

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



École Nationale Polytechnique
Département Automatique

Rapport de stage

Réalisation d'une plateforme web pour visualiser l'espace spatial algérien

Deep Research Network

Rédigé par :

BOUAZZA Tarek

tarek.bouazza@g.enp.edu.dz

Encadrant :

Mr. MOUSSAOUI Zeryab

zeryab.moussaoui@deep-research-network.com

Table des matières

Table des figures	iv
1 Présentation du stage	2
2 Projet et objectifs	3
2.1 Présentation de l'ensemble du projet	3
2.1.1 Contexte général	3
2.1.2 Objectif de la plateforme web	4
2.1.3 Utilisateurs potentiels	4
2.2 Spécification du besoin	4
2.2.1 Spécifications techniques	4
2.2.2 Spécification fonctionnelle	5
3 Travail réalisé	6
3.1 Organisation	6
3.2 Récupération des données des satellites	7
3.2.1 Accès aux données SATCAT	7
3.2.2 Géocodage inversé	8
3.3 Réalisation de la plateforme web	8

3.3.1	Utilisation de CesiumJS	8
3.4	Hébergement du site dans un serveur local	9
3.4.1	Création de serveur HTTP local avec Apache	9
3.4.2	Déploiement du site dans le hôte virtuel	11
Conclusion		12
A Code associé au projet		13

Table des figures

3.1	Données disponibles sur le site CelesTrack	7
3.2	Les données des premiers satellites lancés à l'aide de Skyfield	7
3.3	Recupération des satellites situés au-dessus de l'Algérie	8
3.4	Exemple de visualisation avec CesiumJS	9
3.5	Vérification du fonctionneent du service Apache	10
3.6	La page web par défaut du serveur Apache	10
3.7	La plateforme web déployée dans le hôte virtuel	11

Remerciements

Ce stage pratique effectué sous la tutelle de Mr Zeryab MOUSSAOUI, responsable de Deep Research Network, a été bénéfique à ma formation d'ingénieur grâce aux compétences pratiques acquises durant ces 3 semaines.

Je tiens à remercier Mr MOUSSAOUI Zeryab, d'abord pour m'avoir proposé ce sujet de stage et également pour l'intérêt qu'il a porté à l'évolution de ce travail ainsi qu'au développement de mes connaissances. Je le remercie particulièrement pour son apport incommensurable et sa disponibilité malgré ses multiples occupations, et de m'avoir fourni toute la documentation et l'aide nécessaire.

Enfin, je remercie toutes les personnes ayant contribué de près ou de loin au bon déroulement de ce stage et à l'accomplissement de ce précieux travail.

CHAPITRE 1

Présentation du stage

Ce stage pratique s'inscrit dans le cadre pédagogique de la formation d'ingéniorat d'état et Master 2 en Automatique à l'Ecole Nationale Polytechnique.

Le stage a été proposé par Monsieur Zeryab MOUSSAOUI, ingénieur à Airbus et le responsable du Deep Reseach Network. L'objectif est la réalisation complète d'un projet informatique professionnel dans l'optique de répondre à des besoins spécifiques.

Ce stage est essentiellement centré sur la familiarisation avec de nouveaux outils informatiques, notamment ceux utilisés pour manipuler des données provenant des engins aérospatiaux tels que les satellites, ainsi que la maîtrise des outils et langages du développement des applications web.

CHAPITRE 2

Projet et objectifs

2.1 Présentation de l'ensemble du projet

2.1.1 Contexte général

Plus de 2000 satellites sont en orbite autour de la terre, Ces objets spatiaux qui fournissent des services variés dont les thèmes sont multiples tel que les communications et l'observation de la terre sont en déplacement permanent. Ce qui représente un trafic conséquent, invisible à l'œil nu. Pour cela il est important d'arriver à visualiser la position de ces objets et prévoir leurs passages à travers les moyens qu'on possède.

De plus, l'observation des satellites est un domaine fascinant de l'astronomie. Il est très intéressant d'observer le passage dans le ciel d'un satellite particulier que ce soit au jumelle ou au télescope. Afin d'y arriver, il est nécessaire d'avoir la position et les déplacements des objets se trouvant au-dessus de l'observateur.

