

Web Three.js VPS编辑器需求澄清

VPS:视觉定位系统

Texture: 识别纹理

*注: 上传使用Cos上传, 凭据找我提供。上传类似的业务逻辑在业务部分完成, Three.js编辑用原生js开发, 与业务部分iframe隔离。坐标轴单位为Pixel (像素)。

上传上去的数据基础格式为三个部分:

1. x、y、w、h (控制锚定物位置和大小的参数)
2. 识别纹理 (图片) 的URL
3. 锚定物的类型 (enum: TEXT、PIC、VIDEO、MODEL、ANIMATE_MODEL) 和锚定物的URL

```
1  {
2      "sceneId": "<uuid>",
3      "texture": {
4          "url": "https://host/**.png",
5          "size": {
6              "width": 600,
7              "height": 300
8          }
9      },
10     "anchor": [
11         {
12             "type": "PIC"
13             "position": {
14                 "x": 0,
15                 "y": 0
16             },
17             "size": {
18                 "width": 600,
19                 "height": 300
20             },
21             "url": "https://host/**.png"
22         }, ...
23     },
24     "extra": "{\"id\": \"1\"}" // 被二次序列化之后的JSON字符串
25 }
```

STEP1: 可以上传纹理 (PNG、JPG、WEBP图片) 到Three.js场景编辑器 (3D) 中

STEP2: 整个系统分为两个部分: Scene列表和3D编辑器。

- 刚进入是一个Scene列表 (注: Scene指由一个Texture和若干Anchor组成的识别场景, 最基础的识别单元), Scene列表展示位Cover Grid视图, 右上角有新增Scene按钮。
- 点击新增后到3D编辑器。3D编辑器由左侧SidePanel和中心3DAxis场景组成。
 - 左侧SidePanel称之为“资产管理器”, 资产指VPS识别之后需要相对位置锚定的各类图片、文字、视频、3D模型等资源。

- 中心区域是3D坐标轴AXIS，坐标原点对齐上传的纹理的左上角。
- 用户可以从资产管理器拖动资产相对锚定到纹理。3DScene的原点为Texture左下角的点，3DScene的AXIS为X-Z，坐标轴单元刻度单位为Pixel（像素）。
- 最上方做Toolbar显示保存按钮，点击保存后需要填写标题，然后数据传输到业务层，在业务层弹框显示“保存成功，数据为：xxxxx”。保存时的x、y相对于Texture的左上方点。最终提交的数据为一个Scene的Bean（{}）。
- 保存的每一个Texture和Anchor的Pair对Scene保存到Scene列表。