FLIGHT PROFILE PLAYBACK & ANALYSIS USING CESIUM AND CZML

This internship report submitted in partial fulfillment of the requirement for the award of degree

BACHELOR OF TECHNOLOGY
In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P. SRI CHARAN (21BCE9093)

Under the esteemed guidance of

Shri. Ananth Nath Talla (Scientist – 'F', DLRL)

DEPARTMENT OF SCOPE, COMPUTER SCIENCE AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY – AMARAVATI

(Deemed to be University)

AMARAVATI CAMPUS 522237

2021-2025

DLRL
ISO 9001:2015 Certified Lab

NO: DLRL / HRD / 150742 / 10 / 20 - 20
Government of India, Ministry of Defence
Defence Electronics Research laboratory
Chandrayangutta, Hyderabad- 500005

HUMAN RESOURCE DEVELOPMENT

Dated: 18 | Dec | 2023

CERTIFICATE

This is to certify that **P.SRICHARAN (21BCE9093) of Vellore Institute of Technology - Amaravati, VIT-AP (Deemed to be University),** has undergone project training from 1 Nov, 2023 to 15 Dec, 2023 in the Defence Electronics Research Laboratory, Hyderabad-05. The project **"FLIGHT PROFILE PLAYBACK & ANALYSIS USING CESIUM AND CZML"** is a record of the bonafide work undertaken by him towards partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering**. He has completed the assigned task satisfactorily.

UVVKrishnaveni for

(AnanthNath Talla)	**(UVV Krishnaveni)**	**(JRC Sarma)**
Sc 'F'	**Sc 'G'**	**Sc 'F'**
Guide,	**Additional Director**	**WingHead HRD**
Airborne ESM Technologies	**Dte of Airborne and Naval EW Technologies**	**DLRL,Hyderabad**

DECLARATION

I hereby declare that the work entitled **"FLIGHT PROFILE PLAYBACK & ANALYSIS USING CESIUM AND CZML"** is an original work done in the Department of Computer Science and Engineering, VELLORE INSTITUTE OF TECHNOLOGY – AMARAVATI, VIT-AP (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of **B.Tech in Computer Science and Engineering**. The work has not submitted to any other college or university for the award of any degree or diploma.

P.Sri charan

P SRI CHARAN

DEPARTMENT OF SCOPE, COMPUTER SCIENCE AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY - AMARAVATI

VIT-AP

VIT-AP UNIVERSITY

(DEEMED TO BE UNIVERSITY)

DECLARATION

I hereby declare that the work entitled **"FLIGHT PROFILE PLAYBACK & ANALYSIS USING CESIUM AND CZML"** is an original work done in the Department of Computer Science and Engineering, VELLORE INSTITUTE OF TECHNOLOGY – AMARAVATI, VIT-AP (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of **B.Tech in Computer Science and Engineering**. The work has not submitted to any other college or university for the award of any degree or diploma.

ROLL.NO	NAME	SIGNATURE
21BCE9093	P. SRI CHARAN	P.Sricharan

ACKNOWLEDGEMENT

This is an acknowledgement of the intensive drive and technical competence of many individuals who have contributed to the success of my project.

We are grateful to **Sri N Srinivas Rao, OS & Director, DLRL**, Hyderabad and Sri JRC Sarma, Sc 'F', Wing Head HRD and Members of the HRD for granting us permission for the practical training through development of this project in DLRL.

A Special note of Thanks to **Smt UVV Krishnaveni**, Sc 'G', Additional Director of **Dte of Airborne and Naval Technologies**, DLRL who encouraged us in our work.

I am immensely thankful to Sri **V. Dhananjayulu**, Sc 'F' & Division Head, **Airborne ESM Technologies** Division, DLRL for giving me this opportunity and also providing the facilities at Division.

I am obliged and grateful to our guide **Shri T. AnanthNath,** Sc 'F'& **Airborne ESM Technologies** Division, DLRL and **Shri J. Sreenandan,** Sc 'C' **&Airborne ESM Technologies** Division, DLRL for their valuable suggestions and sagacious guidance in all respects during the course of our training.

I would like to express our gratitude to my Sponsor of DLRL employee, who was very friendly and co-operative.

Our sincere thanks are due to **Dr. CH. Pradeep Reddy** Head of the Dept, Department of CSE of College **VIT-AP**, and the faculty, for the encouragement and guidance provided.

DLRL PROFILE

DEFENCE ELECTRONICS RESEARCH LABORATORY (D.L.R.L) was established in the year 1962 under the aegis of Defence Research and Development Organization (DRDO), Ministry of Defence, to meet the current and future needs of tri services Army, Navy and Air force equipping them with Electronics Warfare Systems.

DLRL has been entrusted with the primary responsibility of the design and development of Electronic Warfare Systems covering both Communication and RADAR Frequency bands.

DLRL consists of large number of dedicated technical and scientific manpower adequately supported by sophisticated hardware and software development facilities. Computers and dedicated Workstations are extensively used for, design and development of sub-systems. Main software required for various types of applications is developed in-house. The quality assurance group is responsible for quality assurance of software developed for Electronic Warfare Systems.

DLRL has number of supporting and technology groups to help the completion of the projects on time and to achieve a quality product. Some of the supporting and technology groups are Printed Circuited Board Group, Antenna Group, Microwave and Millimeter Wave Components Group, Mechanical Engineering Group, LAN, Human Resource Development Group etc. apart from work centers who carryout system design and development activities.

Long-Term self-reliance in Technologies / Systems has been driving principle in its entire development endeavor to make the nation self reliant and independent.

In house Printed Circuit Board facilities provide faster realization of the digital hardware. Multi-layer Printed Circuit Board fabrication facilities are available to cater for a high precision and denser packaging.

The Antenna Group is responsible for design and development of wide variety of antennas covering a broad electromagnetic spectrum (HF to Millimeter Frequencies). The Group also develops RADOMES, which meet stringent environmental conditions for the EW equipment to suit the platform.

The MMW Group is involved in the design and development of MMW Sub-systems and also various Microwave Components like Solid State Amplifier, Switches, Couplers and Filters using the latest state- of-the–art technology.

The Hybrid Microwave Integrated Circuit Group provides custom-made microwave components and super components in the microwave frequency region using both thin film and thick film technology.

In the Mechanical Engineering Group the required hardware for EW Systems is designed and developed and the major tasks involved include Structural and Thermal Engineering.

The Technical Information Center, the place of knowledge bank is well equipped with maintained libraries, books, journals, processing etc. Latest Technologies in the electronic warfare around the globe are catalogued and easily accessible.

