

# WebGIS 项目报告

## 项目地址

<https://github.com/poorwym/WebGIS.git>

同步在HW5文件夹

## 使用方法

- 安装依赖

```
npm install
```

- 运行服务器

```
npm run dev
```

## 技术栈

- 前端框架: React + Vite
- 地图引擎: Cesium
- 状态管理: React Hooks
- 样式管理: CSS + Tailwind CSS + MaterialUI

## 主要功能

### 1. 地图显示

- 支持多种影像图层切换
  - 天地图矢量图层
  - ArcGIS 影像图层
- 支持地形显示

## 2. 交互控制

- 光照效果开关
- 水面效果开关
- 影像图层选择器
- 地形提供者选择器

## 项目结构

```
|── src/
|   ├── components/      # 组件目录
|   ├── data/             # 数据配置
|   ├── hooks/            # 自定义 Hooks
|   ├── assets/           # 静态资源
|   ├── App.jsx           # 主应用组件
|   └── main.jsx          # 应用入口
├── public/              # 公共资源
└── package.json         # 项目配置
```

## 核心组件

### CesiumContainer

- 主要的地图容器组件
- 负责 Cesium 场景的初始化和渲染
- 管理地图视图和交互

关键代码实现：

```
function CesiumContainer({ imageProvider, terrainProvider, lighting, waterMask }) {
  const [viewer, setViewer] = useState(null);

  // 初始化 Cesium Viewer
  useEffect(() => {
    const newViewer = new Cesium.Viewer("cesiumContainer", {
      baseLayerPicker: false,
      geocoder: false,
      homeButton: false,
      sceneModePicker: false,
      navigationHelpButton: false,
      animation: false,
      timeline: false,
      fullscreenButton: false,
      infoBox: false,
      imageryProvider: false,
    });
    setViewer(newViewer);
    return () => {
      if (newViewer && !newViewer.isDestroyed()) {
        newViewer.destroy();
      }
    };
  }, []);

  // 处理光照和水面效果
  useEffect(() => {
    if (!viewer) return;
    viewer.scene.globe.enableLighting = lighting;
    viewer.scene.globe.showWaterEffect = waterMask;
    if (terrainProvider) {
      viewer.scene.setTerrain(terrainProvider);
    }
  }, [viewer, lighting, terrainProvider, waterMask]);

  // 处理影像图层
  useEffect(() => {
    if (!viewer) return;
    viewer.imageryLayers.removeAll();
    if (imageProvider) {
      viewer.imageryLayers.addImageryProvider(imageProvider);
    }
  }, [viewer, imageProvider]);
}
```

```
    return <div id="cesiumContainer" className='w-full h-full'></div>;
}
```

## 控制组件

### LightSwitch

控制场景光照效果的开关组件：

```
function LightSwitch({option, setOption}) {
  return (
    <FormGroup>
      <FormControlLabel
        control={
          <Switch
            checked={option.lighting}
            onChange={(e) => setOption({...option, lighting: e.target.checked})}
          />
        }
        label="Lighting"
      />
    </FormGroup>
  );
}
```

### WaterMaskSwitch

控制水面效果显示的开关组件：

```

function WaterMaskSwitch({option, setOption}) {
  return (
    <FormGroup>
      <FormControlLabel
        control={
          <Switch
            checked={option.waterMask}
            onChange={(e) => setOption({...option, waterMask: e.target.checked})}
          />
        }
        label="Water Mask"
      />
    </FormGroup>
  );
}

```

## ImagerProviderSelector

影像图层选择器组件：

```

function ImagerProviderSelector({option, setOption}) {
  return (
    <Box sx={{ minWidth: 120 }}>
      <FormControl fullWidth>
        <InputLabel>Image Provider</InputLabel>
        <Select
          value={option.imageryProvider}
          label="Image Provider"
          onChange={(e) => setOption({...option, imageryProvider: e.target.value})}
        >
          {Object.keys(image_provider_source).map((key) => (
            <MenuItem key={key} value={key}>{key}</MenuItem>
          ))}
        </Select>
      </FormControl>
    </Box>
  );
}