2.1.2 Objectif de la plateforme web

L'objectif principal de cette plateforme est de visualiser l'espace spatial algérien. D'une manière plus précise, Il s'agit de permettre aux utilisateurs de voir la position (ainsi que d'autres caractéristiques tel que l'orbite) de tous les satellites actifs en orbite qui se situeraient au-dessus du territoire algérien à l'instant de l'observation et leur offrir un moyen afin de les suivre en temps réel.

2.1.3 Utilisateurs potentiels

Les services que doit fournir la plateforme cadrent parfaitement avec le besoin des centres de recherche en astronomie qui cherchent souvent des moyens pour localiser et suivre les objets spatiaux tels que les satellites. Ces services sont également utiles pour les clubs et associations d'amateurs astronomes et leurs activités d'observation des satellites en orbite.

Ci-dessous une liste des utilisateurs potentiels de notre plateforme :

- *L'Agence Spatiale Algérienne (ASAL).*
- *Le CRAAG, Centre de Recherche en Astronomie, Astrophysique et Géophysique.*
- *Clubs scientifiques et Associations Algériennes de l'Astronomie amateurs*
Tel que : Association Algérienne des Jeunes Astronomes Amateurs (AAJAA),
Association SIRIUS des astronomes amateurs. etc.

2.2 Spécification du besoin

2.2.1 Spécifications techniques

Coté Serveur

La partie Serveur du site qui devra être codée en Python, doit utiliser l'API de SAT-CAT¹ pour récupérer les données des satellites situés au-dessus du territoire algérien (en filtrant le reste des satellites) sous format TLE² à partir de la base de données du site CelesTrack³. Les TLE sont mises à jour quotidiennement afin d'éviter toute dérive dans les calculs.

1. <https://www.celestrak.com/satcat/>

2. https://en.wikipedia.org/wiki/Two-line_element_set

3. <https://www.celestrak.com>

Une fois les TLE des satellites récupérées, un propagateur de trajectoire (tel que le SGP4) doit permettre d'avoir des prévisions de la trajectoire de ces derniers jusqu'aux prochaines mises à jour de la base de données.

Outils suggérés : Python, SATCAT REST API, CelesTrack, Skyfield, Reverse Geocoder, SGP4 propagator / Pykep.

Coté Client

Cette partie doit permettre de visualiser les positions ainsi que les orbites des satellites obtenus à partir du serveur (expliqué précédemment) sur une globe terrestre virtuelle créée à l'aide de la librairie CesiumJS⁴.

Le partage de données de visualisation des satellites entre le serveur Python et le client Javascript (CesiumJS) doit se faire à l'aide de CZML⁵ (des fichiers de format JSON pour décrire des scènes graphiques dynamiques).

Outils suggérés : HTML, CSS, Javascript, CesiumJS, Python, CZML (JSON)

Le site web doit être initialement hébergé localement à l'aide du serveur Apache⁶ pour la phase de conception et tests. Une fois prêt, il sera hébergé sur un serveur web dans une seconde phase.

2.2.2 Spécification fonctionnelle

Afin de capter l'attention des utilisateurs potentiels, un certain nombre de caractéristiques doit être garanti par la plateforme

1. Un site web responsive et efficace qui satisfait le niveau des performances auquel peuvent s'attendre les différents utilisateurs de la plateforme.
2. Permettre aux utilisateurs d'avoir les positions réelles des satellites à chaque fois que les fichiers TLE fournis par CelesTrack sont actualisés.
3. Prédire et visualiser à l'avance l'orbite de chaque satellite sur la plateforme à l'aide d'un propagateur des trajectoires.

4. <https://cesium.com/cesiumjs/>

5. <https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Guide>

6. <https://httpd.apache.org/>

3.1 Organisation

Le stage pratique s'est déroulé en trois grandes phases

- Se familiariser avec les paquets de Python et API permettant de récupérer les TLE des satellites actifs dans l'espace et les filtrer pour garder ceux situés au-dessus du sol algérien.
- Apprendre à utiliser les outils de base du développement frontend (HTML/CSS/JavaScript) et de visualisation géospatiale (CesiumJS) pour construire la plateforme web.
- Découvrir des outils d'hébergement local des sites web et de création des domaines (Apache) pour effectuer des testes et valider le bon fonctionnement de la plateforme.

3.2 Récupération des données des satellites

SATCAT est un format de données qui fournit la liste des satellites connus lancés depuis 1957 avec des informations comme la date de lancement.

