import React, { useState, useCallback } from 'react';

import { Canvas, useThree } from '@react-three/fiber';

import { OrbitControls, useGLTF } from '@react-three/drei';

import \* as THREE from 'three';  // Import THREE for vector operations

// Coordinates mapping for buildings

const buildingCoordinates = {

  'B8': [-10, 20, 45],

  'B12': [30, 20, 100],

  'B9': [30, 20, 40],

  'B11': [0, 35, 80],

  'B19': [-55, 25, 110],

  'B16': [-100, 25, 65],

  'B20': [-140, 25, 80],

  'B22': [-180, 25, 110],

  'B21': [-165, 25, 135],

  'B23': [-110, 25, 115],

  'B13': [-190, 25, 170],

  'PINEMESS': [-110, 35, 150],

  'B14': [-140, 25, 170],

  'B15': [-60, 25, 150],

  'B18': [-20, 25, 135],

  'B10': [65, 25, 75],

  'B17': [105, 25, 25],

  'B24': [150, 25, 5],

  'B26': [170, 30, 35],

  'B25': [130, 30, 60],

  'A19': [165, 30, 95],

  'A17': [105, 40, 110],

  'A18': [105, 50, 140],

  'A13': [-15, 35, 166],

  'AVL(GROUND-F\_A13)': [-15, 35, 166],

  'MNC-LAB(GROUND-F\_A13)': [-15, 35, 166],

  'A13-1A(GROUND-F\_A13)': [-15, 35, 166],

  'A13-2A(1ST-F\_A13)': [-15, 35, 166],

  'A13-2B(1ST-F\_A13)': [-15, 35, 166],

  'A13-2C(1ST-F\_A13)': [-15, 35, 166],

  'A13-2D(1ST-F\_A13)': [-15, 35, 166],

  'A13-3A(2ND-F\_A13)': [-15, 35, 166],

  'A13-L1(2ND-F\_A13)': [-15, 35, 166],

  'A13-L2(2ND-F\_A13)': [-30, 35, 206],

  'NKN\_CONFERENCE\_ROOM(3RD-F\_A13)': [-30, 35, 206],

  'A13-F1(3RD-F\_A13)': [-30, 35, 206],

  'A13-F2(3RD-F\_A13)': [-30, 35, 206],

  'A13-F7(3RD-F\_A13)': [-30, 35, 206],

  'A13-F4(3RD-F\_A13)': [-30, 35, 206],

  'A13-F3(3RD-F\_A13)': [-30, 35, 206],

  'A13-F5(3RD-F\_A13)': [-30, 35, 206],

  'A13-F6(3RD-F\_A13)': [-30, 35, 206],

  'A13-F9(3RD-F\_A13)': [-30, 35, 206],

  'A13-F11(3RD-F\_A13)': [-30, 35, 206],

  'A13-F12(3RD-F\_A13)': [-30, 35, 206],

  'A13-F13(3RD-F\_A13)': [-30, 35, 206],

  'A13-F14(3RD-F\_A13)': [-30, 35, 206],

  'A13-F15(3RD-F\_A13)': [-30, 35, 206],

  'SMSS\_CHAIRPERSON\_ROOM(3RD-F\_A13)': [-30, 35, 206],

  'SMSS\_\_OFFICE(3RD-F\_A13)': [-30, 35, 206],

  'A14': [-15, 35, 186],

  'BIOGEOCHEMISTRY\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'DP\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'TINKERING\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'INNORVATION\_OFFICE(1ST-F\_A14)': [-15, 35, 186],

  'STEAM\_INNORVATION\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'CAM\_LAB(GROUND-F\_A14)': [-15, 35, 186],