```

## TerrainProviderSelector

地形图层选择器组件：

```

function TerrainProviderSelector({option, setOption}) {
  return (
    <Box sx={{ minWidth: 120 }}>
      <FormControl fullWidth>
        <InputLabel>Terrain Provider</InputLabel>
        <Select
          value={option.terrainProvider}
          label="Terrain Provider"
          onChange={(e) => setOption({...option, terrainProvider: e.target.value})}
        >
          {Object.keys(terrain_provider_source).map((key) => (
            <MenuItem key={key} value={key}>{key}</MenuItem>
          ))}
        </Select>
      </FormControl>
    </Box>
  );
}

```

## 技术实现细节

### 1. 地图服务配置

```

Cesium.Ion.defaultAccessToken = '...'
Cesium.Camera.DEFAULT_VIEW_RECTANGLE = Cesium.Rectangle.fromDegrees(
  114.218926, 30.585424,
  114.218926, 30.585424
)

```

### 2. 影像提供者管理

- 支持动态切换不同的影像图层
- 实现了 WebMapTileService 和 ArcGIS 地图服务的集成
- 异步加载和错误处理机制

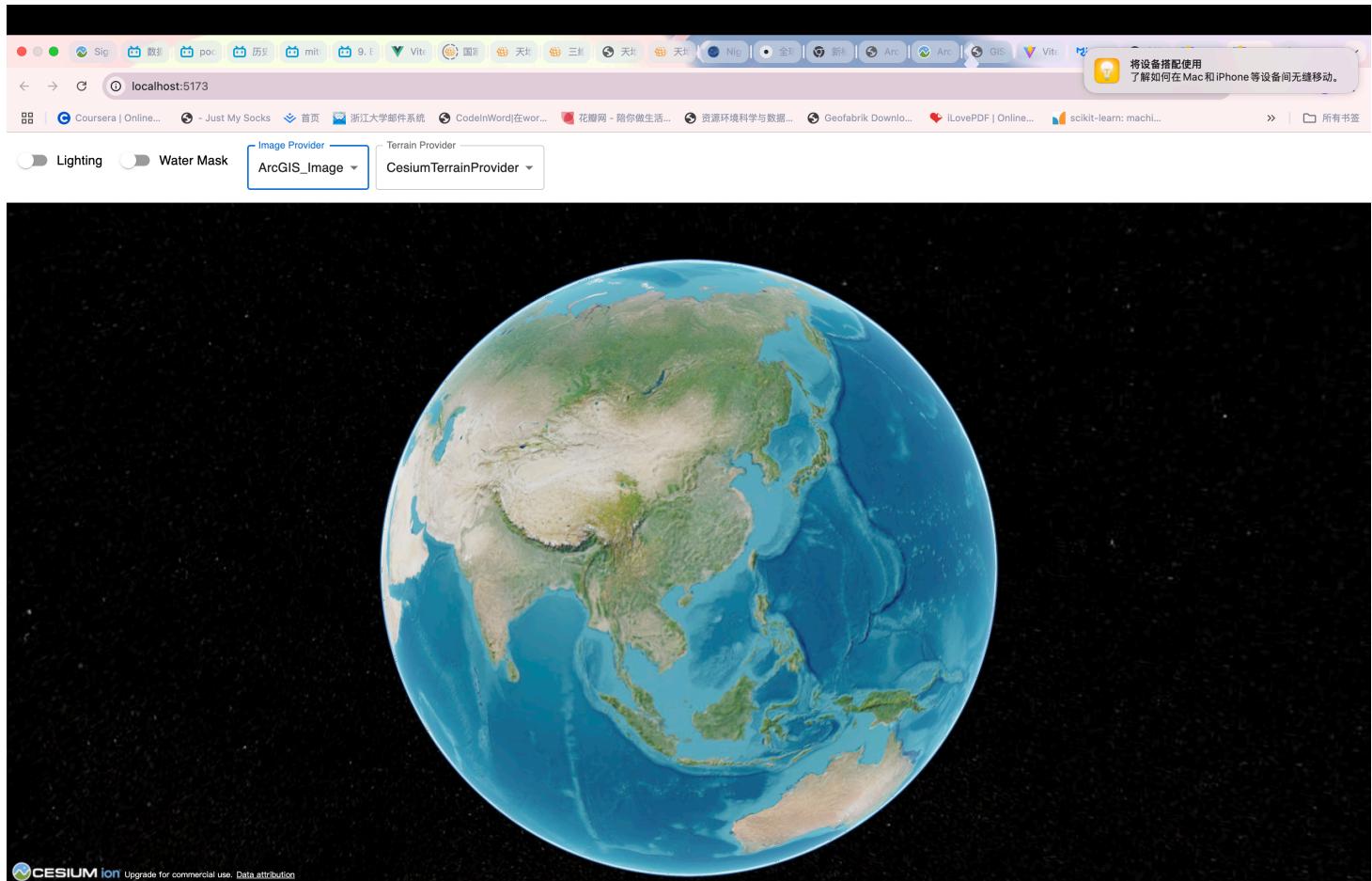
### 3. 地形系统

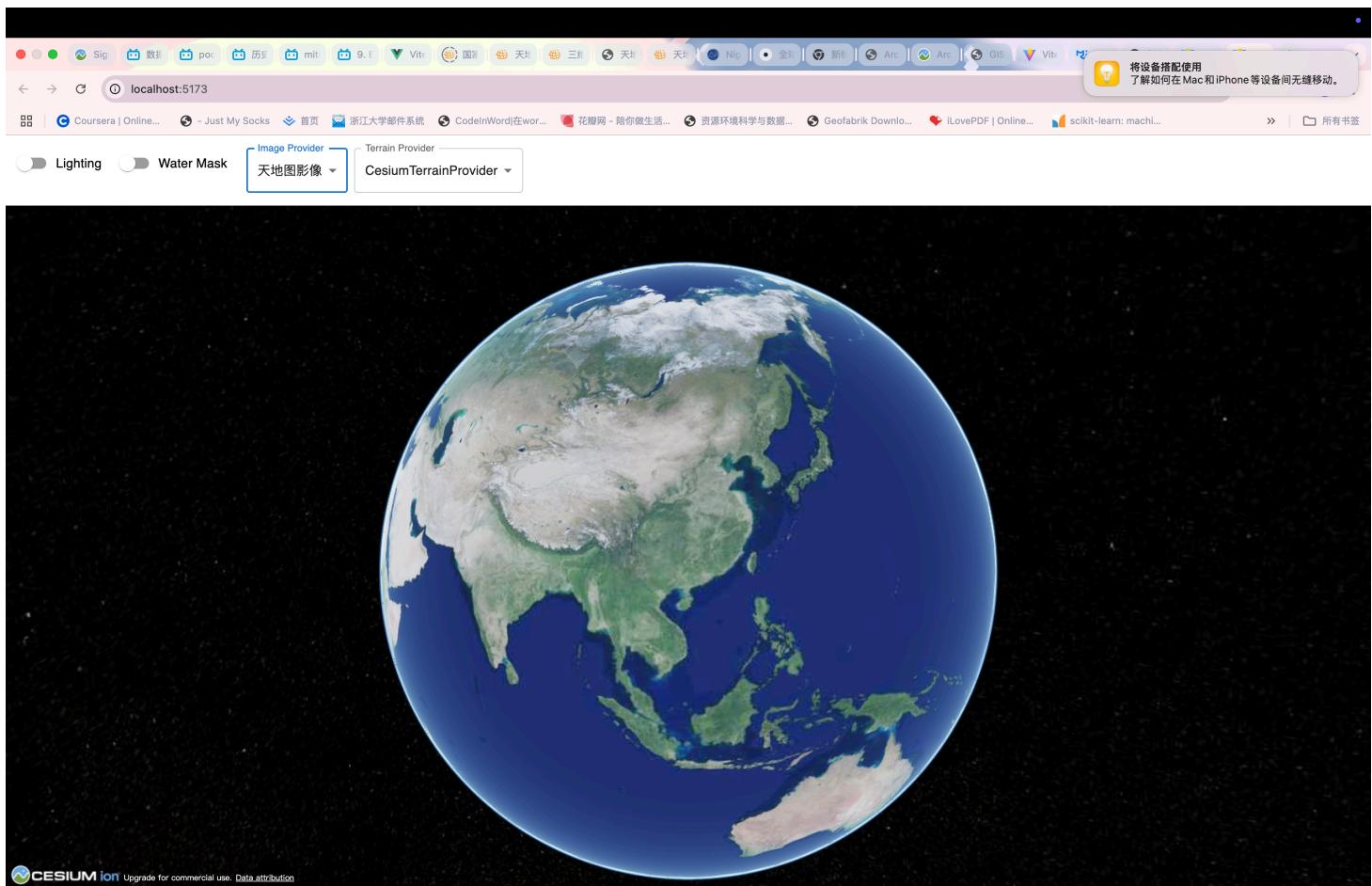
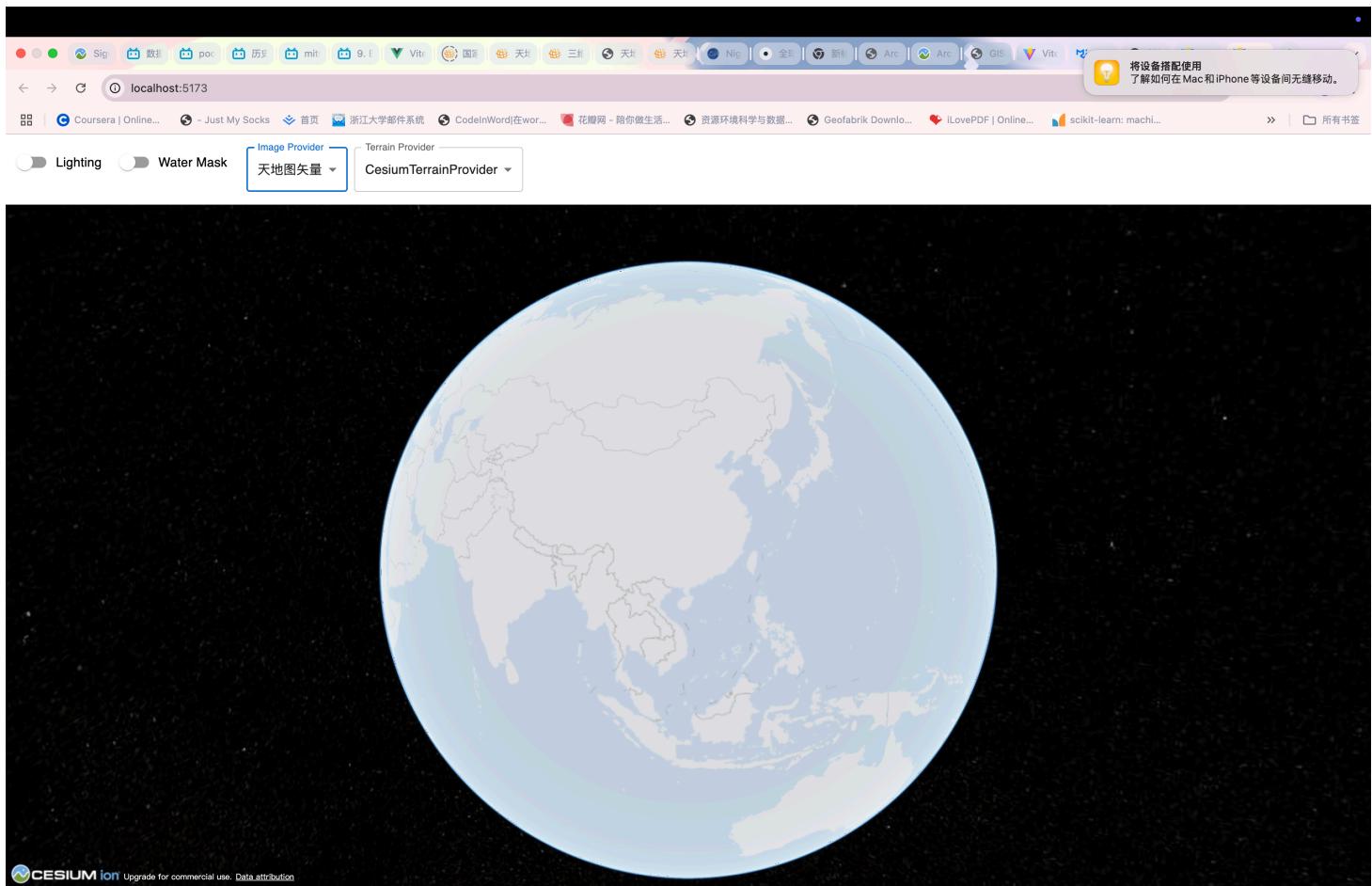
- 使用 Cesium World Terrain 作为地形数据源
- 支持水面效果的可选显示