Le format TLE code une liste d'éléments orbitaux d'un objet en orbite (satellite) pour un instant donné (epoch), il se compose de deux lignes de 69 caractères.

En tant que service à la communauté des utilisateurs de satellites, CelesTrack est un site qui permet de récupérer le SATCAT et les TLEs.

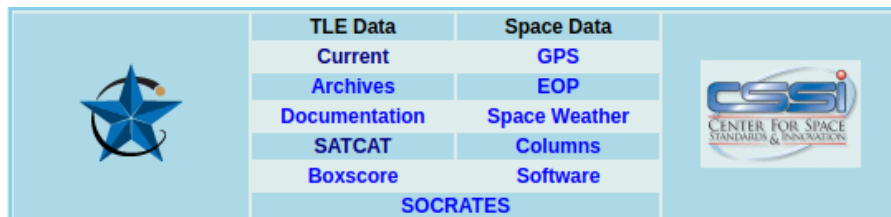


FIGURE 3.1 – Données disponibles sur le site CelesTrack

3.2.1 Accès aux données SATCAT

Pour notre application, nous avons utilisé la librairie Skyfield¹ de Python pour récupérer les TLE des satellites

1957-001A	00001	D SL-1 R/B	CIS	1957-10-04	TYMSC	1957-12-01	96.2	65.1	938	214	20.4200
1957-001B	00002	*D SPUTNIK 1	CIS	1957-10-04	TYMSC	1958-01-03	96.1	65.0	1080	64	N/A
1957-002A	00003	*D SPUTNIK 2	CIS	1957-11-03	TYMSC	1958-04-14	103.7	65.3	1659	211	0.0800
1958-001A	00004	*D EXPLORER 1	US	1958-02-01	AFETR	1970-03-31	88.5	33.2	215	183	N/A
1958-002B	00005	* VANGUARD 1	US	1958-03-17	AFETR		132.7	34.2	3833	651	0.1220
1958-003A	00006	*D EXPLORER 3	US	1958-03-26	AFETR	1958-06-28	103.6	33.5	1739	117	N/A
1958-004A	00007	D SL-1 R/B	CIS	1958-05-15	TYMSC	1958-12-03	102.7	65.1	1571	206	N/A
1958-004B	00008	*D SPUTNIK 3	CIS	1958-05-15	TYMSC	1960-04-06	88.4	65.1	255	139	11.8400
1958-005A	00009	*D EXPLORER 4	US	1958-07-26	AFETR	1959-10-23	92.8	50.3	585	239	N/A
1958-006A	00010	*D SCORE	US	1958-12-18	AFETR	1959-01-21	98.2	32.3	1187	159	N/A
1959-001A	00011	* VANGUARD 2	US	1959-02-17	AFETR		121.5	32.9	2939	553	0.3931
1959-001B	00012	VANGUARD R/B	US	1959-02-17	AFETR		125.8	32.9	3327	554	0.5266
1959-002A	00013	*D DISCOVERER 1	US	1959-02-28	AFWTR	1959-03-03	92.7	90.0	697	114	N/A
1959-003A	00014	*D DISCOVERER 2	US	1959-04-13	AFWTR	1959-04-26	90.4	89.9	346	239	N/A
1959-004A	00015	*D EXPLORER 6	US	1959-08-07	AFETR	1961-06-30	762.3	47.0	42276	251	N/A
1958-002A	00016	VANGUARD R/B	US	1958-03-17	AFETR		137.3	34.3	4226	651	0.2282
1959-004B	00017	D THOR ABLE R/B	US	1959-08-07	AFETR	1961-06-30	719.0	47.1	40215	198	0.0200
1959-005A	00018	*D DISCOVERER 5	US	1959-08-13	AFWTR	1959-09-28	89.1	80.0	323	137	N/A
1959-006A	00019	*D DISCOVERER 6	US	1959-08-19	AFWTR	1959-10-20	90.9	84.0	469	171	N/A
1959-007A	00020	* VANGUARD 3	US	1959-09-18	AFETR		124.6	33.4	3264	507	0.6412
1959-008A	00021	* LUNA 3	CIS	1959-10-04	TYMSC	1960-04-20	21563.2	55.0	499998	500	N/A
1959-009A	00022	* EXPLORER 7	US	1959-10-13	AFETR		96.3	50.3	679	485	0.5003
1959-009B	00023	D JUNO II R/B	US	1959-10-13	AFETR	1989-07-16	89.2	50.3	249	225	0.5778
1959-010A	00024	*D DISCOVERER 7	US	1959-11-07	AFWTR	1959-11-26	91.5	81.6	539	161	N/A
1959-011A	00025	*D DISCOVERER 8	US	1959-11-20	AFWTR	1960-03-08	88.0	80.7	239	115	N/A
1959-005B	00026	D DISCOVERER 5 CAPSULE	US	1959-08-13	AFWTR	1961-02-11	88.7	78.9	280	143	N/A
1960-001A	00027	* PIONEER 5	US	1960-03-11	AFETR						0.0008 SU0
1960-002A	00028	D THOR ABLE R/B	US	1960-04-01	AFETR	1991-07-03	87.8	48.3	168	163	1.6017
1960-002B	00029	*- TIROS 1	US	1960-04-01	AFETR		97.7	48.4	664	630	0.8030
1960-003A	00030	D THOR ABLESTAR R/B	US	1960-04-13	AFETR	1961-08-18	88.4	51.3	194	194	N/A
1960-003B	00031	*D TRANSIT 1B	US	1960-04-13	AFETR	1967-10-05	88.7	51.2	220	201	N/A
1960-004A	00032	*D DISCOVERER 11	US	1960-04-15	AFWTR	1960-04-26	89.7	80.1	377	142	N/A