  'SHSS\_OFFICE(2ND-F\_A14)': [-15, 35, 186],

  'LANGUAGE\_LAB(2ND-F\_A14)': [-15, 35, 186],

  'CONFERENCE\_ROOM(2ND-F\_A14)': [-15, 35, 186],

  'HCI\_CENTER(3RD-F\_A14)': [-15, 35, 186],

  'MATERIAL\_SCIENCE\_LAB(3RD-F\_A14)': [-15, 35, 186],

  'QUANTUM\_TECH\_CENTRE(3RD-F\_A14)': [-15, 35, 186],

  'A11': [-130, 40, 240],

  'A10': [-150, 40, 215],

  'A9': [-220, 40, 265],

  'CENTRAL\_LIBRARY': [30, 35, 150], // Ensure correct key

  'TULSI\_MESS': [-95, 35, 193], // Ensure correct key

  'TRAGOPAN\_CANTEEN': [-95, 35, 193], // Ensure correct key

  'PEEPAL\_MESS': [195, 35, -20], // Ensure correct key

  'CHAAT\_JUNCTION\_CANTEEN ': [195, 35, -20], // Ensure correct key

  'OAK\_MESS': [65, 35, 40],

  'MONAL\_CANTEEN': [65, 35, 40],

  'ORIGIN': [0, 20, 0],

  'SPORTS\_COMPLEX': [-240, 20, 170],

  'HEALTH\_CENTRE': [-275, 20, 200],

  'GUEST\_HOUSE': [-255, 25, 100],

  'AUDITORIUM': [-295, 25, 150],

  'VILLAGE\_SQUARE': [-275, 10, 130],

  'ALDER\_MESS': [165, 30, 95],

  'KUKU\_CANTEEN': [165, 30, 95],

  'DRONGO\_CANTEEN': [-110, 35, 150],

  'CAFE O MOCHA': [-305, 15, 100],

  'SUPERMARKET': [-315, 10, 110],

  'ROBOTRONICS\_LAB(4TH-F-A18)': [105, 50, 140],

  'MANAS\_LAB(4TH-F-A18)': [105, 50, 140],

  'ACS\_LAB(4TH-F-A18)': [105, 50, 140],

  'A-18-2A(3RD-F-A18)': [105, 50, 140],

  'IKSHMA\_CLASSROOM(3RD-F-A18)': [105, 50, 140],

  'SCEE-INFO-LAB(3RD-F-A18)': [105, 50, 140],

  'SP\_COM\_LAB(2ND-F-A18)': [105, 50, 140],

  'VLSI\_LAB(2ND-F-A18)': [105, 50, 140],

  'A18-A1(1ST-F-A18)': [105, 50, 140],

  'SCEE\_CONF-ROOM(1ST-F-A18)': [105, 50, 140],

  'DATA\_SCIENCE\_LAB(1ST-F-A18)': [105, 50, 140],

  'CHEMISTRY\_LAB(1ST-F-A18)': [105, 50, 140],

  'SCEE\_ELECTRONIC\_LAB(GROUND\_F-A18)': [105, 50, 140],

  'A-17-1-A(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-B(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-D(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-C(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-E(GROUND-F-A17)': [105, 40, 110],

  'A-17-2-A(1ST-F-A17)': [105, 40, 110],

  'A-17-2-B(1ST-F-A17)': [105, 40, 110],

  'A-17-2-C(1ST-F-A17)': [105, 40, 110],

  'A-17-2-D(1ST-F-A17)': [105, 40, 110],

  'A-17-2-E(1ST-F-A17)': [105, 40, 110],

  'CSP\_LAB(2ND-F-A17)': [105, 40, 110],

  'SCEE\_OFFICE(2ND-F-A17)': [105, 40, 110],

  'SCEE\_CHAIRPERSON\_ROOM(2ND-F-A17)': [105, 40, 110],

  'MIC\_LAB(3RD-F-A17)': [105, 40, 110],

  'MIC\_LAB(3RD-F-A17)': [105, 40, 110],

  'PHOTONICS\_LAB(3RD-F-A17)': [105, 40, 110],

  'NSS(1ST\_F\_A19)': [165, 30, 95],

  'YANTRIK\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'ROBOTRONICS\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'E-CELL(1ST\_F\_A19)': [165, 30, 95],

  'STAC\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'KAMAND\_PROMPT\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'HNT\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'NIRMAAN\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'KAMAND\_BIO\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'TECHNICAL\_OFFICE(1ST\_F\_A19)': [165, 30, 95],

  'DESIGNAUTS\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'WRITING\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'ART\_GREEKS\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'DEBATING\_AND\_QUIZZING\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'GYMKHANA\_MEETING\_ROOM(2ND\_F\_A19)': [165, 30, 95],

  'PMC\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'MUSIC\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'SPICMACAY\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'DRAMA\_CLUB(3-F\_PEEPAL\_MESS)': [195, 35, -20],

  'DANCE\_CLUB(3-F\_PEEPAL\_MESS)': [195, 35, -20],

  'CULTURAL\_SOCIETY\_OFFICE(3-F\_PEEPAL\_MESS)': [195, 35, -20],

}

function CoordinateMarker({ position, color, isSelected }) {

  return (

    <mesh position={position}>

      <tetrahedronGeometry args={[isSelected ? 8 : 5, 4]} /> {/\* Increase size if selected \*/}

      <meshBasicMaterial color={isSelected ? 'red' : color} /> {/\* Highlight color if selected \*/}

    </mesh>

  );