The Techniques Division of ECM wing is one such work center where design and development of subsystems required for ECM applications are undertaken. ESM Work Centers design and development of DF Rx, Rx Proc etc. for various ESM Systems using state-art-of-the technology by employing various techniques to suit the system requirements by the end users. All the subsystems are designed and developed using microwave, and processor/DSP based Digital hardware in realizing the real time activities in Electronic Warfare. Most of the work centers are connected through DLRL LAN (Local LAN) for faster information flow and multi point access of information critical to the development activities. Information about TIC, stores and general administration can be downloaded easily.

The Human Resource Division play a vital role in conducting various CEP courses, organizing service and technical seminars to upgrade the knowledge of scientists in the laboratory.

DLRL has been awarded **ISO 9001:2015** certification for Design and Development of Electronic System of assured quality for Defence Services; utilize advanced and cost-effective technologies & systems on time. DLRL shall comply with the requirements of quality Management Systems with a focus on its continual improvement.

ABSTRACT

This project titled **“FLIGHT PROFILE PLAYBACK & ANALYSIS USING CESIUM AND CZML”** delves into the realm of three-dimensional aircraft animation using Cesium, a geospatial platform, and CZML (Cesium Markup Language). The primary focus is on creating dynamic and visually engaging animations of aircraft trajectories, incorporating CZML for precise path definition and orientation control.

In this project, we explore the seamless integration of CZML files, which encode time-dynamic graphical scenes, into the Cesium framework. The CZML files generated using Python serve as a blueprint for animating aircraft movements along defined paths. The project's first phase involves plotting and animating the aircraft along the specified trajectory, showcasing the capabilities of Cesium in geospatial visualization.

The second phase introduces orientation control, enabling the manipulation of the aircraft's heading, roll, and pitch. This enhancement adds a layer of realism to the animation, allowing for a more comprehensive and interactive representation of aircraft motion. Through careful coordination of CZML parameters, the project achieves a lifelike simulation of aircraft maneuvers.

The significance of this project lies in its applications across various domains, including aviation simulations, educational tools, and visualization of flight paths. The integration of Cesium and CZML facilitates the creation of immersive environments for understanding and analyzing aircraft movements.

By combining Cesium's geospatial capabilities with CZML's temporal features, this project not only demonstrates the technical process of 3D aircraft animation but also opens avenues for future developments in virtual environments, flight simulations, and geospatial data representation. The project serves as a foundation for exploring advanced features and extending the scope of 3D aircraft visualization using Cesium and CZML.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 SOFTWARE ENVIRONMENT

The software environment refers to the underlying infrastructure and tools that support the development and execution of a particular software application. In the context of 3D aircraft animation in Cesium and CZML (Cesium Language), the software environment involves the technologies and frameworks necessary to create and visualize three-dimensional animations of aircraft.

Cesium is a geospatial platform for creating applications with 3D mapping and visualization capabilities. It is often used for building web-based applications that require geospatial data and interactive maps. Cesium supports WebGL for rendering high-performance 3D graphics in a web browser.

1.2 OVERVIEW OF 3d AIRCRAFT ANIMATION IN CESIUM AND CZML

3D Aircraft Animation: This refers to the process of creating dynamic and realistic animations of aircraft in a three-dimensional space. In the context of Cesium, these animations can be embedded within web applications to provide users with an immersive and interactive experience.

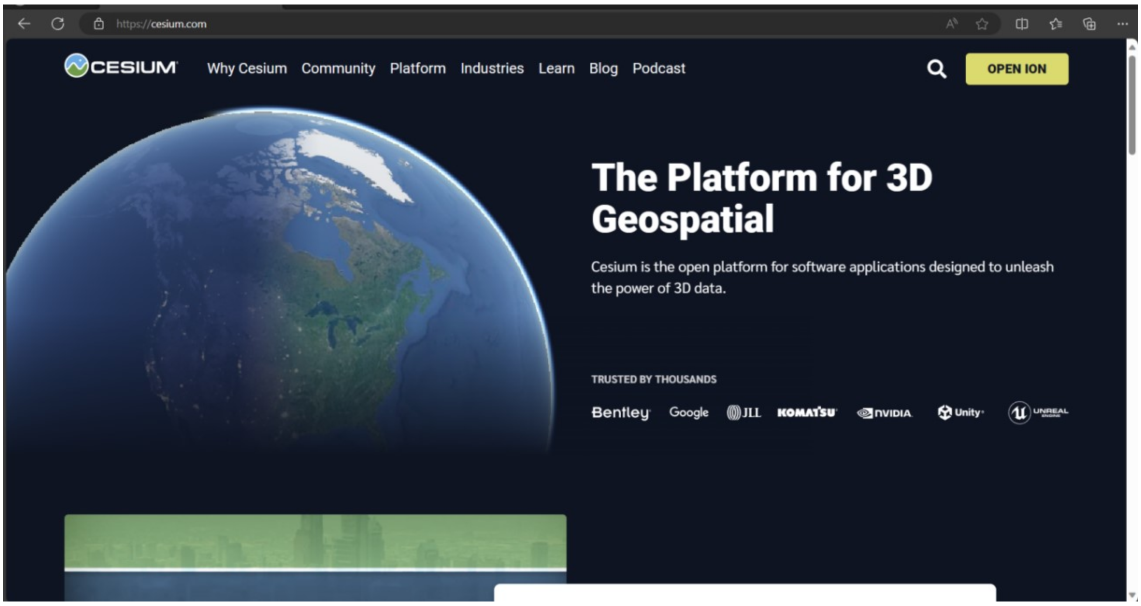
Cesium: Cesium is an open-source JavaScript library that enables the creation of 3D globes and maps in a web browser. It leverages WebGL for hardware-accelerated graphics, allowing for the rendering of high-quality, interactive 3D content. Cesium is commonly used for applications such as geospatial visualization, simulations, and virtual globes.

CZML (Cesium Language): CZML is a declarative language designed for describing time-dynamic graphical scenes in Cesium. It allows developers to define the properties and behaviors of 3D objects, including aircraft, over time. CZML is often used to script animations, movements, and other dynamic aspects of the 3D environment.

2. CESIUM AND CZML

2.1 INTRODUCTION TO CESIUM

Cesium is an open-source JavaScript library designed for creating3D globes and maps in web browsers. It is built on top of WebGL, a web standard for rendering 3D graphics, enabling high-performance visualization of geospatial data. Cesium provides a platform for developing interactive and dynamic applications that involve geographical and spatial information. Key features of Cesium include:



Source :: [Cesium: The Platform for 3D Geospatial]

3D Visualization: Cesium allows developers to display and interact with three-dimensional models of the Earth's surface. This is particularly useful for applications that involve terrain, geographical features, and dynamic objects.

Geospatial Data Integration: Cesium supports the integration of various geospatial data sources, including satellite imagery, terrain data, and vector data. It can visualize data in multiple formats and projections.