FIGURE 3.2 – Les données des premiers satellites lancés à l'aide de Skyfield

1. <https://rhodesmill.org/skyfield/api.html>

3.2.2 Géocodage inversé

Le géocodage inversé consiste à effectuer l'opération inverse du géocodage, c'est-à-dire d'attribuer une adresse à des coordonnées géographiques (latitude et longitude). Pour notre besoin, nous l'avons utilisé pour filtrer la liste des Satellites récupérées par Skyfield et garder uniquement ceux qui se trouvent au-dessus du territoire algérien.

Ci-dessous un exemple des données collectées des satellites situés au-dessus de l'Algérie avec Skyfield et un paquet Python `reverse_geocoder`².

Retrieving data of Satellites that are above Algeria

```
from skyfield import api as sf
from skyfield.api import EarthSatellite, Topos, load
import reverse_geocoder as rg

# getsat and getlonlat functions already defined

SatList = []
for name in names:
    sat = getsat(name)
    ts = load.timescale()
    t = ts.now()
    geometry = sat.at(t)
    subpoint = geometry.subpoint()
    alt = float(subpoint.elevation.km)
    if alt < 1000:
        longitude, latitude = getlonlat(subpoint)
        coordinates = (longitude, latitude) # Defining the satellite's coordinates
        SatLocation = rg.search(coordinates) # Reverse geocoding to get the satellite's projection
        SatLocation[0]
        if SatLocation[0]['cc'] == 'DZ':
            altitude = int(subpoint.elevation.km)
            SatList.append({'Satellite':name, 'Elevation': altitude})
            print('Satellite (Name/ID): ', name, '-', 'Hauteur (km): ', altitude)

Satellite (Name/ID): 14032 - Hauteur (km): 518
Satellite (Name/ID): COSMOS 1455 - Hauteur (km): 518
Satellite (Name/ID): 24298 - Hauteur (km): 846
```

FIGURE 3.3 – Recupération des satellites situés au-dessus de l'Algérie

3.3 Réalisation de la plateforme web

3.3.1 Utilisation de CesiumJS

CesiumJS est un outil open source de création de cartes 2D et de globes 3D au sein d'un navigateur web. Il utilise une librairie JavaScript comme langage de programmation et WebGL comme interface 3D dynamique pour les rendus matériels accélérés.

2. https://pypi.org/project/reverse_geocoder/

La visualisation géospatiale dynamique 3D de CesiumJS permet de voir le relief mondial en haute résolution.

Pour interpréter les données 3D et 2D implantées dans CesiumJS, l'application utilise (CZML) au format JSON afin de décrire les scènes graphiques dynamiques.

L'ensemble des composants "CesiumJS" est entièrement open source, il est téléchargeable pour l'installer et l'utiliser localement.