}

function Model() {

  const { scene } = useGLTF('/static/map2NOTREE.glb'); // Ensure the path to the model is correct

  return <primitive object={scene} />;

}

function CameraControls({ targetPosition }) {

  const { camera } = useThree();

  const controls = React.useRef();

  React.useEffect(() => {

    camera.position.set(0, 150, -200);  // Adjust this to zoom out further on load

    controls.current.update();

  }, [camera]);

  React.useEffect(() => {

    if (targetPosition) {

      controls.current.target.set(...targetPosition);

      const newCameraPosition = new THREE.Vector3(

        targetPosition[0] + 50,

        targetPosition[1] + 50,

        targetPosition[2] + 50

      );

      let frame = 0;

      const totalFrames = 60;

      const animateCamera = () => {

        frame++;

        const easedFrame = THREE.MathUtils.smoothstep(frame / totalFrames, 0, 1); // Smooth transition

        camera.position.lerp(newCameraPosition, easedFrame);

        camera.lookAt(...targetPosition);

        if (frame < totalFrames) {

          requestAnimationFrame(animateCamera);

        } else {

          controls.current.update();

        }

      };

      animateCamera();

    }

  }, [targetPosition, camera]);

  return <OrbitControls ref={controls} />;

}

function ModelView() {

  const [fromBuilding, setFromBuilding] = useState('');

  const [toBuilding, setToBuilding] = useState('');

  const [fromSuggestions, setFromSuggestions] = useState([]);

  const [toSuggestions, setToSuggestions] = useState([]);

  const [fromPosition, setFromPosition] = useState(null);

  const [toPosition, setToPosition] = useState(null);

  const [targetPosition, setTargetPosition] = useState(null);

  const [selectedBuilding, setSelectedBuilding] = useState(null); // State to track selected building

  // Debounced search handler

  const handleFromChange = useCallback((event) => {

    const value = event.target.value;

    setFromBuilding(value);

    if (value) {

      const debounceTimeout = setTimeout(() => {

        const filteredSuggestions = Object.keys(buildingCoordinates).filter((key) =>

          key.toLowerCase().includes(value.toLowerCase())

        );

        setFromSuggestions(filteredSuggestions);

      }, 300);  // 300ms debounce

      return () => clearTimeout(debounceTimeout);  // Clear the timeout if input changes again

    } else {

      setFromSuggestions([]);

    }

  }, []);

  const handleToChange = useCallback((event) => {

    const value = event.target.value;

    setToBuilding(value);

    if (value) {

      const debounceTimeout = setTimeout(() => {

        const filteredSuggestions = Object.keys(buildingCoordinates).filter((key) =>

          key.toLowerCase().includes(value.toLowerCase())

        );

        setToSuggestions(filteredSuggestions);

      }, 300);  // 300ms debounce

      return () => clearTimeout(debounceTimeout);  // Clear the timeout if input changes again

    } else {

      setToSuggestions([]);

    }

  }, []);

  const handleFromSuggestionClick = (suggestion) => {

    setFromBuilding(suggestion);

    setFromSuggestions([]);

    const position = buildingCoordinates[suggestion.toUpperCase()];

    if (position) {

      setFromPosition(position);

      setTargetPosition(position);

      setSelectedBuilding(suggestion); // Set the selected building

    }

  };

  const handleToSuggestionClick = (suggestion) => {

    setToBuilding(suggestion);

    setToSuggestions([]);

    const position = buildingCoordinates[suggestion.toUpperCase()];

    if (position) {

      setToPosition(position);

      setTargetPosition(position);

      setSelectedBuilding(suggestion); // Set the selected building

    }

  };

const [highlightedFromIndex, setHighlightedFromIndex] = useState(-1);

const [highlightedToIndex, setHighlightedToIndex] = useState(-1);

  return (

    <div>

      {/\* Stylish Search Bar Container \*/}

      <div style={{ position: 'absolute', right: '0px', top: '50%', transform: 'translateY(-50%)', width: '250px' }}>

        <div style={{ marginBottom: '20px', position: 'relative' }}>

        <input

        type="text"

        value={fromBuilding}

        onChange={handleFromChange}

        placeholder="From (e.g., B8)"

        style={searchBarStyle}

        onKeyDown={(e) => {

          if (e.key === 'Enter' && fromSuggestions.length > 0) {

            handleFromSuggestionClick(fromSuggestions[highlightedFromIndex >= 0 ? highlightedFromIndex : 0]); // Select highlighted or first suggestion on Enter