WebGL Rendering: Leveraging the power of WebGL, Cesium delivers smooth and hardware-accelerated rendering of 3D graphics directly within web browsers. This eliminates the need for plugins or external applications.

Time-Dynamic Content: Cesium supports the visualization of time-varying data, allowing developers to create animations and simulations that change over time.
This is essential for applications that involve dynamic events or movements.

Cross-Platform Compatibility: Cesium is designed to work across different browsers and devices, ensuring a consistent and accessible experience for user.

2.2 UNDERSTANDING CZML (Cesium Mapping Language)

CZML (Cesium Markup Language) is a declarative language developed by the Cesium team specifically for describing time-dynamic scenes in Cesium applications. It serves as a powerful tool for scripting the behavior of 3D objects and events over time. Key aspects of CZML include:

Declarative Syntax: CZML uses a JSON-like syntax to declare the properties and behavior of objects in the 3D scene. This includes defining positions, orientations, appearances, and other attributes.

Time-Dynamic Nature: CZML is inherently designed to handle time-varying data. Developers can specify how the properties of objects change over time, enabling the creation of animated and interactive scenes.

Extensibility: CZML is extensible, allowing developers to define custom properties and behaviors beyond the core set of features. This flexibility makes it suitable for a wide range of applications.

Hierarchy and Relationships: CZML supports the creation of hierarchical structures and relationships between objects. This is useful for representing complex scenes with multiple interacting elements.

Interoperability: CZML is designed to work seamlessly with Cesium. It can be used to define the properties of 3D models, cameras, sensors, and other entities within the Cesium environment.

By understanding and utilizing Cesium along with CZML, developers can create rich and dynamic geospatial applications that involve interactive 3D visualization and time-dynamic content. This combination is particularly valuable for applications such as mapping, simulation, and virtual globes.

3. WORK CARRIED OUT

3.1 GENERATING CZML FILE FOR AIRCRAFTUSING NAV DATA

In the context of Cesium and CZML, generating CZML files for aircraft paths involves creating structured data files that define the trajectory and movement of aircraft over time. These files typically contain information about the position, orientation, and other dynamic properties of the aircraft at specific points in time. Here's a brief overview of the process:

Position Data: CZML files include data points that specify the position of the aircraft in three-dimensional space at different timestamps. These positions are often defined using longitude, latitude, and altitude coordinates.

Time Information: Each data point in the CZML file is associated with a specific timestamp. This time information allows for the animation of the aircraft's movement over a defined period.

Additional Properties: Apart from position and time, CZML files can include additional properties such as color, scale, and other attributes that may change over time. These properties contribute to the overall animation and visualization of the aircraft.

Tool Support: Developers may use tools or scripts to generate CZML files efficiently. These tools can convert raw data or waypoints into the CZML format, ensuring proper formatting and adherence to the Cesium specification.

NAV FILE WITH POSITION ATTRIBUTE:


```
[
  {
    "id":"document",
    "version":"1.0"
  },
  {
    "id":"Vehicle",
    "availability":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
    "label":{
      "fillColor":[
        {
          "interval":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
          "rgba":[
            255,255,0,255
          ]
        }
      ],
      "font":"bold 10pt Segoe UI Semibold",
      "horizontalOrigin":"CENTER",
      "outlineColor":{
        "rgba":[
          0,0,0,255
        ]
      },
      "pixelOffset":{
        "cartesian2":[
          0.0,20.0
        ]
      },
      "scale":1.0,
      "show":[
        {
          "interval":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
          "boolean":true
        }
      ],
      "style":"FILL",
      "text":"Test Vehicle",
      "verticalOrigin":"CENTER"
    },
```

```
    "model":{
      "gltf":"models/CesiumAir/Cesium_Air.glb",
      "minimumPixelSize":100,
      "maximumScale":50
    },
    "orientation" : {
      "velocityReference": "#position"
    },
    "viewFrom": {
      "cartesian": [ -2080, -1715, 779 ]
    },
    "properties" : {
        "fuel_remaining" : {
            "epoch":"2222-10-13T13:45:49Z",
            "number": [
                0, 22.5,
                1500, 21.2
            ]
        }
    },
    "path":{
      "material":{
          "solidColor":{
            "color":{
                "interval":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
                "rgba":[
                  255,255,0,255
                ]
              }
            }
      },
      "width":[
        {
          "interval":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
          "number":5.0
        }
      ],
      "show":[
        {
          "interval":"2222-10-13T13:45:49Z/2222-10-13T15:12:13Z",
          "boolean":true
        }
      ]
        },
    "position":{
      "interpolationAlgorithm":"LAGRANGE",
      "interpolationDegree":1,
      "epoch":"2222-10-13T13:45:49Z",
      "cartographicDegrees": [

      ]
    }
  }
]
```

