1. 整体操作逻辑
2. 用户上传所有的全景图片（上传按钮在navbar的elements下的一个upload按钮）
3. 后端保存文件后返回一个json,包括上传文件的地址（数组）和文件名(数组)
4. 将返回的json存到state.fileUrlArr中
5. 前端sidebar中的场景预览监听state.fileUrlArr生成图片列表（src），热点参数区监听state.fileUrlArr，根据数组长度显示target(跳转热点)的数量。
6. 在场景预览中每个图片上绑定事件，点击图片获取图片的src,将src传到state.imgFile中。
7. Editor组件（四种情况：球，圆柱…,目前暂时完成了球），监听imgFile和state.sum（点击保存后生成，包括了各个信息）,从而实现点击图片就会显示图片对应的场景和热点信息。
8. 配置信息（热点信息，热点位置等）在点击了navbar后保存在state.sum中，点击navbar中导出后，根据state.sum的信息生成xml字符串。
9. 将xml字符串传给后端，后端生成xml文件保存。
10. 添加热点步骤及实现过程
11. 跳转热点：

点击elements中的hotspot按钮，触发drawHotspot函数，drawHotspot函数触发state中isHotspot函数，将state.isHotspot的状态取反（false=> true）；

在editor组件中（以equirectangle.vue为例）的鼠标点击事件中，进行if判断，如果state.isHotspot为true,则将鼠标点击坐标存入positionArr中，当数组的length达到3时，利用three.js的sprite技术绘制热点，并将positionArr清空，将state.isHotspot取反（退出绘制）；

如果state.isHotspot为false,则实现点击热点（选中）后将其删除；

跳转热点在xml中的信息主要是url（跳转到哪个场景中），pan,tilt(热点位置信息)，target（暂时没找到数据来源，暂时全写死为0/0/0）。

Sidebar中的热点信息中的target与hotTarget计算属性绑定（v-model），hotTarget的set函数触发state的saveTarget函数，将选中的target存到state.target数组中（push）;

绘制结束后点击navbar的确定按钮后，触发submitInformation函数，submitinformation函数触发state中isSubmit函数，将state.isSubmit状态取反，

在editor组件中监听state.isSubmit，如果isSubmit的状态发生改变就将scene中所有的跳转热点保存到sprites数组中，数组的每一项是一个对象，代表着一个跳转热点，对象属性包括了panAndTilt(value为数组)，position（value为数组，target(value为state.target中的一项（按照添加的顺序）)），然后将这个sprites数组保存到state.hotSpots中；

注意：当切换场景时，触发两个事件，一个是清空state.target，一个是清空state.hotSpots，确保每个场景热点信息的独立性。

点击保存，触发saveInformation函数，saveInformation函数触发state中的saveInformation函数，将该场景中的所有信息作为state.sum数组的一项保存。

保存后，当用户切换场景（点击不同的预览图）后回到原来的场景时，取state.sum中该场景的hotSpots信息，根据hotSpots信息绘制之前保存的跳转热点。

生成xml字符串时，第i个场景中遍历sum[i].hotSpots数组，第j个热点的url= sum[i].hotSpots[j].target,热点的pan=sum[i].hotSpots[j].panAndTilt[0],

tilt=sum[i].hotSpots[j].panAndTilt[1];

1. 图片热点：

点击elements中的ImageHotspot按钮，触发drawImageHotspot函数，drawImageHotspot函数触发state中isImageHotspot函数，将state.isImageHotspot的状态取反（false=> true）；

在editor组件中（以equirectangle.vue为例）的鼠标点击事件中，进行if判断，如果state.isImageHotspot为true,则将鼠标点击坐标存入positionArr中，当positionArr的length为不同值时绘制不同的几何体（点，线，面），并将positionArr清空，将state.isImageHotspot取反（退出绘制）；

如果state.isImageHotspot为false,则实现点击热点（选中）后将其删除；

图片热点分成两种情况，一是点击显示图片，二是点击跳转到某个网页

**显示图片：**

显示图片图片热点在xml中的信息主要是target（点击后显示什么照片），pan,tilt(热点位置信息)。

未实现：Sidebar中的热点信息中的image上传图片按钮计划上传图片后后端返回一个地址，将返回的地址保存到state中某个数组变量中（类似于state.target）.

