# SDN

## 1.three中包含哪些要素？

1.渲染器Renderer。项目中使用的是WebGLRenderer，传入场景Scene和摄像机camera到渲染器中，通过WebGL将三维空间渲染到canvas上。初始化Renderer的时候，会自动创建一个canvas

2.场景对象Scene，是一个树状结构，展示各个对象的位置和方向。包含网格对象Mesh、三维物体Object3D和摄像机

3.网格对象Mesh，为用一种特定的材质(Material)来绘制的一个特定的几何体(Geometry)

4.几何体Geometry，球体、立方体

5.材质Material，几何体表面属性，包括颜色、灯光亮度等

6.摄像机Camera，项目中使用的是正交摄像机 OrthographicCamera，通过设置left, right top, bottom, near, 和far指定一个长方体

7.光照Light，项目中使用的是环境光AmbientLight，会均匀的照亮场景中的所有物体

## 2.分页查询怎么实现？

使用antd里面的table组件，设置table的pagination属性，包括current、pagesize、count。首次查询时给出current=1，pagesize，服务器返回pagesize个条目，和总数count。根据返回的条目显示第一页信息，根据count来显示下方的翻页按钮

## 3.物理结构拓扑是怎么实现的？

物理结构拓扑，总共包括三个部分，平面、节点和链接。平面通过boxGeometry和MeshBasicMaterial来创建，设置boxGeometry的高度为1，对box进行压缩。节点通过SphereGeometry和MeshBasicMaterial来创建。链接通过LineGeometry和LineMaterial来创建。

## 4.如何实现业务的隐藏与显示

通过闭包来实现的。定义一个嵌套函数，外部函数中定义数组，存放上个业务中的节点、链接和平面，然后返回一个内部函数，内部函数中，根据外部数组，来隐藏上个业务所涉及的节点、链接、平面。并展示本次业务涉及到的节点，并且把这些节点存入外部数组中，下一次调用该函数进行清除。

展示部分通过collpase折叠面板来展示多个业务名称。当展开某个业务时，会触发Collpase的onChange函数，onChange函数里面，调用闭包函数，来展示业务

## 5.业务创建、业务删除

业务创建：通过input和radio填写业务详细信息，通过select和连表查询来选择业务经过的网络、节点和链接，给定业务路径后，向后端提交。

业务删除：使用antd的table，设置rowSelection的onChange函数，来获取被选中业务的id，向后台提交这些id，进行删除

## 6.物理网络创建

获取file对象，项目中为input，然后建立一个FileReader实例，调用readAsText方法读取文件。这个方法是异步的，需要绑定FileReader实例的onloadend函数。等到读取文件完毕后，FileReader.readyState转为DONE，之后会调用这个函数。可以在该函数中，通过FileReader.result来获得读取到的文件内容

input的type=’file’，accept=’.json’

## 7.网络类型的切换

通过一个按钮来进行模拟网络和真实网络的切换，向后台发送请求，后台切换数据库。并且将网络类型纳入redux的管理，从而使得其他组件也能获取到当前网络的类型

## 8.项目中哪些地方用到了redux

1.当前的网络类型netType，默认情况下为模拟网，点击切换后切换为真实网，有两个action分别用来切换两个不同的网络。在列表展示网络、网络拓扑呈现、列表展示业务时，会通过redux来获取netType，来让组件进行更新。

2.展示物理网络拓扑的时候，点击节点，出一个小卡片，来显示节点的相关信息。使用redux来保存点击的节点，card组件收到之后，用节点的id去发送请求并展示

## 9.react-Router在项目中是如何体现的呢

通过Link和route配合，来实现点击导航栏，加载对应的组件

## 10.懒加载是怎么实现的？

封装了一个lazyComponent函数，函数内部React.lazy和import导入组件包，然后在Suspense组件中渲染lazy组件。

lazy组件接受一个函数，函数中调用import，返回一个Promise，这个Promise需要resolve一个default export的组件。

Suspense组件的fallback属性，可以指定在组件加载时的显示信息

## 11.项目介绍

目前市场上，各个厂家会针对自己生产的SDN设备，给出相应的SDN控制器软件。这些SDN控制器只能管理自己厂商生产的设备，不能控制别的厂商生产的设备。因此，各个厂商为SDN控制器留出了北向接口，来供用户进行二次开发。我们这个项目就是在SDN控制北向接口的基础上，进行二次开发，实现一个控制系统控制多种类型SDN设备的情景。

项目分为前端、后端和底层，我是负责其中前端开发的部分。主要包括物理网络创建与呈现，业务创建与删除，业务隐藏与展示，网络和业务的列表展示，以及网络类型的切换等功能。