          } else if (e.key === 'ArrowDown') {

            setHighlightedFromIndex((prevIndex) => Math.min(prevIndex + 1, fromSuggestions.length - 1)); // Navigate down

          } else if (e.key === 'ArrowUp') {

            setHighlightedFromIndex((prevIndex) => Math.max(prevIndex - 1, 0)); // Navigate up

          }

        }}

/>

          {fromSuggestions.length > 0 && (

            <ul style={suggestionsStyle}>

              {fromSuggestions.map((suggestion,index) => (

                <li

                  key={suggestion}

                  onClick={() => handleFromSuggestionClick(suggestion)}

                  style={{

                    ...suggestionItemStyle,

                    backgroundColor: highlightedFromIndex === index ? '#ddd' : 'white', // Highlight if selected

                    color: highlightedFromIndex === index ? '#333' : '#555',

                  }}

                >

                  {suggestion}

                </li>

              ))}

            </ul>

          )}

        </div>

        <div style={{ marginBottom: '10px', position: 'relative' }}>

        <input

        type="text"

        value={toBuilding}

        onChange={handleToChange}

        placeholder="To (e.g., B12)"

        style={searchBarStyle}

        onKeyDown={(e) => {

          if (e.key === 'Enter' && toSuggestions.length > 0) {

            handleToSuggestionClick(toSuggestions[highlightedToIndex >= 0 ? highlightedToIndex : 0]); // Select highlighted or first suggestion on Enter

          } else if (e.key === 'ArrowDown') {

            setHighlightedToIndex((prevIndex) => Math.min(prevIndex + 1, toSuggestions.length - 1)); // Navigate down

          } else if (e.key === 'ArrowUp') {

            setHighlightedToIndex((prevIndex) => Math.max(prevIndex - 1, 0)); // Navigate up

          }

        }}

/>

          {toSuggestions.length > 0 && (

            <ul style={suggestionsStyle}>

              {toSuggestions.map((suggestion,index) => (

                <li

                  key={suggestion}

                  onClick={() => handleToSuggestionClick(suggestion)}

                  style={{

                    ...suggestionItemStyle,

                    backgroundColor: highlightedToIndex === index ? '#ddd' : 'white', // Highlight if selected

                    color: highlightedToIndex === index ? '#333' : '#555',

                  }}

                >

                  {suggestion}

                </li>

              ))}

            </ul>

          )}

        </div>

      </div>

      <Canvas style={{ height: '70vh', width: '60vw' ,paddingLeft: '200px'}}>

        <ambientLight intensity={1} />

        <pointLight position={[10, 10, 10]} intensity={1} />

        <Model />

        {fromPosition && (

        <CoordinateMarker

        position={fromPosition}

        color="yellow"

        isSelected={selectedBuilding === fromBuilding} // Compare with selected building

          />

            )}

        {toPosition && (

         <CoordinateMarker

        position={toPosition}

        color="white"

        isSelected={selectedBuilding === toBuilding} // Compare with selected building

       />

        )}

        <CameraControls targetPosition={targetPosition} />

      </Canvas>

    </div>

  );

}

// Custom CSS styles for the search bar and suggestions

const searchBarStyle = {

  width: '100%',

  padding: '12px 15px',

  fontSize: '18px', // Increased font size for better readability

  borderRadius: '25px', // More rounded corners

  border: '1px solid #ddd',

  boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)', // Softer shadow

  outline: 'none',

  transition: 'all 0.3s ease', // Smooth transition effect

  backgroundColor: 'linear-gradient(145deg, #f6f6f6, #ffffff)', // Subtle gradient

  color: '#333', // Darker text color for contrast

};

// Style for suggestions dropdown

const suggestionsStyle = {

  position: 'absolute',

  top: '50px', // Increased for spacing

  left: '0',

  backgroundColor: 'white',

  border: '1px solid #ddd',

  borderRadius: '15px', // Rounded corners for suggestions box

  listStyle: 'none',

  padding: '0',

  margin: '0',

  width: '100%',

  zIndex: 10,

  boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)', // Shadow for dropdown box

};

