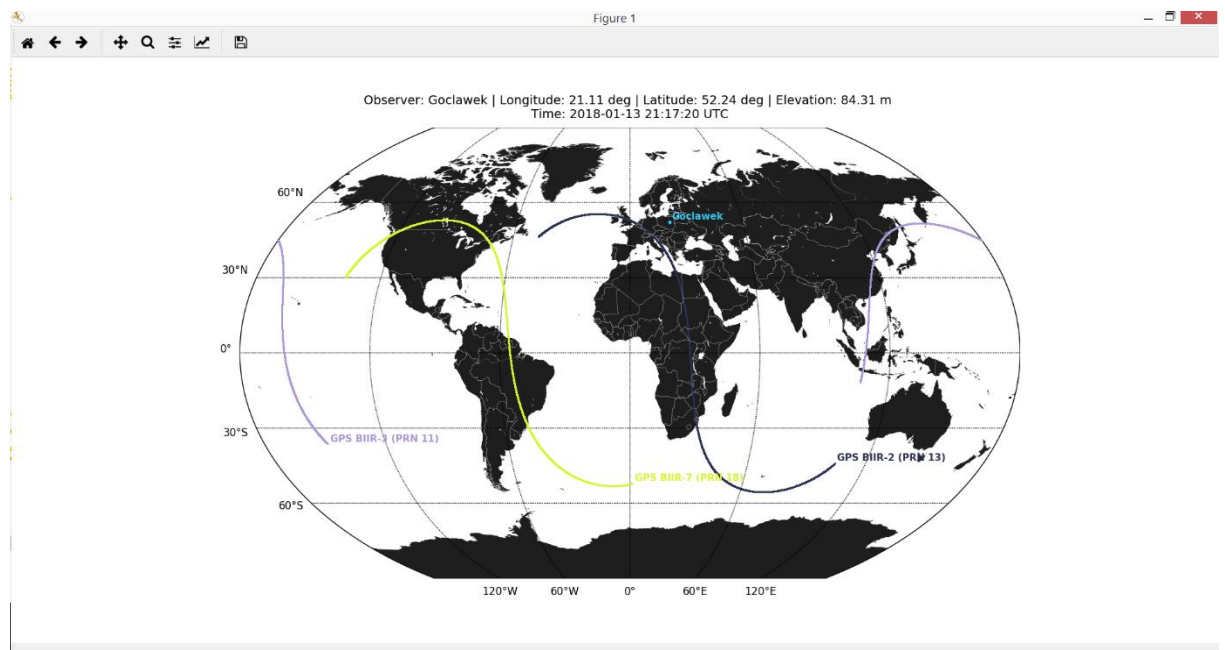
Pół wielka	= 7000.409 [km]
Mimośród:	= 0.002621 [-]
Inklinacja:	= 97.9 [deg]
Długość węzła wstępującego	= 100.9 [deg]
Długość pericentrum	= 105.1 [deg]
Anomalia średnia	= 255.2 [deg]
Anomalia mimośrodowa	= 255.0 [deg]
Okres	= 01:37:09 [H]
Prędkość średnia	= 7.5407 [km/s]

Kolejnym krokiem programu jest określenie parametrów wizualizacji położenia obiektu. W tym przypadku wykorzystana została biblioteka basemap. Jako typ mapy wykorzystany został Kavrayskiy VII, wszystkie lądy na mapie przedstawione zostały jako czarne poprzecinane białymi konturami państw a kolor wód został ustawiony jako biały. Osie zostały podpisane przez odpowiadające na mapie długości i szerokości geograficzne. Jako tytuł wykresu wyświetlana jest nazwa miejsca obserwatora oraz jego położenie, a poniżej jest zegar z czasem UTC który jest aktualizowany co dwie sekundy. Miejsce obserwatora zaznaczone na mapie jest poprzez kropkę z adnotacją powyżej. Kolory satelitów są generowane losowo, w zależności od ilości wybranych obiektów tyle samo generowanych jest kolorów. W związku z tym kolor wizualizacji każdego z

satelitów jest inny. Ponadto każdy satelita ma adnotację ze swoją nazwą w kolorze zgodnym z kolorem jego trasy na mapie.



Rysunek 10 Wygląd programu.



Rysunek 11 Wygląd programu

Następnym etapem pracy programu jest pętla nieskończona, w której co dwie sekundy wyznaczany jest czas UTC i przypisany zostaje jako czas obserwatora. Kolejnym etapem jest pętla która iteruje poprzez każdą z wybranych satelitów. Dalej wykorzystując bibliotekę pyephem dane wybranych satelitów są podstawiane i zestawiane z danymi obserwatora. Kolejnym krokiem programu jest zwrócenie położenia satelity to znaczy jej długości i szerokości geograficznej oraz azymutu i wyniesienia. Mając dane satelity program przelicza wartości długości i szerokości geograficznej na wartości dla których możliwe jest narysowanie drogi satelity na mapie.