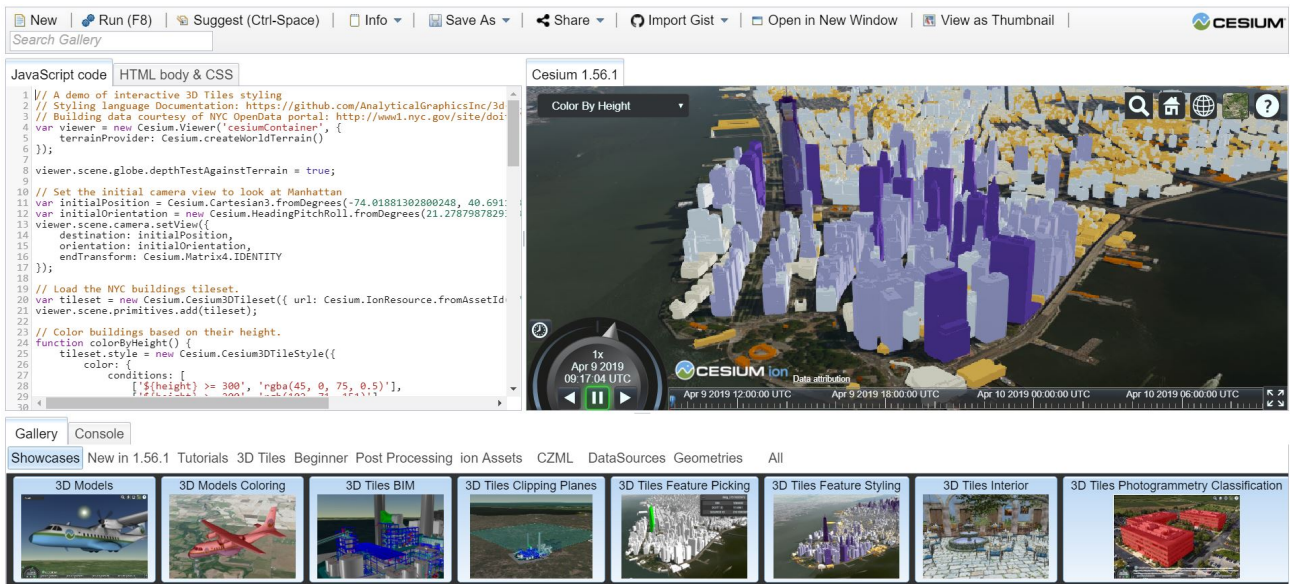


FIGURE 3.4 – Exemple de visualisation avec CesiumJS

3.4 Hébergement du site dans un serveur local

Un serveur local simule au mieux les conditions réelles d'un serveur d'hébergement web, il est idéal pour tester les différentes techniques et outils au cours du développement d'un site web.

3.4.1 Création de serveur HTTP local avec Apache

Apache est un logiciel de serveur Web Open Source développé par Apache Software Foundation. La serveur écoute sur le port 80 pour les requête HTTP et sert la mise à disposition de fichiers HTML.

Un serveur HTTP permet d'héberger des sites web qui seront accessibles avec un navigateur. Il permet au site web de communiquer avec le navigateur en utilisant le protocole HTTP.

Il est possible de combiner le logiciel Apache avec des bases de données tels que MySQL³ mais aussi avec des scripts PHP⁴, Python afin de permettre la création de pages dynamiques.