// Style for each suggestion item

const suggestionItemStyle = {

  padding: '12px', // Increased padding for better spacing

  cursor: 'pointer',

  borderBottom: '1px solid #eee',

  transition: 'background-color 0.3s ease, color 0.3s ease', // Smooth hover effect

  borderRadius: '10px', // Subtle rounding for each item

  fontSize: '16px', // Larger font for readability

  color: '#555', // Slightly darker text

};

export default ModelView;

import React, { useState, useCallback } from 'react';

import { Canvas, useThree } from '@react-three/fiber';

import { OrbitControls, useGLTF } from '@react-three/drei';

import \* as THREE from 'three';  // Import THREE for vector operations

// Coordinates mapping for buildings

const buildingCoordinates = {

  'B8': [-10, 20, 45],

  'B12': [30, 20, 100],

  'B9': [30, 20, 40],

  'B11': [0, 35, 80],

  'B19': [-55, 25, 110],

  'B16': [-100, 25, 65],

  'B20': [-140, 25, 80],

  'B22': [-180, 25, 110],

  'B21': [-165, 25, 135],

  'B23': [-110, 25, 115],

  'B13': [-190, 25, 170],

  'PINEMESS': [-110, 35, 150],

  'B14': [-140, 25, 170],

  'B15': [-60, 25, 150],

  'B18': [-20, 25, 135],

  'B10': [65, 25, 75],

  'B17': [105, 25, 25],

  'B24': [150, 25, 5],

  'B26': [170, 30, 35],

  'B25': [130, 30, 60],

  'A19': [165, 30, 95],

  'A17': [105, 40, 110],

  'A18': [105, 50, 140],

  'A13': [-15, 35, 166],

  'AVL(GROUND-F\_A13)': [-15, 35, 166],

  'MNC-LAB(GROUND-F\_A13)': [-15, 35, 166],

  'A13-1A(GROUND-F\_A13)': [-15, 35, 166],

  'A13-2A(1ST-F\_A13)': [-15, 35, 166],

  'A13-2B(1ST-F\_A13)': [-15, 35, 166],

  'A13-2C(1ST-F\_A13)': [-15, 35, 166],

  'A13-2D(1ST-F\_A13)': [-15, 35, 166],

  'A13-3A(2ND-F\_A13)': [-15, 35, 166],

  'A13-L1(2ND-F\_A13)': [-15, 35, 166],

  'A13-L2(2ND-F\_A13)': [-30, 35, 206],

  'NKN\_CONFERENCE\_ROOM(3RD-F\_A13)': [-30, 35, 206],

  'A13-F1(3RD-F\_A13)': [-30, 35, 206],

  'A13-F2(3RD-F\_A13)': [-30, 35, 206],

  'A13-F7(3RD-F\_A13)': [-30, 35, 206],

  'A13-F4(3RD-F\_A13)': [-30, 35, 206],

  'A13-F3(3RD-F\_A13)': [-30, 35, 206],

  'A13-F5(3RD-F\_A13)': [-30, 35, 206],

  'A13-F6(3RD-F\_A13)': [-30, 35, 206],

  'A13-F9(3RD-F\_A13)': [-30, 35, 206],

  'A13-F11(3RD-F\_A13)': [-30, 35, 206],

  'A13-F12(3RD-F\_A13)': [-30, 35, 206],

  'A13-F13(3RD-F\_A13)': [-30, 35, 206],

  'A13-F14(3RD-F\_A13)': [-30, 35, 206],

  'A13-F15(3RD-F\_A13)': [-30, 35, 206],

  'SMSS\_CHAIRPERSON\_ROOM(3RD-F\_A13)': [-30, 35, 206],

  'SMSS\_\_OFFICE(3RD-F\_A13)': [-30, 35, 206],

  'A14': [-15, 35, 186],

  'BIOGEOCHEMISTRY\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'DP\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'TINKERING\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'INNORVATION\_OFFICE(1ST-F\_A14)': [-15, 35, 186],