PYTHON CODE TO GENERATE CZML FILE:

```
import pandas as pd
import numpy as np
from pathlib import Path
import math
import os
import re

import warnings
warnings.filterwarnings('ignore')

# Function to replace date values in the JSON text
def replace_values(json_text, new_date_before,new_start_time):
    # Regex pattern to match date in yyyy-mm-dd format
    pattern = r"(?P<before>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z)\/(?P<after>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z)"
    # Replace found dates with the new date provided
    matches=re.finditer(pattern,json_text)
    for match in matches:
        before=match.group("before")
    modified_text = json_text.replace(before,new_date_before+"T"+new_start_time+"Z")
    return modified_text

# Read content from the .czml file
file_path = 'M.txt'  # Update with your file path
with open(file_path, 'r') as file:
    file_content = file.read()

# Get a date from date column
data = pd.read_csv('ref_data.csv')
date = data.iloc[0,0]
year = date[-4:]
month = date[:2]
day = date[3:5]

# Getting start time from csv
start_time = data.iloc[0,1].strip()
new_start_time = start_time[:-4]
print(new_start_time)
```

```
# Getting end time from csv
end_time = data.iloc[-1,1].strip()
new_end_time = end_time[:-4]
print(new_end_time)

new_date_before = year +"-" + month + "-" + day
new_date_after = year +"-" + month + "-" + day

print(new_date_before)

# Replace the dates in the content with the new date
updated_content = replace_values(file_content, new_date_before,new_start_time)

# Write the updated content back to the .czml file
with open(file_path, 'w') as file:
    file.write(updated_content)

# Function to replace date values in the JSON text
def replace_values(json_text,new_date_after,new_end_time):
    # Regex pattern to match date in yyyy-mm-dd format
    pattern = r"(?P<before>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z)\/(?P<after>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z)"
    # Replace found dates with the new date provided
    matches=re.finditer(pattern,json_text)
    for match in matches:
        after=match.group("after")

    modified_text = json_text.replace(after,new_date_after+"T"+new_end_time+"Z")

    return modified_text

# Read content from the .czml file
file_path = 'M.txt'  # Update with your file path
with open(file_path, 'r') as file:
    file_content = file.read()


# Replace the dates in the content with the new date
updated_content = replace_values(file_content,new_date_after,new_end_time)
```

```
# Write the updated content back to the .czml file
with open(file_path, 'w') as file:
    file.write(updated_content)

# Read data from CSV ad pick the required columns
df=pd.read_csv('ref_data.csv',usecols = ['Time',' Latitude (deg)',' Longitude (deg)',' Baro Altitude (m)'])


#Process the time
df['Time_sec']=0
df['Time_diff']=0
for i in range (0,len(df)):
    t=df.Time[i].split(':')
    #print(t)
    time=float(t[0])*3600+float(t[1])*60+float(t[2])
    #print(time)
    df.Time_sec[i]=time
    if (i==0):
        df.Time_diff[i]=0
    else:
        df.Time_diff[i]=df.Time_sec[i]-df.Time_sec[0]
        df.Time_diff[i]=round(df.Time_diff[i],3)

df.to_csv('Text_file.csv', columns=['Time_diff', ' Latitude (deg)',' Longitude (deg)',' Baro Altitude (m)'],header=False, index=False)


#add comma at the line end
def add(csv_file_path):
  with open(csv_file_path, "r") as f:
    lines = f.readlines()
  lines[-1] = lines[-1].rstrip("\n")
  for i in range(len(lines) - 1):
    values = lines[i].split()
    values[-1] += ","
    lines[i] = " ".join(values) + "\n"
  with open(csv_file_path, "w", newline="") as f:
    f.writelines(lines)
csv_file_path = "Text_file.csv"
add(csv_file_path)
```

```
##
#searching and placing the processed values in textfile

k=0

with open('Text_file.csv', mode='r', encoding= 'utf-8') as file:
    data1 = file.readlines()
    #print(data1[:3])

#os.remove('cs.txt')


with open('M.txt', mode='r', encoding= 'utf-8') as file:
    data = file.readlines()
    #print(data[:5])
    for i in range (0,len(data)):
        if (data[i].find('cartographicDegrees') != (-1)):
            break
for k in range (0,len(data1)):
    data[i]= data[i]+ data1[k] #'*************************\n'

with open('M2.txt', mode='w', encoding= 'utf-8') as file:
    file.writelines(data)

os.rename ('M2.txt', 'M2.czml')

#"cartographicDegrees": [
print('end')
```

HTML FILE:

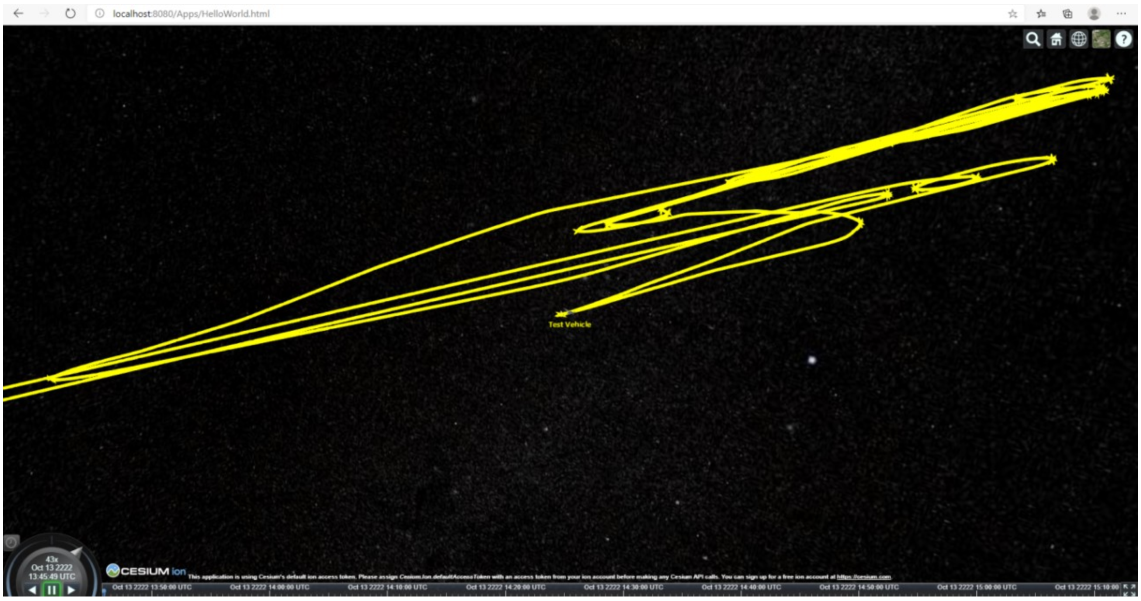
```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Use correct character set. -->
  <meta charset="utf-8">
  <!-- Tell IE to use the latest, best version. -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- Make the application on mobile take up the full browser screen and disable user scaling. -->
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no">
  <title>Hello World!</title>
  <script src="../Build/Cesium/Cesium.js"></script>
  <style>
      @import url(../Build/Cesium/Widgets/widgets.css);
      html, body, #cesiumContainer {
          width: 100%; height: 100%; margin: 0; padding: 0; overflow: hidden;
      }
  </style>
</head>
<body>
  <div id="cesiumContainer"></div>
  <script>
    var viewer = new Cesium.Viewer('cesiumContainer', {
                infoBox : false,
                selectionIndicator : false,
                shadows : true,
                shouldAnimate : true
        });
        var pinBuilder = new Cesium.PinBuilder();
        var url = Cesium.buildModuleUrl("Assets/Textures/maki/airport.png");
        var groceryPin = Cesium.when(
          pinBuilder.fromUrl(url, Cesium.Color.GREEN, 48),
          function (canvas) {
                return viewer.entities.add({
                  name: "Grocery store",
                  position: Cesium.Cartesian3.fromDegrees(77.6619303,12.9501196),
                  billboard: {
                        image: canvas.toDataURL(),
                        verticalOrigin: Cesium.VerticalOrigin.BOTTOM,
                  },
                });
          }
        );
        var url1 = Cesium.buildModuleUrl("Assets/Textures/maki/airport.png");
        var groceryPin = Cesium.when(
          pinBuilder.fromUrl(url1, Cesium.Color.GREEN, 48),
          function (canvas) {
                return viewer.entities.add({
                  name: "Grocery store",
                  position: Cesium.Cartesian3.fromDegrees(77.705332,13.198914),
                  billboard: {
                        image: canvas.toDataURL(),
                        verticalOrigin: Cesium.VerticalOrigin.BOTTOM,
                  },
                });
          }
        );
        var questionPin = viewer.entities.add({
          name: "Chimney Hills",
          position: Cesium.Cartesian3.fromDegrees(77.4910, 13.0731),
          billboard: {
                image: pinBuilder.fromText("C", Cesium.Color.BLACK, 48).toDataURL(),
                verticalOrigin: Cesium.VerticalOrigin.BOTTOM,
          },
        });
        var questionPin = viewer.entities.add({
          name: "AkashNG",
          position: Cesium.Cartesian3.fromDegrees(77.55, 13.052),
          billboard: {
                image: pinBuilder.fromText("Akash", Cesium.Color.BLACK, 48).toDataURL(),
                verticalOrigin: Cesium.VerticalOrigin.BOTTOM,
          },
        });
        // Now move the Aircraft
        var statusDisplay = document.createElement("div");
        var fuelDisplay = document.createElement("div");
        var czmlPath = "../Apps/SampleData/";
        var vehicleEntity;
        // Add a blank CzmlDataSource to hold our multi-part entity/entities.
        var dataSource = new Cesium.CzmlDataSource();
        viewer.dataSources.add(dataSource);
```