- 动态更新地形提供者

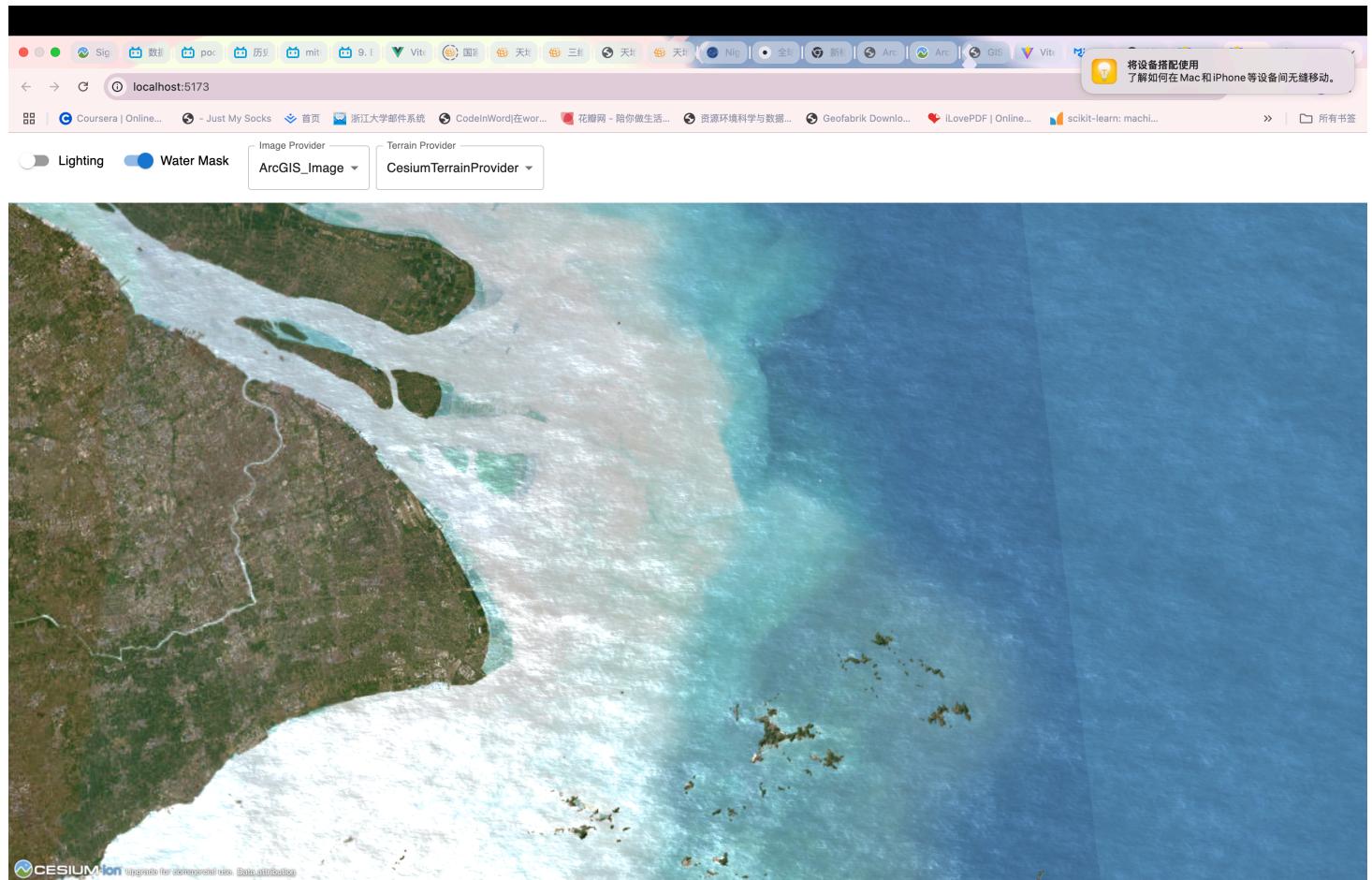
## 实验结果

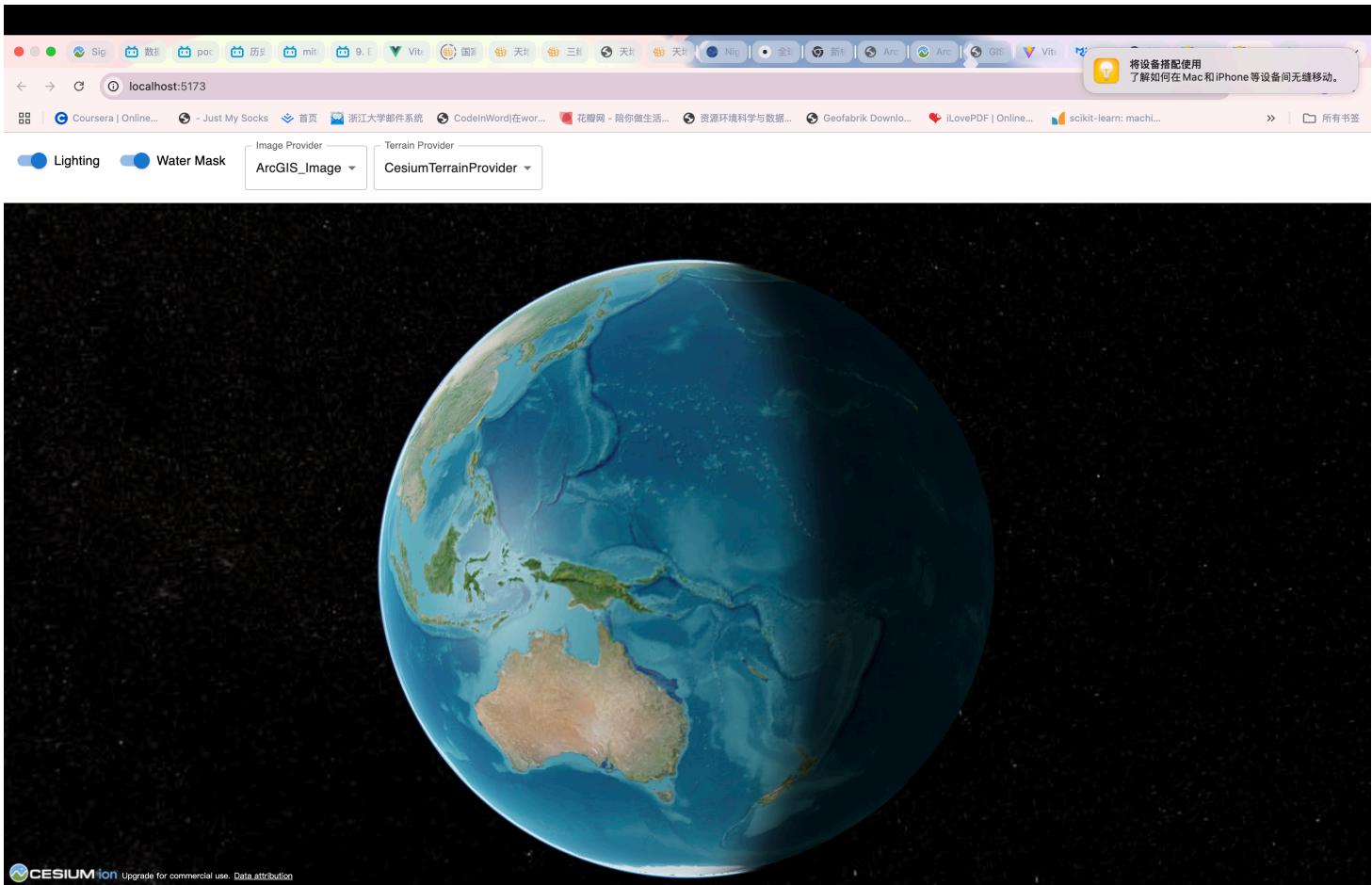
支持ArcGIS影像，天地图影像与天地图矢量三个数据源





# 支持开启光照与water mask





# 支持Cesium World Terrain