  'STEAM\_INNORVATION\_LAB(1ST-F\_A14)': [-15, 35, 186],

  'CAM\_LAB(GROUND-F\_A14)': [-15, 35, 186],

  'SHSS\_OFFICE(2ND-F\_A14)': [-15, 35, 186],

  'LANGUAGE\_LAB(2ND-F\_A14)': [-15, 35, 186],

  'CONFERENCE\_ROOM(2ND-F\_A14)': [-15, 35, 186],

  'HCI\_CENTER(3RD-F\_A14)': [-15, 35, 186],

  'MATERIAL\_SCIENCE\_LAB(3RD-F\_A14)': [-15, 35, 186],

  'QUANTUM\_TECH\_CENTRE(3RD-F\_A14)': [-15, 35, 186],

  'A11': [-130, 40, 240],

  'A10': [-150, 40, 215],

  'A9': [-220, 40, 265],

  'CENTRAL\_LIBRARY': [30, 35, 150], // Ensure correct key

  'TULSI\_MESS': [-95, 35, 193], // Ensure correct key

  'TRAGOPAN\_CANTEEN': [-95, 35, 193], // Ensure correct key

  'PEEPAL\_MESS': [195, 35, -20], // Ensure correct key

  'CHAAT\_JUNCTION\_CANTEEN ': [195, 35, -20], // Ensure correct key

  'OAK\_MESS': [65, 35, 40],

  'MONAL\_CANTEEN': [65, 35, 40],

  'ORIGIN': [0, 20, 0],

  'SPORTS\_COMPLEX': [-240, 20, 170],

  'HEALTH\_CENTRE': [-275, 20, 200],

  'GUEST\_HOUSE': [-255, 25, 100],

  'AUDITORIUM': [-295, 25, 150],

  'VILLAGE\_SQUARE': [-275, 10, 130],

  'ALDER\_MESS': [165, 30, 95],

  'KUKU\_CANTEEN': [165, 30, 95],

  'DRONGO\_CANTEEN': [-110, 35, 150],

  'CAFE O MOCHA': [-305, 15, 100],

  'SUPERMARKET': [-315, 10, 110],

  'ROBOTRONICS\_LAB(4TH-F-A18)': [105, 50, 140],

  'MANAS\_LAB(4TH-F-A18)': [105, 50, 140],

  'ACS\_LAB(4TH-F-A18)': [105, 50, 140],

  'A-18-2A(3RD-F-A18)': [105, 50, 140],

  'IKSHMA\_CLASSROOM(3RD-F-A18)': [105, 50, 140],

  'SCEE-INFO-LAB(3RD-F-A18)': [105, 50, 140],

  'SP\_COM\_LAB(2ND-F-A18)': [105, 50, 140],

  'VLSI\_LAB(2ND-F-A18)': [105, 50, 140],

  'A18-A1(1ST-F-A18)': [105, 50, 140],

  'SCEE\_CONF-ROOM(1ST-F-A18)': [105, 50, 140],

  'DATA\_SCIENCE\_LAB(1ST-F-A18)': [105, 50, 140],

  'CHEMISTRY\_LAB(1ST-F-A18)': [105, 50, 140],

  'SCEE\_ELECTRONIC\_LAB(GROUND\_F-A18)': [105, 50, 140],

  'A-17-1-A(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-B(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-D(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-C(GROUND-F-A17)': [105, 40, 110],

  'A-17-1-E(GROUND-F-A17)': [105, 40, 110],

  'A-17-2-A(1ST-F-A17)': [105, 40, 110],

  'A-17-2-B(1ST-F-A17)': [105, 40, 110],

  'A-17-2-C(1ST-F-A17)': [105, 40, 110],

  'A-17-2-D(1ST-F-A17)': [105, 40, 110],

  'A-17-2-E(1ST-F-A17)': [105, 40, 110],

  'CSP\_LAB(2ND-F-A17)': [105, 40, 110],

  'SCEE\_OFFICE(2ND-F-A17)': [105, 40, 110],

  'SCEE\_CHAIRPERSON\_ROOM(2ND-F-A17)': [105, 40, 110],

  'MIC\_LAB(3RD-F-A17)': [105, 40, 110],

  'MIC\_LAB(3RD-F-A17)': [105, 40, 110],

  'PHOTONICS\_LAB(3RD-F-A17)': [105, 40, 110],

  'NSS(1ST\_F\_A19)': [165, 30, 95],

  'YANTRIK\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'ROBOTRONICS\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'E-CELL(1ST\_F\_A19)': [165, 30, 95],

  'STAC\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'KAMAND\_PROMPT\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'HNT\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'NIRMAAN\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'KAMAND\_BIO\_CLUB(1ST\_F\_A19)': [165, 30, 95],

  'TECHNICAL\_OFFICE(1ST\_F\_A19)': [165, 30, 95],

  'DESIGNAUTS\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'WRITING\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'ART\_GREEKS\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'DEBATING\_AND\_QUIZZING\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'GYMKHANA\_MEETING\_ROOM(2ND\_F\_A19)': [165, 30, 95],

  'PMC\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'MUSIC\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'SPICMACAY\_CLUB(2ND\_F\_A19)': [165, 30, 95],

  'DRAMA\_CLUB(3-F\_PEEPAL\_MESS)': [195, 35, -20],

  'DANCE\_CLUB(3-F\_PEEPAL\_MESS)': [195, 35, -20],

  'CULTURAL\_SOCIETY\_OFFICE(3-F\_PEEPAL\_MESS)': [195, 35, -20],

}

function CoordinateMarker({ position, color, isSelected }) {

  return (

    <mesh position={position}>

      <tetrahedronGeometry args={[isSelected ? 8 : 5, 4]} /> {/\* Increase size if selected \*/}