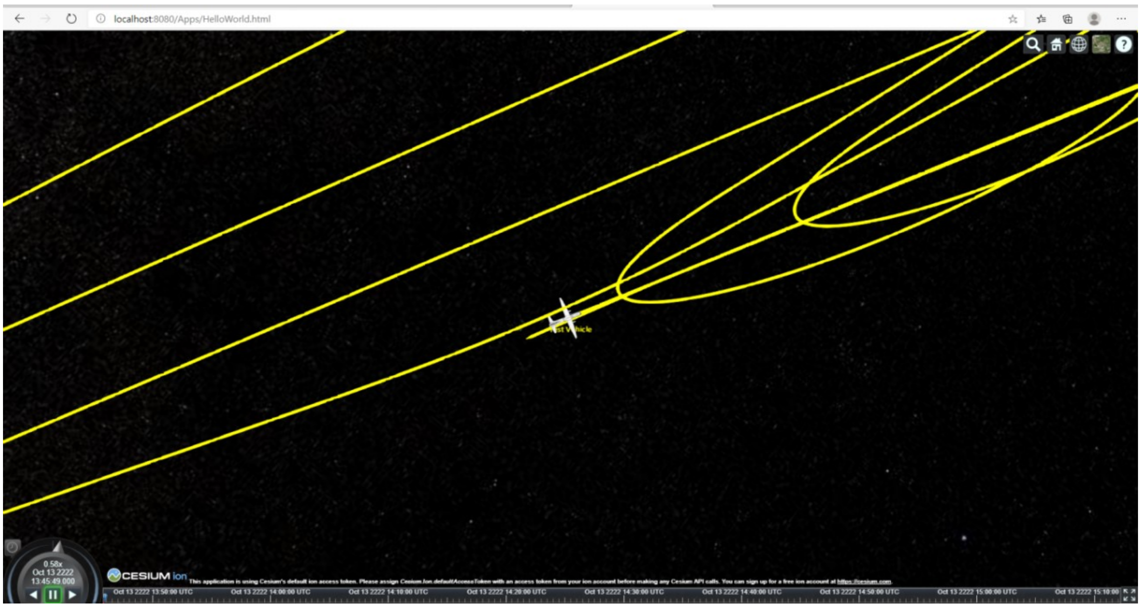
```
            // This demo shows how a single path can be broken up into several CZML streams.
      var partsToLoad = [
        {
            url: "../Apps/SampleData/MultipartVehicle_part1.czml",
            range: [0, 5750.574],
            requested: false,
            loaded: false,
        }
      ];
      function updateStatusDisplay() {
        var msg = "";
        partsToLoad.forEach(function (part) {
            msg += part.url + " - ";
            if (part.loaded) {
              msg += "Loaded.<br/>";
            } else if (part.requested) {
              msg += "Loading now...<br/>";
            } else {
              msg += "Not needed yet.<br/>";
            }
        });
        statusDisplay.innerHTML = msg;
      }
      // Helper function to mark a part as requested, and process it into the dataSource.
      function processPart(part) {
        part.requested = true;
        updateStatusDisplay();
        dataSource.process(czmlPath + part.url).then(function () {
            part.loaded = true;
            updateStatusDisplay();
            // Follow the vehicle with the camera.
            if (!viewer.trackedEntity) {
              viewer.trackedEntity = vehicleEntity = dataSource.entities.getById(
                    "Vehicle"
              );
            }
        });
      }
      // Load the first part up front.
      processPart(partsToLoad[0]);
```

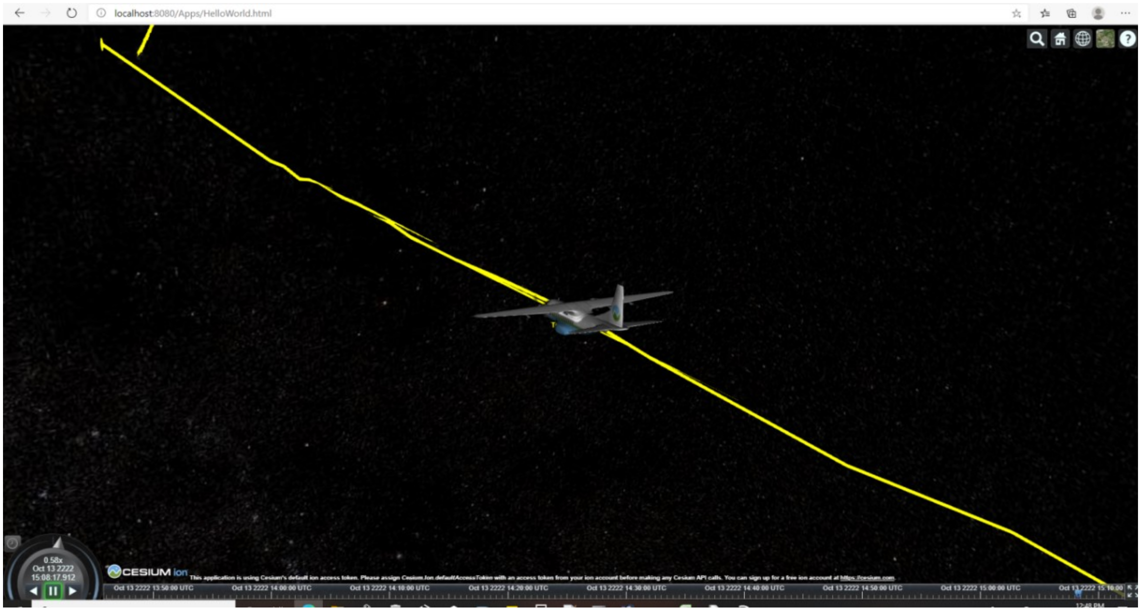
```
  </script>

</body>
</html>
```

WITH THE CZML OUTPUT OF THIS CODE WE INCORPORATE THAT FILE IN INDEX.HTML FILE

LET'S SEE THE OUTPUT:







3.2 INCORPORATING ORIENTATION ATTRIBUTE IN NAV FILE

Orientation control in CZML involves specifying the orientation or rotation of the aircraft at different points in time. This is crucial for accurately representing the heading, pitch, and roll of the aircraft as it moves through the 3D environment. Here's a brief explanation:

Orientation Data: CZML allows developers to include orientation information for each data point. This information describes how the aircraft is oriented in terms of heading, pitch, and roll angles.