SDN这个项目是我们实验室的项目，他是对SDN设备的管理系统，前端部分主要负责对SDN网络的3d可视化，以及创建网络、创建业务和展示业务等功能

## 12.技术难点及解决方案

1.整个项目加载很慢，因为物理网络拓扑部分节点和链路很多，计算时间较长。对3d显示组件实行了懒加载，使得它在被需要显示的时候才加载。

2.网络拓扑列表显示很慢，整个网络拓扑的信息十分巨大，会导致较长的请求和加载时间。最后采取分页查询的方式，一次查询10个网络，减少等到时间。

## 13.three中的坐标系

three使用的是右手坐标系，即右方为x正方向、上方为y正方向、屏幕里指向屏幕外为z正方向

## 14.3d组件如何实现更新的

通过requestAnimationFrame函数，该函数在你更新跟显示有关的内容时，会调用传入的回调函数来更新页面。

## 15.平面和节点交错，如何做到不遮盖的

设置了构成plane的material的透明度opacity=0.6，可以有一部分透明效果

## 17.事件是怎么定义的

定义了一个vnEvent事件类，该事件类对three中的canvas绑定监听函数。当点击事件发生时，通过点击事件event获取到点击发生的x和y坐标，传入raycaster，来计算得到点击事件发生的节点（Mesh对象）。并将该节点同步到redux中，在card组件上通过节点id去发送请求，来获取该节点的信息。

## 18.raycaster

Raycaster，射线追踪器。用来判断鼠标触碰了canvas中的哪个物体。其基本原理是：从鼠标处发射一条射线，穿透场景的视椎体，通过各种矩阵计算，找出视锥体中哪些对象与射线相交。

## 19.servicemapping如何与three相关联的

遍历每一条业务，对业务所涉及到的链路进行存储，等到展示业务的时候，根据业务id，来将存储的链路进行显示，隐藏业务的时候，将链路也进行隐藏。

## 20.webgl

定义：WebGL是一种 3D 绘图标准，它把 JS和 OpenGL结合在一起，为 HTML5 Canvas 提供硬件 3D 加速渲染。WebGL借助系统显卡在浏览器中展示 3D 场景和模型。

WebGL两个优点：

第一，不需要任何浏览器插件就可以实现3d效果 ;

第二，通过OpenGL接口使用底层硬件进行加速渲染。

坐标系：右手坐标系，同three

步骤：html中建立canvas画布，使用canvas获取webgl绘图上下文，建立顶点着色器生成图元，建立片段着色器对图元进行光栅化处理

基本图元：点、线段、三角形

着色器：1.顶点着色器：计算顶点的位置2.片段着色器：对图元进行光栅化处理

GLSL：着色器使用的语言，为栅格化图形提供计算功能

## 21.webgl和three

webgl是大部分浏览器直接支持的一种3D绘图标准

Three.js以WebGL为基础，对其中的接口进行封装，简化了很多细节，使用起来不再那么繁琐

Three.js在WebGL基础上都为我们做了什么？

1.生成了顶点着色器；

2.生成了片段着色器。

3.自动生成各种矩阵进行运算；

## 22.requestAnimationFrame

requestAnimationFrame是HMML5新增的window对象上的方法，它可以按照屏幕刷新的频率调用传入的回调函数，常用来实现动画效果

requestAnimationFrame和setInterval对比

优点：

CPU节能：使用setInterval实现的动画，当页面被隐藏或最小化时，setInterval仍然在后台执行动画任务。而requestAnimationFrame在页面未激活时，该页面的屏幕刷新任务也会被系统暂停。当页面被激活时，任务会从上次停留的地方继续执行，有效节省了CPU开销。

流畅度：requestAnimationFrame由系统决定回调函数的执行时机。60Hz的刷新频率，每次刷新的间隔中会执行一次回调函数，不会引起丢帧，卡顿。而setInterval任务被放入异步队列，只有当主线程任务执行完后才会执行队列中的任务，因此实际执行时间总是比设定时间要晚；且setInterval的固定时间间隔不一定与屏幕刷新时间相同，会引起丢帧。

函数节流：requestAnimationFrame可保证每个刷新间隔内，函数只被执行一次，这样既能保证流畅性，也能更好的节省函数执行的开销。

## 23.ErrorBoundry

ErrorBoundry可以捕获发生在其子组件树任何位置的 JS错误，并打印这些错误，同时展示降级 UI，而并不会渲染那些发生崩溃的子组件。

它无法捕获异步代码、服务端渲染、和自身产生的错误

ErrorBoundry中需要定义getDerivedStateFromError和componentDidCatch两个方法，getDerivedStateFromError可以渲染备用UI组件，componentDidCatch可以用来显示错误信息