      <meshBasicMaterial color={isSelected ? 'red' : color} /> {/\* Highlight color if selected \*/}

    </mesh>

  );

}

function Model() {

  const { scene } = useGLTF('/static/map2NOTREE.glb'); // Ensure the path to the model is correct

  return <primitive object={scene} />;

}

function CameraControls({ targetPosition }) {

  const { camera } = useThree();

  const controls = React.useRef();

  React.useEffect(() => {

    camera.position.set(0, 150, -200);  // Adjust this to zoom out further on load

    controls.current.update();

  }, [camera]);

  React.useEffect(() => {

    if (targetPosition) {

      controls.current.target.set(...targetPosition);

      const newCameraPosition = new THREE.Vector3(

        targetPosition[0] + 50,

        targetPosition[1] + 50,

        targetPosition[2] + 50

      );

      let frame = 0;

      const totalFrames = 60;

      const animateCamera = () => {

        frame++;

        const easedFrame = THREE.MathUtils.smoothstep(frame / totalFrames, 0, 1); // Smooth transition

        camera.position.lerp(newCameraPosition, easedFrame);

        camera.lookAt(...targetPosition);

        if (frame < totalFrames) {

          requestAnimationFrame(animateCamera);

        } else {

          controls.current.update();

        }

      };

      animateCamera();

    }

  }, [targetPosition, camera]);

  return <OrbitControls ref={controls} />;

}

function ModelView() {

  const [fromBuilding, setFromBuilding] = useState('');

  const [toBuilding, setToBuilding] = useState('');

  const [fromSuggestions, setFromSuggestions] = useState([]);

  const [toSuggestions, setToSuggestions] = useState([]);

  const [fromPosition, setFromPosition] = useState(null);

  const [toPosition, setToPosition] = useState(null);

  const [targetPosition, setTargetPosition] = useState(null);

  const [selectedBuilding, setSelectedBuilding] = useState(null); // State to track selected building

  // Debounced search handler

  const handleFromChange = useCallback((event) => {

    const value = event.target.value;

    setFromBuilding(value);

    if (value) {

      const debounceTimeout = setTimeout(() => {

        const filteredSuggestions = Object.keys(buildingCoordinates).filter((key) =>

          key.toLowerCase().includes(value.toLowerCase())

        );

        setFromSuggestions(filteredSuggestions);

      }, 300);  // 300ms debounce

      return () => clearTimeout(debounceTimeout);  // Clear the timeout if input changes again

    } else {

      setFromSuggestions([]);

    }

  }, []);

  const handleToChange = useCallback((event) => {

    const value = event.target.value;

    setToBuilding(value);

    if (value) {

      const debounceTimeout = setTimeout(() => {

        const filteredSuggestions = Object.keys(buildingCoordinates).filter((key) =>

          key.toLowerCase().includes(value.toLowerCase())

        );

        setToSuggestions(filteredSuggestions);

      }, 300);  // 300ms debounce

      return () => clearTimeout(debounceTimeout);  // Clear the timeout if input changes again

    } else {

      setToSuggestions([]);

    }

  }, []);

  const handleFromSuggestionClick = (suggestion) => {

    setFromBuilding(suggestion);

    setFromSuggestions([]);

    const position = buildingCoordinates[suggestion.toUpperCase()];

    if (position) {

      setFromPosition(position);

      setTargetPosition(position);

      setSelectedBuilding(suggestion); // Set the selected building

    }

  };

  const handleToSuggestionClick = (suggestion) => {

    setToBuilding(suggestion);

    setToSuggestions([]);

    const position = buildingCoordinates[suggestion.toUpperCase()];

    if (position) {

      setToPosition(position);

      setTargetPosition(position);

      setSelectedBuilding(suggestion); // Set the selected building

    }

  };

const [highlightedFromIndex, setHighlightedFromIndex] = useState(-1);

const [highlightedToIndex, setHighlightedToIndex] = useState(-1);

  return (

    <div>

      {/\* Stylish Search Bar Container \*/}

      <div style={{ position: 'absolute', right: '0px', top: '50%', transform: 'translateY(-50%)', width: '250px' }}>

        <div style={{ marginBottom: '20px', position: 'relative' }}>

        <input

        type="text"

        value={fromBuilding}

        onChange={handleFromChange}

        placeholder="From (e.g., B8)"

        style={searchBarStyle}

        onKeyDown={(e) => {

          if (e.key === 'Enter' && fromSuggestions.length > 0) {

            handleFromSuggestionClick(fromSuggestions[highlightedFromIndex >= 0 ? highlightedFromIndex : 0]); // Select highlighted or first suggestion on Enter