Położenie satelitów jest rysowane na mapie a ich dokładne parametry są wypisywane z dwu sekundowymi odstępami. Wyznaczone położenie dla trzech satelitów wygląda następująco dla czasu 2018-01-12 23:58 i obserwatora w Warszawie:

HST:

Altitude: -75.4 deg | Azimuth: 195.88 deg | Longitude: -92.38 deg |

Latitude: -19.58 deg

BRITE-PL 2:

Altitude: -34.38 deg | Azimuth: 160.89 deg | Longitude: 147.93 deg |

Latitude: -69.99 deg

FLYING LAPTOP:

Altitude: -42.58 deg | Azimuth: 330.77 deg | Longitude: 2.23 deg |

Latitude: 60.02 deg

5.2 Ograniczenia Programu

Program ze względu na każdorazową aktualizację plików TLE i pobieranie pozycji obserwatora musi być połączony z siecią, co uniemożliwia korzystanie z programu w trybie offline. Kolejnym ograniczeniem jest posiadanie własnego klucza do Elevation API, klucz ten jest darmowy i można go stosunkowo łatwo uzyskać. Jeżeli chodzi o program jest on w pełni skalowalny i nie ma problemu przy wyborze większej ilości satelitów. W programie nie są przypisane wszystkie możliwe satelity ale możliwe jest ich dodanie w sposób prosty na początku programu.

6. Wnioski

W ramach projektu na przedmiot Mechanika Nieba powstał program do analizowania ruchu satelitów okołoziemskich. Program spełnia swoją rolę w pełni jako satellite ground tracker.

Zwraca on wymagane parametry dla trackera takie jak:

- Długość i szerokość geograficzną obiektu obserwowanego;
- Położenie obserwatora;
- Położenie satelity względem obserwatora;
- Parametry orbity.

Nie ustępuje on niczym pod względem zwracanych parametrów trackerom dostępnym w sieci, jednakże aby był to kompaktowym program do śledzenia drogi satelitów należało by dodać graficzny interfejs użytkownika który usprawnił by pracę z programem. Program jest bardzo edukacyjny gdyż wizualizuje ślad orbity na powierzchni ziemi i pozwala zauważyć różnice pomiędzy orbitami przy zmianie parametrów takich jak:

- Półoś wielka;
- Mimośród;
- Inklinacja
- Długość węzła wstępującego;
- Długość pericentrum;
- Anomalia średnia.

Oprócz powyższych wniosków należy zwrócić uwagę na to, że projekt był bardzo edukujący dla autora, gdyż w jego skład wchodziła nie tylko nauka o ruchu satelitów i orbitach okołoziemskich ale także nauka bibliotek astronomicznych języka Python.

7. Literatura

- [1] <http://www.celestrak.com/NORAD/elements/> [2017-11-24]
- [2] <http://studylibpl.com/doc/901918/wstep-do-mechaniki-nieba> [2018-01-14]
- [3] https://en.wikipedia.org/wiki/Two-line_element_set [2017-11-24]
- [4] <http://rhodesmill.org/pyephem/> [2017-11-24]
- [5] <http://pythonhosted.org/OrbitalPy/> [2017-11-24]
- [6] <http://geocoder.readthedocs.io> [2017-11-24]
- [7] <https://github.com/googlemaps/google-maps-services-python> [2017-11-24]
- [8] <https://matplotlib.org/basemap/> [2017-11-24]
- [9] <http://www.n2yo.com> [2018-01-14]
- [10] https://en.wikipedia.org/wiki/Ground_track [2018-01-14]
- [11] <https://i.stack.imgur.com/ORd8G.png> [2018-01-14]
- [12] http://ccar.colorado.edu/asen5050/projects/projects_2013/Smith_Alan/Visibility_files/image008.jpg [2018-01-14]
- [13] http://hakenberg.de/numerics/orbits/geostationary_track.jpg [2018-01-14]
- [14] <https://upload.wikimedia.org/wikipedia/commons/thumb/b/bf/Molniya.jpg/400px-Molniya.jpg> [2018-01-14]
- [15] <https://archive.li/tfx1p/210d7bd0a0d6fc8171b07edb47d03a57b23de6f0.jpg> [2018-01-14]
- [16] https://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/Azimuth-Altitude_schematic.svg/1200px-Azimuth-Altitude_schematic.svg.png [2018-01-14]

Program

- [17] <https://github.com/AstroSlazak/Mechanika-Nieba/blob/master/Projekt.py>