Quaternion Representation: Orientations in CZML are often represented using quaternions, a mathematical concept that encapsulates rotational information. Quaternions provide a compact and efficient way to represent 3D rotations.

Smooth Transitions: Developers can define smooth transitions between different orientations to create realistic animations. This is important for ensuring that the aircraft's movement appears natural and visually pleasing.

Integration with Other Properties: Orientation control is often integrated with other properties in the CZML file, such as position and time. This allows for a comprehensive representation of the aircraft's behavior over the animation timeline.

Testing and Fine-Tuning: Incorporating orientation control may involve testing and fine-tuning to achieve the desired visual result. Developers can adjust orientation parameters to enhance the realism and accuracy of the animation.

IN THE TEXT FILE WE WILL ADD ORIENTATION ATTRIBUTE:

```
"orientation": {
            "epoch": "2222-10-13T13:45:49Z",
            "interpolationAlgorithm": "LAGRANGE",
            "interpolationDegree": 1,
            "unitQuaternion": [
      ]
        },
```

3.3 CONVERSION FROM EULER TO QUATERNION

In he context of 3D aircraft animation using Cesium and CZML, understanding the conversion from Euler angles to quaternions is essential for accurately representing the orientation of the aircraft. The transition from Euler angles (commonly representing heading, pitch, and roll) to quaternions ensures a mathematically robust and efficient representation of 3D rotations.

Euler to Quaternion Conversion Formula:

The conversion from Euler angles (Head,Pitch,Roll) to a quaternion (qx, qy, qz, qw) can be expressed using the following formula:

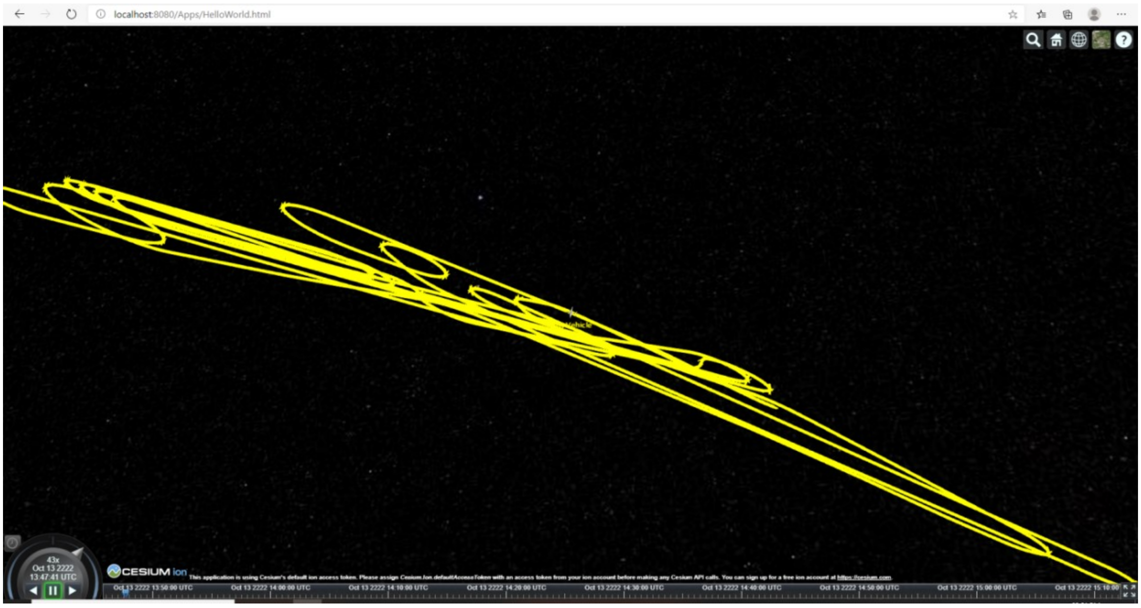
qx, qy, qz (Vector Part): These components represent the imaginary part of the quaternion and determine the axis of rotation. The vector (qx, qy, qz) points along the axis of rotation, and its magnitude is proportional to the sine of half the rotation angle.

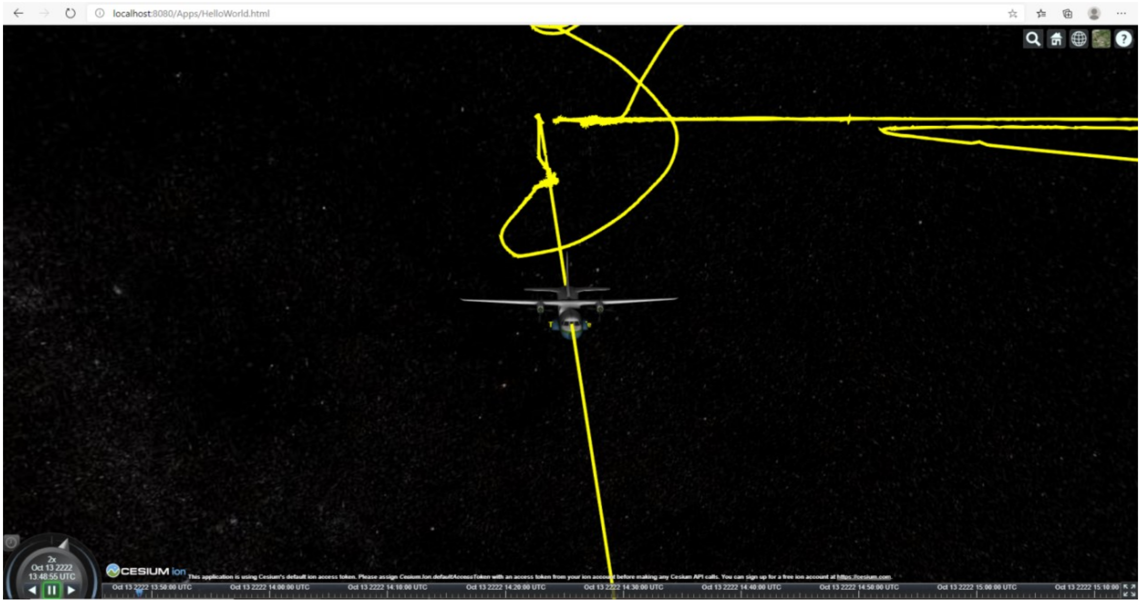
qw (Scalar Part): This component represents the real part of the quaternion and determines the amount of rotation. The scalar qw is proportional to the cosine of half the rotation angle.