          } else if (e.key === 'ArrowDown') {

            setHighlightedFromIndex((prevIndex) => Math.min(prevIndex + 1, fromSuggestions.length - 1)); // Navigate down

          } else if (e.key === 'ArrowUp') {

            setHighlightedFromIndex((prevIndex) => Math.max(prevIndex - 1, 0)); // Navigate up

          }

        }}

/>

          {fromSuggestions.length > 0 && (

            <ul style={suggestionsStyle}>

              {fromSuggestions.map((suggestion,index) => (

                <li

                  key={suggestion}

                  onClick={() => handleFromSuggestionClick(suggestion)}

                  style={{

                    ...suggestionItemStyle,

                    backgroundColor: highlightedFromIndex === index ? '#ddd' : 'white', // Highlight if selected

                    color: highlightedFromIndex === index ? '#333' : '#555',

                  }}

                >

                  {suggestion}

                </li>

              ))}

            </ul>

          )}

        </div>

        <div style={{ marginBottom: '10px', position: 'relative' }}>

        <input

        type="text"

        value={toBuilding}

        onChange={handleToChange}

        placeholder="To (e.g., B12)"

        style={searchBarStyle}

        onKeyDown={(e) => {

          if (e.key === 'Enter' && toSuggestions.length > 0) {

            handleToSuggestionClick(toSuggestions[highlightedToIndex >= 0 ? highlightedToIndex : 0]); // Select highlighted or first suggestion on Enter

          } else if (e.key === 'ArrowDown') {

            setHighlightedToIndex((prevIndex) => Math.min(prevIndex + 1, toSuggestions.length - 1)); // Navigate down

          } else if (e.key === 'ArrowUp') {

            setHighlightedToIndex((prevIndex) => Math.max(prevIndex - 1, 0)); // Navigate up

          }

        }}

/>

          {toSuggestions.length > 0 && (

            <ul style={suggestionsStyle}>

              {toSuggestions.map((suggestion,index) => (

                <li

                  key={suggestion}

                  onClick={() => handleToSuggestionClick(suggestion)}

                  style={{

                    ...suggestionItemStyle,

                    backgroundColor: highlightedToIndex === index ? '#ddd' : 'white', // Highlight if selected

                    color: highlightedToIndex === index ? '#333' : '#555',

                  }}

                >

                  {suggestion}

                </li>

              ))}

            </ul>

          )}

        </div>

      </div>

      <Canvas style={{ height: '70vh', width: '60vw' ,paddingLeft: '200px'}}>

        <ambientLight intensity={1} />

        <pointLight position={[10, 10, 10]} intensity={1} />

        <Model />

        {fromPosition && (

        <CoordinateMarker

        position={fromPosition}

        color="yellow"

        isSelected={selectedBuilding === fromBuilding} // Compare with selected building

          />

            )}

        {toPosition && (

         <CoordinateMarker

        position={toPosition}

        color="white"

        isSelected={selectedBuilding === toBuilding} // Compare with selected building

       />

        )}

        <CameraControls targetPosition={targetPosition} />

      </Canvas>

    </div>

  );

}

// Custom CSS styles for the search bar and suggestions

const searchBarStyle = {

  width: '100%',

  padding: '12px 15px',

  fontSize: '18px', // Increased font size for better readability

  borderRadius: '25px', // More rounded corners

  border: '1px solid #ddd',

  boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)', // Softer shadow

  outline: 'none',

  transition: 'all 0.3s ease', // Smooth transition effect

  backgroundColor: 'linear-gradient(145deg, #f6f6f6, #ffffff)', // Subtle gradient

  color: '#333', // Darker text color for contrast

};

// Style for suggestions dropdown

const suggestionsStyle = {

  position: 'absolute',

  top: '50px', // Increased for spacing

  left: '0',

  backgroundColor: 'white',

  border: '1px solid #ddd',

  borderRadius: '15px', // Rounded corners for suggestions box

  listStyle: 'none',

  padding: '0',

  margin: '0',

  width: '100%',

  zIndex: 10,

  boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)', // Shadow for dropdown box

};

// Style for each suggestion item

const suggestionItemStyle = {

  padding: '12px', // Increased padding for better spacing

  cursor: 'pointer',

  borderBottom: '1px solid #eee',

  transition: 'background-color 0.3s ease, color 0.3s ease', // Smooth hover effect

  borderRadius: '10px', // Subtle rounding for each item

  fontSize: '16px', // Larger font for readability

  color: '#555', // Slightly darker text

};

export default ModelView;