Environnement

L'environnement dans lequel le serveur Apache a été installé et configuré est la version 18.04 de Ubuntu, un système d'exploitation GNU/Linux.

```
tarek@Tarek: ~ 80x24
tarek@Tarek:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Tue 2020-02-04 11:46:01 CET; 5h 44min ago
   Process: 1110 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCE
   Main PID: 1155 (apache2)
      Tasks: 55 (limit: 4915)
     CGroup: /system.slice/apache2.service
             └─1155 /usr/sbin/apache2 -k start
               └─1156 /usr/sbin/apache2 -k start
                 └─1157 /usr/sbin/apache2 -k start

févr. 04 11:45:59 Tarek systemd[1]: Starting The Apache HTTP Server...
févr. 04 11:46:01 Tarek systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

FIGURE 3.5 – Vérification du fonctionnement du service Apache

Après avoir installé et activé le serveur HTTP Apache dans l'environnement Ubuntu, voici le message qui s'affiche par défaut dans le navigateur.

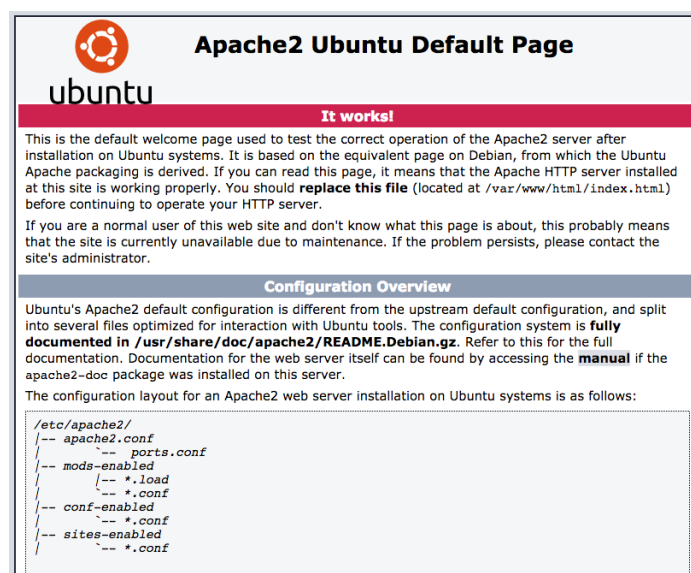


FIGURE 3.6 – La page web par défaut du serveur Apache

3. <https://www.mysql.com/>

4. <https://www.php.net/>

3.4.2 Déploiement du site dans le hôte virtuel

La mise en place de notre espace de travail se fait grâce à la fonctionnalité d'hôte virtuel (VirtualHost) dont dispose le serveur HTTP Apache.

C'est une fonctionnalité qui permet d'héberger plusieurs domaines, tout en personnalisant entièrement la configuration de chaque domaine sur un seul serveur Web.

Dans notre cas, le hôte virtuel a été créé pour le déploiement local de la plateforme web pour tester son fonctionnement.

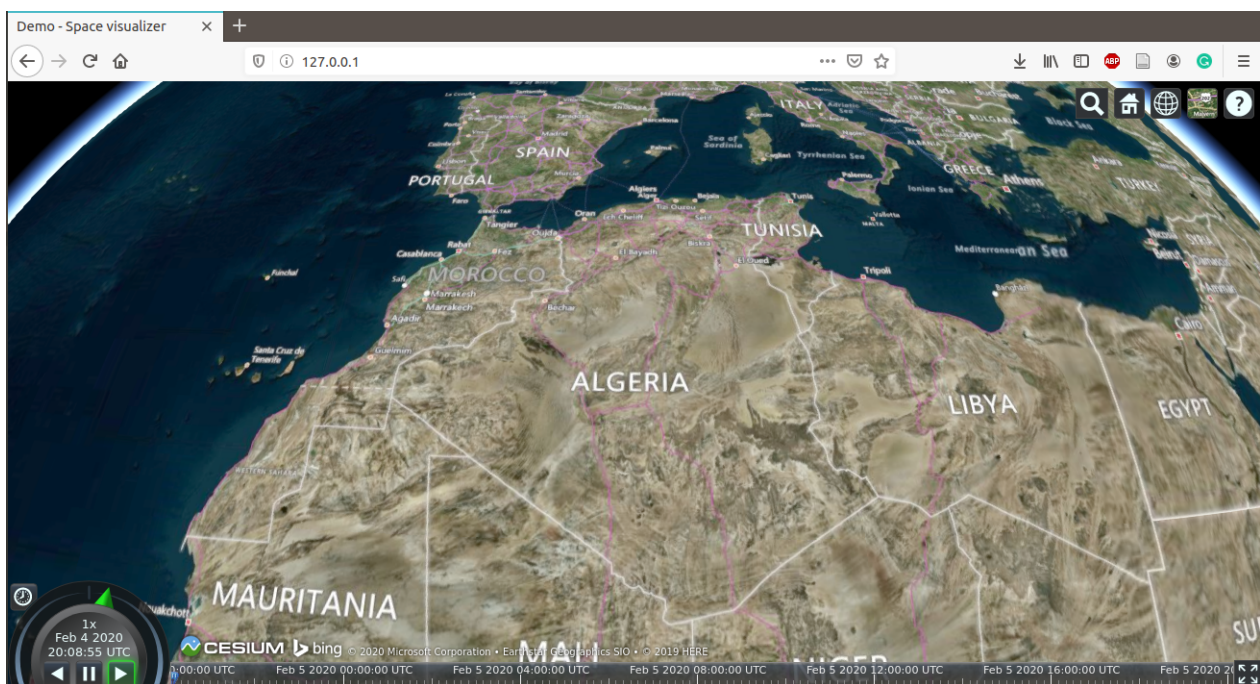


FIGURE 3.7 – La plateforme web déployée dans le hôte virtuel

Conclusion générale

Synthèse du travail

Pour conclure, dans ce projet du stage pratique, nous avons pu réaliser une plateforme web qui sert comme base afin de visualiser les satellites actifs situés au-dessus du sol algérien. On a pu se familiariser avec plusieurs outils informatiques, notamment ceux du développement web, de visualisation géospatiale ainsi que des outils de manipulation des données astronomiques.

En raison des contraintes de temps, je n'ai pas pu tout réaliser pour finir ce projet. Ceci est dû au fait que ces technologies soient toutes nouvelles pour moi, il fallait que je passe une grande partie de la durée déterminée de ce stage (fixée à trois semaines) à découvrir tous ces outils et apprendre à les utiliser.

Perspectives

Dans ce cadre là, on présente les différents points envisageables à améliorer :

- Visualiser les satellites récupérées sur la plateforme et propager leurs orbites.
- Utiliser la paquet Pykep à la place de Skyfield pour la récupération des TLE des satellites ainsi que la propogation de leurs orbites.
- Remplacer les fonction personnalisées par des API pour des calculs plus précis.

Il est à préciser que le projet a été mis en open-source sur Github, toute remarque et contribution serait donc la bienvenue.

Le lien du projet : <https://github.com/tarekbouazza/algerian-space-visualizer/>

ANNEXE A

Code associé au projet

Code Python

Le code ci-dessous est celui qu'on a écrit pour récupérer le nom et l'altitude de chaque Satellites situés au-dessus de l'Algérie.

Importer les paquets nécessaires

