from flask import Flask, jsonify

import random

import cesium

app = Flask(\_\_name\_\_)

Define the radar sensor

radar\_sensor = {

'name': 'Radar Sensor',

'position': [0, 0, 0],

'orientation': [0, 0, 0],

'sensible\_element': {

'type': 'antenna',

'frequency\_range': [2.4, 5.8, 10.2] # GHz

}

}

Define the detection ranges for each frequency

detection\_ranges = [

{'frequency': 2.4, 'range': [0, 1000]}, # meters

{'frequency': 5.8, 'range': [0, 500]}, # meters

{'frequency': 10.2, 'range': [0, 2000]} # meters

]

Define a function to generate random detection data

def generate\_detection\_data():

detection\_data = []

for detection\_range in detection\_ranges:

frequency = detection\_range['frequency']

range\_min, range\_max = detection\_range['range']

range\_value = random.uniform(range\_min, range\_max)

detection\_data.append({

'frequency': frequency,

'range': range\_value

})

return detection\_data

Define a route to return the radar sensor data

@app.route('/radar-sensor', methods=['GET'])

def get\_radar\_sensor\_data():

detection\_data = generate\_detection\_data()

return jsonify({

'radar\_sensor': radar\_sensor,

'detection\_data': detection\_data

})

Define a route to return the CesiumJS scene

@app.route('/cesium-scene', methods=['GET'])

def get\_cesium\_scene():

cesium\_scene = {

'name': 'Radar Sensor Scene',

'entities': [

{

'name': 'Radar Sensor',

'position': radar\_sensor['position'],

'orientation': radar\_sensor['orientation'],

'geometry': {

'type': 'Sphere',

'radius': 10

}

}

],

'animations': []

}

return jsonify(cesium\_scene)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

To visualize the radar sensor data in CesiumJS, you can use the following JavaScript code:

```

// Import CesiumJS

const Cesium = require('cesium');

// Create a Cesium Viewer

const viewer = new Cesium.Viewer('cesiumContainer');

// Load the radar sensor data from the Flask backend

fetch('/radar-sensor')

.then(response => response.json())

.then(data => {

const radarSensor = data.radar\_sensor;

const detectionData = data.detection\_data;

// Create a Cesium entity for the radar sensor

const radarSensorEntity = viewer.entities.add({

name: radarSensor.name,

position: Cesium.Cartesian3.fromDegrees(radarSensor.position[0], radarSensor.position[1], radarSensor.position[2]),

orientation: Cesium.HeadingPitchRoll.fromDegrees(radarSensor.orientation[0], radarSensor.orientation[1], radarSensor.orientation[2]),

geometry: new Cesium.SphereGeometry({

radius: 10

})

});

// Create Cesium entities for the detection ranges

detectionData.forEach(detectionRange => {

const frequency = detectionRange.frequency;

const range = detectionRange.range;

const detectionRangeEntity = viewer.entities.add({

name: `Detection Range ${frequency} GHz`,

position: Cesium.Cartesian3.fromDegrees(radarSensor.position[0], radarSensor.position[1], radarSensor.position[2]),

orientation: Cesium.HeadingPitchRoll.fromDegrees(radarSensor.orientation[0], radarSensor.orientation[1], radarSensor.orientation[2]),

geometry: new Cesium.SphereGeometry({

radius: range

})

});

});

});

```