绘制结束后点击navbar的确定按钮后，触发submitInformation函数，submitinformation函数触发state中isSubmit函数，将state.isSubmit状态取反，

在editor组件中监听state.isSubmit，如果isSubmit的状态发生改变就将scene中所有的图片热点保存到imgHotSpots数组中，数组的每一项是一个对象，代表着一个图片热点，对象属性包括了panAndTilt(value为数组)，position（value为数组，target(value为上传的图片的url)），hotUrl(value为null)然后将这个imgHotSpots数组保存到state.imgHotSpots中；

点击保存，触发saveInformation函数，saveInformation函数触发state中的saveInformation函数，将该场景中的所有信息作为state.sum数组的一项保存。

保存后，当用户切换场景（点击不同的预览图）后回到原来的场景时，取state.sum中该场景的imgHotspots信息，根据imgHotspots信息绘制之前保存的跳转热点。

**打开外部链接：**

打开外部图片热点在xml中的信息主要是url（点击后跳转到哪个网站），pan,tilt(热点位置信息)。

Sidebar中的热点参数中的url是个input输入框，输入后按回车将输入的地址保存到state.hotUrl中（类似于state.target）.

绘制结束后点击navbar的确定按钮后，触发submitInformation函数，submitinformation函数触发state中isSubmit函数，将state.isSubmit状态取反，

在editor组件中监听state.isSubmit，如果isSubmit的状态发生改变就将scene中所有的图片热点保存到imgHotSpots数组中，数组的每一项是一个对象，代表着一个图片热点，对象属性包括了panAndTilt(value为数组)，position（value为数组，hotUrl(value为用户输入的url)），target(value为null)然后将这个imgHotSpots数组保存到state.imgHotSpots中；

注意：当切换场景时，触发两个事件，一个是清空state.target，一个是清空state.hotSpots，确保每个场景热点信息的独立性。

点击保存，触发saveInformation函数，saveInformation函数触发state中的saveInformation函数，将该场景中的所有信息作为state.sum数组的一项保存。

保存后，当用户切换场景（点击不同的预览图）后回到原来的场景时，取state.sum中该场景的imgHotspots信息，根据imgHotspots信息绘制之前保存的跳转热点。

将上述两种情况合并处理：

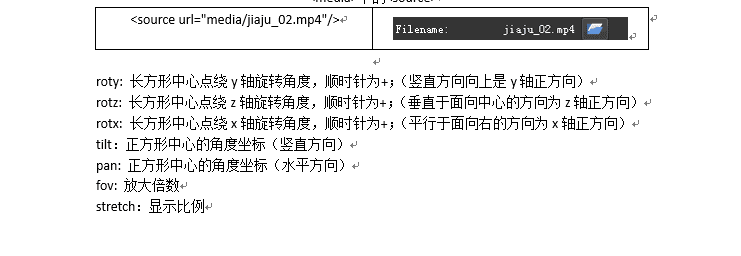
生成xml字符串时，第i个场景中遍历sum[i].imgHotSpots数组，第k个热点的url= sum[i].imgHotSpots[k].hotUrl?sum[i].imgHotSpots[k].hotUrl:’’

热点的pan=sum[i].hotSpots[k].panAndTilt[0],

tilt=sum[i].hotSpots[k].panAndTilt[1];

target=sum[i].imgHotSpots[k].target?sum[i].imgHotSpots[k].target:’\_blank

1. 视频热点：



’

6.11更新

1.首先绘制热点（跳转热点或者图像热点），绘制完成后，将热点的位置信息保存在state.tempLocation中，选中这个热点，热点参数区显示需要输入的参数。

对于跳转热点，输入目标场景（hotTarget）和场景说明(hotIntroduction),

点击确定，将state.tempLocation，hotTarget,hotIntroduction作为一个对象保存（push）在state.jumpHotSpots数组中。

点击保存场景，将state.isSubmit改变，获取场景中存在的热点（state.hotSpots）,将state.jumpHotSpots中与state.hotSpots中坐标相同的取出来，重新生成state.jumpHotSpots,最后生成XML时利用state.jumpHotSpots.即可。

对于URL热点，输入URL，点击确定，将state.tempLocation,hotUrl作为一个对象保存（push）在state.urlHotSpots数组中。

点击保存场景，将state.isSubmit改变，获取场景中存在的imgHotSpots,将state.urlHotSpots中与state.imgHotSpots中坐标相同的取出来，重新生成state.urlHotSpots保存进sum中，最后生成XML时利用state.urlHotSpots即可。