```
import os
import re
import urllib
from skyfield import api as sf
from skyfield.api import EarthSatellite, Topos, load
import reverse_geocoder as rg
from datetime import datetime
```

Récupérer les TLEs des Satellites à l'aide de Skyfield

```
datadir = os.environ['HOME']+'/programming/python/astro/skyfield-data/'
loader = sf.Loader(datadir, expire=False)

# Satellite TLEs.
celestrak = 'http://celestrak.com/NORAD/elements/'
geosats = loader.tle(celestrak+'geo.txt')
gpsats = loader.tle(celestrak+'supplemental/gps.txt')
sciencesats = loader.tle(celestrak+'science.txt')
stationsats = loader.tle(celestrak+'stations.txt')
tdrss = loader.tle(celestrak+'tdrss.txt')
visualsats = loader.tle(celestrak+'visual.txt')
```

```
# Make catalogs indexable by either name or catalog number.
satcats = [geosats, gpssats, sciencesats, stationsats, tdrss, visualsats]
sats = {}
catalog = {}
for cat in satcats:
    names = [key for key in cat.keys()]
    for name in names:
        sat = cat[name]
        satnum = sat.model.satnum
        cat[satnum] = sat
        sats[satnum] = sat
        sats[name] = sat
```

Créer une fonction qui permet de récupérer les satellites par nom et par ID

```
def getsat(satid):

    """
    Return a skyfield EarthSatellite.
    <satid> is case independent if it is a satellite name (str).
    Retrieve directly from Celestrak by ID if not in local database.
    """

    if isinstance(satid, str):
        satid = satid.upper()
    if satid in sats.keys():
        return sats[satid]
    if not isinstance(satid, int):
        msg = 'satid must be an integer for satellites not in the local set'
        raise Exception(msg)
    base = 'http://celestrak.com/cgi-bin/TLE.pl?CATNR='
    url = base + str(satid)
    with urllib.request.urlopen(url) as fd:
        lines = fd.readlines()
    for k, line in enumerate(lines):
        if 'PRE' in line.decode():
            name = lines[k+1].decode().strip()
            if name == 'No TLE found':
                msg = '%i is not in the Celestrak database!' % satid
                raise Exception(msg)
            tle1 = lines[k+2].decode().strip()
            tle2 = lines[k+3].decode().strip()
            break
    sat = sf.EarthSatellite(tle1, tle2, name)
    return sat
```

Créer une fonction qui permet d'obtenir la latitude et la longitude des Satellites

```
def getlonlat(subpoint):

    """
    Retrieve each Satellite longitude and latitude from the
    EarthSattelite.subpoint object And convert them to degree decimal
    """