## 24.延迟加载图片

设置图片的loading属性为lazy，无需任何其他的JavaScript代码就可以实现懒加载功能

原理：利用window.innerHeight+document.body.scrollTop>img.offsetTop判断图片已经进入可视区域

# 小程序

## 1.小程序生命周期

onLoad ：监听页面加载，触发时机早于onShow和onReady。在初次被加载时调用，且只调用一次。页面被销毁后重新打开，会触发该函数

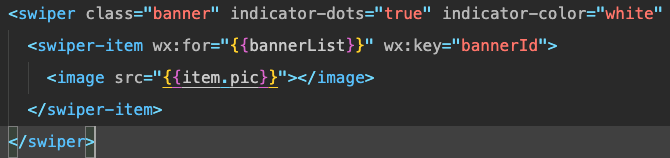
onReady ：监听页面初次渲染完成。onReady在页面没被销毁前只会触发1次，onReady触发时，表示页面已经准备妥当，在逻辑层就可以和视图层进行交互了。

onShow ：监听页面显示，触发事件早于onReady。页面显示之后，Page构造器参数所定义的onShow方法会被调用，一般从别的页面返回到当前页面时，当前页的onShow方法都会被调用。

onHide ：监听页面隐藏。页面不可见时，Page构造器参数所定义的onHide方法会被调用，这种情况会在使用wx.navigateTo切换到其他页面、底部tab切换时触发。

onUnload：监听页面卸载。当前页面使用wx.redirectTo或wx.navigateBack返回到其他页时，当前页面会被微信客户端销毁回收，此时Page构造器参数所定义的onUnload方法会被调用。

## 2.轮播图怎么实现

使用swiper，嵌套swiper-item，swiper-item中使用wx:for来加载不同的页面

## 3.wx:key

为列表中的项目指定唯一的标识符

## 4.scroll-view

可滚动视图区域，通过设置scroll-x来指定横向滚动，scroll-y指定纵向滚动。可以使用view作为子项目

## 5.几种页面跳转方式

1.wx.switchTab(Object object)

跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面

2.wx.reLaunch(Object object)

关闭所有页面，打开到应用内的某个页面

3.wx.redirectTo(Object object)

关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabbar 页面

4.wx.navigateTo(Object object)

保留当前页面，跳转到应用内的某个页面。但是不能跳到 tabbar 页面。使用 wx.navigateBack 可以返回到原页面。小程序中页面栈最多十层

5.wx.navigateBack(Object object)

关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages 获取当前的页面栈，决定需要返回几层

## 6.wx.getStorageSync(string key)

从本地缓存中同步获取指定 key 的内容

## 7.wx.setStorageSync(string key, any data)

将key和value保存在本地缓存中。除非用户主动删除或因存储空间原因被系统清理，否则数据都一直可用

## 8.消息订阅与发布

每日推荐页面，存放歌曲列表。播放页面，根据歌曲id播放歌曲。播放页面切换歌曲，会发布一条消息。每日推荐页面订阅该消息，得出切换后的歌曲id。然后每日推荐页面发布该消息，播放页面收到消息后播放切换后的歌曲

## 9.防抖函数

搜索页面的输入框，如果不设置防抖函数，每输入一个字符都会发送一次请求，造成资源的浪费。设置防抖函数之后，在结束输入之后才会发送一次请求

## 10.节流

登录按钮，设置一个标志位flag，flag为true直接返回，不执行该函数，flag为false则执行，并且把flag置为true。执行完毕之后设置定时器，3s之后flag变为false，可以使得3s内的所有点击事件都被屏蔽掉

## 11.怎么请求后台

通过promise和wx.request封装了查询数据的接口，并指定了success和fail来作为成功和失败的回调函数

## 12.登录怎么实现的

在访问其他页面的时候，会从缓存中获取用户信息，如果缓存中没有，那么就跳转到登录页面，然后登录之后，会从后台得到用户信息，存入缓存。

## 13.歌曲搜索后进入播放页面，点击下一首歌曲会怎么样

歌曲搜索后，跳转到播放组件，会传递一个search=true，播放组件中的下一首和上一首会根据这个search而隐藏，不会让用户点击

## 14.查看历史歌曲是怎么做到的

使用缓存，记录下播放过的歌曲的名称和id，在个人信息页进行展示

## 15.展示搜索历史

使用缓存，记录下搜索过的歌曲的名称和id，在搜索页面进行展示

## 16.微信小程序中的缓存存在多久

永久存在，除非用户主动清空。大小上限为10M

## 17.微信小程序中的事件委托是怎么实现的

歌曲列表点击歌曲播放的时候，歌曲列表父元素用的是scroll-view，子元素用的若干个view，在scroll-view中绑定监听事件，通过event.target来获取到点击的子元素，获取点击到的子元素的id

## 18.event.target和event.currentTarget

event.target指向事件发生的元素，event.currentTarget指向的是绑定该监听事件的元素

## 前端性能优化：

1.css放前面，js放后面

2.defer或者async使得js并行加载

3.懒加载

4.语义化标签

5.图片延迟加载

6.防抖和节流

7.事件委托

## 19.频繁操作DOM时display：none隐藏元素的应用场景

这个写错了，react使用了虚拟dom之后，就不存在这个问题了

## 项目中的难点，以及如何解决的？

回调函数

three

懒加载3d组件