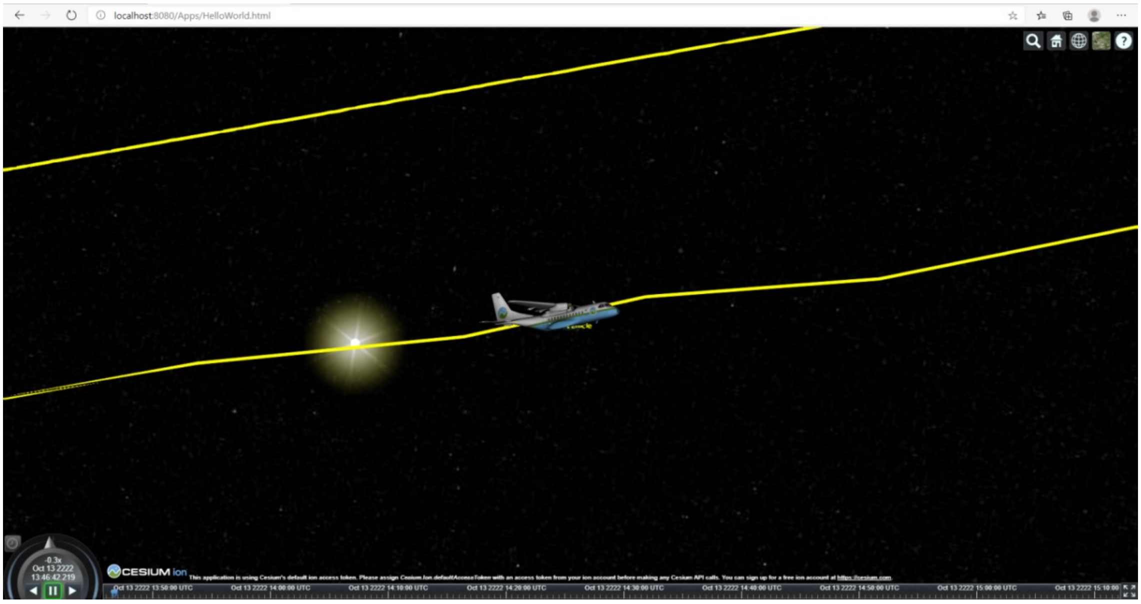
```
qx=sin(Head/2)*cos(Pitch/2)*cos(Roll/2)−cos(Head/2)*sin(Pitch/2)*sin(Roll/2)

qy=cos(Head/2)*sin(Pitch/2)*cos(Roll/2)+sin(Head/2)*cos(Pitch/2)*sin(Roll/2)

qz=cos(Head/2)*cos(Pitch/2)*sin(Roll/2)−sin(Head/2)*sin(Pitch/2)*cos(Roll/2)

qw=cos(Head/2)*cos(Pitch/2)*cos(Roll/2)+sin(Head/2)*sin(Pitch/2)*sin(Roll/2)
```

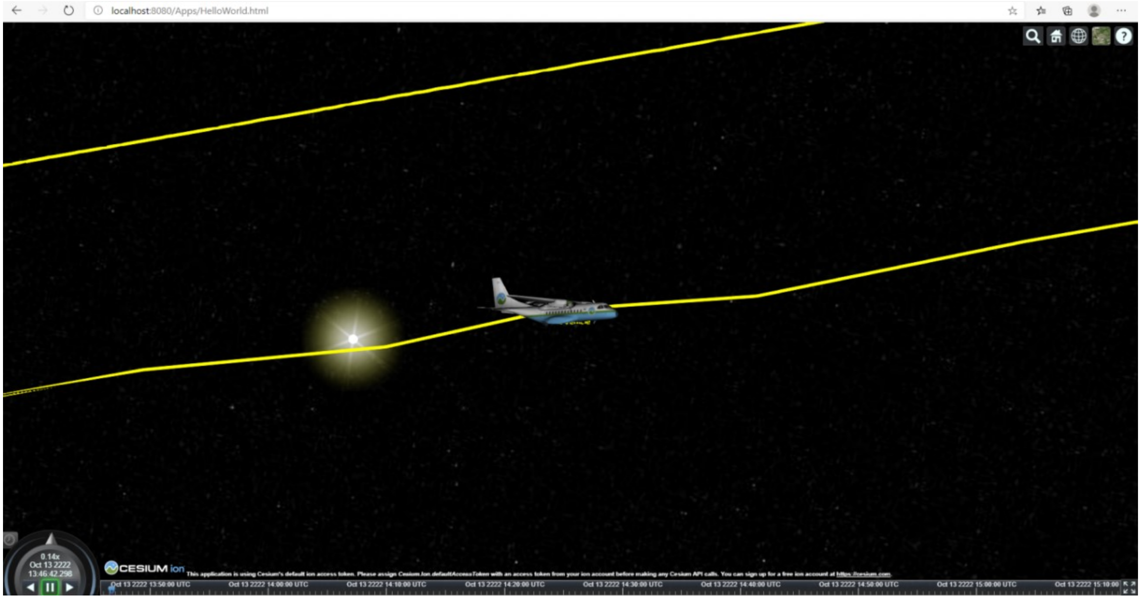
The conversion formulas can be derived using principles from quaternion mathematics and trigonometry. The general formula for converting Euler angles to quaternions is derived from the rotation matrix representation of the Euler angles and then converted to quaternion form.

3.4 INCLUDING CZML CODE IN HTML FILE









By generating CZML files for aircraft paths and incorporating orientation control, developers can create engaging and realistic 3D animations of aircraft movements within the Cesium environment. These animations can be part of applications such as flight simulations, geospatial visualizations, or any scenario that involves dynamic aircraft behavior.

4. RESULTS AND VISUALIZATION

4.1 SHOWCASING 3D AIRCRAFT ANIMATION

The results of 3D aircraft animation using Cesium and CZML are visually compelling and offer a realistic representation of aircraft movements. The animation showcases the dynamic path of the aircraft over time, allowing users to visually track its trajectory in a three-dimensional space. This immersive experience is achieved through the integration of geospatial data, 3D models, and CZML scripts.

Key Visual Elements:

Smooth Trajectory: The aircraft follows a smooth and continuous trajectory, adhering to the defined path in the CZML file. Smooth transitions between positions and orientations contribute to a lifelike representation.