    #LONGITUDE EarthSattelite.Angle object -> Degree Decimal
    lon = subpoint.longitude
    strlongitude= str(lon)
    lonstr = re.split('[deg\']', strlongitude)
    londeg, lonminute, lonseconde = lonstr[0], lonstr[3], lonstr[4]
    longitude=(float(londeg)+ float(lonminute)/60+ float(lonseconde)/(60*60))

    #LATITUDE EarthSattelite.Angle object -> Degree Decimal
    lat = subpoint.latitude
    strlatitude= str(lat)
    latstr = re.split('[deg\']', strlatitude)
    latdeg, latminute, latseconde = latstr[0], latstr[3], latstr[4]
    latitude=(float(latdeg)+ float(latminute)/60 + float(latseconde)/(60*60))
    return longitude, latitude
```

Récupérer et afficher les données des Satellites situés au-dessus de l'Algérie à l'aide de Reverse Geocoder

```
SatList = []
for name in names:
    sat = getsat(name)
    ts = load.timescale()
    t = ts.now()
    geometry = sat.at(t)
    subpoint = geometry.subpoint()
    alt = float(subpoint.elevation.km)
    if alt < 1000:
        longitude, latitude = getlonlat(subpoint)
        coordinates = (longitude, latitude)
        SatLocation = rg.search(coordinates)
        SatLocation[0]
        if SatLocation[0]['cc'] == 'DZ':
            altitude = int(subpoint.elevation.km)
            SatList.append({'Satellite':name, 'Elevation': altitude})

for elem in SatList:
    print('Satellite(Name/ID):',elem['Satellite'], '-', 'Elevation(km):',
          elem['Elevation'])
```

Fichier HTML

Ci-dessous le code HTML5/JavaScript associé la page Web.

CesiumJS a été utilisé pour la visualisation géospatiale.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
7      minimum-scale=1, user-scalable=no">
8    <title>Demo - Space visualizer</title>
9
10   <script src="Cesium/Build/Cesium/Cesium.js"></script>
11
12   <link rel="stylesheet" type="text/css" href="Cesium/Build/Cesium/Widgets/widgets.css">
13   <style>
14     html, body, #cesiumContainer {
15       width: 100%; height: 100%; margin: 0; padding: 0; overflow: hidden;}
16   </style>
17 </head>
18
19 <body>
20   <div id="cesiumContainer"></div>
21
22   <script type="text/javascript">
23     var imagery = Cesium.createDefaultImageryProviderViewModels();
24     var viewer = new Cesium.Viewer('cesiumContainer', {
25       imageryProviderViewModels: imagery,
26       selectedImageryProviderViewModel: imagery[1]
27     });
28     var center = Cesium.Cartesian3.fromDegrees(2, 28);
29     viewer.camera.lookAt(center, new Cesium.Cartesian3(0.0, 0.0, 4200000.0));
30   </script>
31
32 </body>
33 </html>
```

Installation et configuration du serveur Web Apache

Executer les commandes suivantes afin d'installer Apache2

```
$ sudo apt update
$ sudo apt install apache2
```

Vérifier l'installation

```
$ apache2 -version
```

Configurer les paramètres du pare-feu

```
$ sudo ufw app list
$ sudo ufw status
```

Vérifier que le service Apache fonctionne

```
$ sudo systemctl status apache2
```

Configurer le virtual host dans Apache

Un virtual host (hôte virtuel) est utilisé pour gérer les configurations de plusieurs domaines à partir d'un serveur

Configurer le nom du domaine

```
$ sudo mkdir -p /var/www/localdomain.com/html
$ sudo chown -R $USER:$USER /var/www/localdomain.com/html
$ sudo chmod -R 755 /var/www/localdomain.com
```

Créer le fichier localhost pour servir le contenu du serveur

```
$ sudo nano /etc/apache2/sites-available/localdomain.com.conf
```

Entrer les détails de configuration personnalisés suivants pour le nom du domaine :

```
<VirtualHost *:80>
ServerAdmin admin@localdomain.com
ServerName localdomain.com
ServerAlias www.localdomain.com
DocumentRoot /var/www/localdomain.com/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Activer le fichier de configuration de domaine

```
$ sudo a2ensite localdomain.com.conf
$ sudo a2dissite 000-default.conf
$ sudo systemctl restart apache2
```