Realistic Rendering: The 3D model of the aircraft is realistically rendered, considering factors such as lighting, shading, and scale. This attention to detail enhances the overall visual appeal of the animation.

Time Dynamics: The animation incorporates time dynamics, allowing users to observe how the aircraft's position, orientation, and other properties change over the specified time period. This is particularly useful for scenarios involving dynamic events or simulations.

LET'S SEE THE AIRCRAFT AT A POINT:

LATITUDE: 13.821834

LONGITUDE: 78.871380

ALTITUDE: 6084

HEAD: 0

PITCH: 0

ROLL: 60

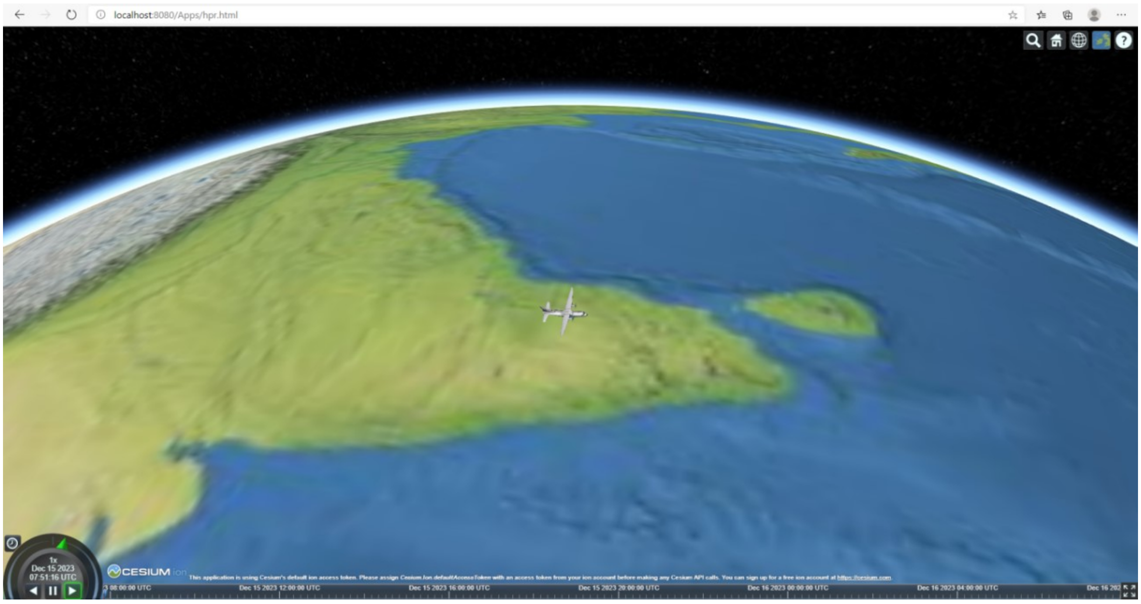
LATTITUDE: 13.957322

LONGITUDE: 78.771121

ALTITUDE: 2078

HEAD: 95.6PITCH: 10.7

ROLL: 25.5



4.2 REAL-TIME VISUALIZATION AND INTERACTIVE FEATURES:

In addition to showcasing the animation, the application provides real-time visualization and interactive features that enhance the user experience. Users can actively engage with the 3D environment, explore details, and customize their viewing experience.

Interactive Features:

Zoom and Pan: Users can zoom in and out of the scene to get a closer look at the aircraft or pan across the virtual landscape. These interactive controls provide flexibility in exploring different perspectives.

Time Slider: A time slider allows users to interactively control the playback of the animation. They can pause, rewind, or fast-forward to specific points in time, gaining better insights into the aircraft's behavior.

Information Pop-ups: Interactive pop-ups provide additional information about the aircraft, such as its current position, altitude, and other relevant details. This enhances the educational or informative aspects of the application.

Real-time Updates:

Dynamic Data Feeds: For applications requiring real-time data, the animation can be dynamically updated based on live feeds. This feature is valuable for scenarios where tracking actual aircraft movements is essential.

Responsive Controls: The application's controls are responsive, providing a seamless experience across different devices. Whether accessed on a desktop or a mobile device, users can navigate and interact with the 3D scene effortlessly.

5. REAL-WORLD UTILIZATION SCENARIOS

Flight Position Analysis: The 3D animation provides a comprehensive visual representation of the aircraft's position during flight trials. Researchers and engineers can analyze the trajectory, altitude, and geographic location of the aircraft at specific time intervals.

Orientation Assessment: Detailed orientation control in the animation allows for a nuanced analysis of the aircraft's heading, roll, and pitch.

Engineers and aviation experts can precisely study how the aircraft maneuvers and responds during different phases of flight.

Real-Time Monitoring: The ability to visualize flight trials in real-time facilitates on-the-fly monitoring of the aircraft's movement. This is particularly beneficial for immediate decision-making during trials, ensuring safety and effective data collection.

Data Validation and Calibration: The animated representation serves as a visual aid for validating and calibrating data obtained from onboard sensors and instruments.

Any discrepancies or anomalies in the flight data can be quickly identified and addressed.

Training and Simulation: The 3D animation can be utilized for training purposes, allowing pilots and ground crews to simulate and understand aircraft behavior under different conditions. This enhances the training process for new pilots and contributes to the continuous improvement of aviation personnel.

6. CONCLUSION AND FUTURE WORK

In conclusion, the integration of Cesium and CZML for 3D aircraft animation has proven to be a powerful and versatile solution. The combination of Cesium's robust geospatial visualization capabilities and CZML's declarative language for time-dynamic scenes enables the creation of immersive and realistic animations. The showcase of 3D aircraft movements, along with real-time visualization and interactive features, enhances the user experience and opens up various possibilities for applications in simulation, education, and geospatial exploration.

The smooth trajectory, realistic rendering, and time dynamics incorporated into the animation contribute to a visually compelling representation of aircraft behavior. The real-time visualization features, including interactive controls and dynamic data feeds, add an extra layer of engagement and utility to the application. This combination of features makes the solution valuable across different domains, from aviation training simulations to interactive educational tools.

As technology continues to evolve, there are several avenues for future work and enhancements in the realm of 3D aircraft animation using Cesium and CZML:

Enhanced Realism: Future developments could focus on improving the realism of aircraft models, environmental effects, and lighting to create an even more immersive experience.

Expanded Interactivity: Further expand interactive features, potentially incorporating augmented reality (AR) or virtual reality (VR) for an even more immersive and hands-on user experience.

Advanced CZML Features: Explore and implement advanced features within CZML to handle more complex scenarios, such as multiple interacting aircraft, dynamic weather effects, or collaborative simulations.

Integration with External Data Sources: Extend the application's capabilities by integrating with external data sources, such as real-world air traffic data, to simulate and visualize actual aircraft movements in real-time.