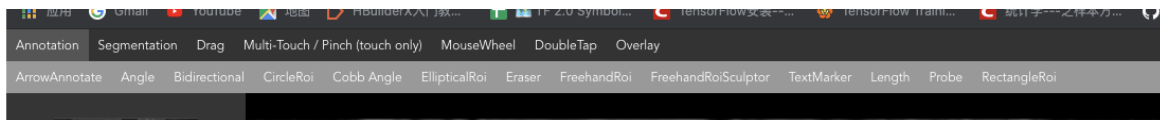


刷子类型工具的数据存储



第一种svg类型的工具

_applyStrategy():mouseup的时候触发，就是在这里画图的

svg工具的数据存储在toolstate里面 全局都可以访问到

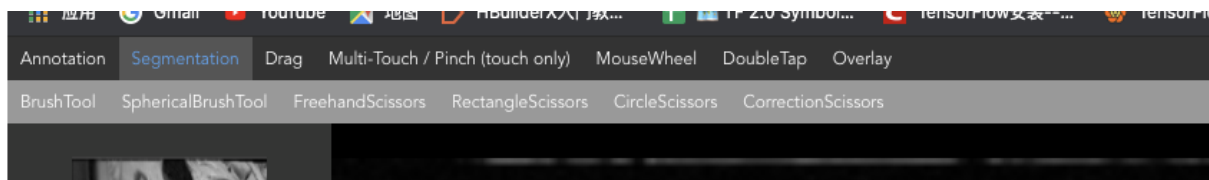
```
saveToolStateButton.addEventListener('click', event => {
  //保存数据前进行数据清空
  toolStates.splice(0, toolStates.length);
  Array.from(toolNames).forEach(toolname => {
    console.log("当前保存的工具数据为: ", toolname);

    //添加单种工具数据
    //根据工具名称获取工具数据并转成json类型
    const toolStateString = JSON.stringify(cornerstoneTools.getToolState(element, toolname));
    if (toolStateString) {
      let toolState = JSON.parse(toolStateString);
      // alert("saved");
      //单种数据保存到toolStates当中
      let tooldata = {tooltype: toolname, toolstate: toolState};
      toolStates.push(tooldata);
      console.log("saved data", toolStates);
    } else {
      // alert("nothing to save!");
    }
  });
});
```

目前可以获取的有这几种工具

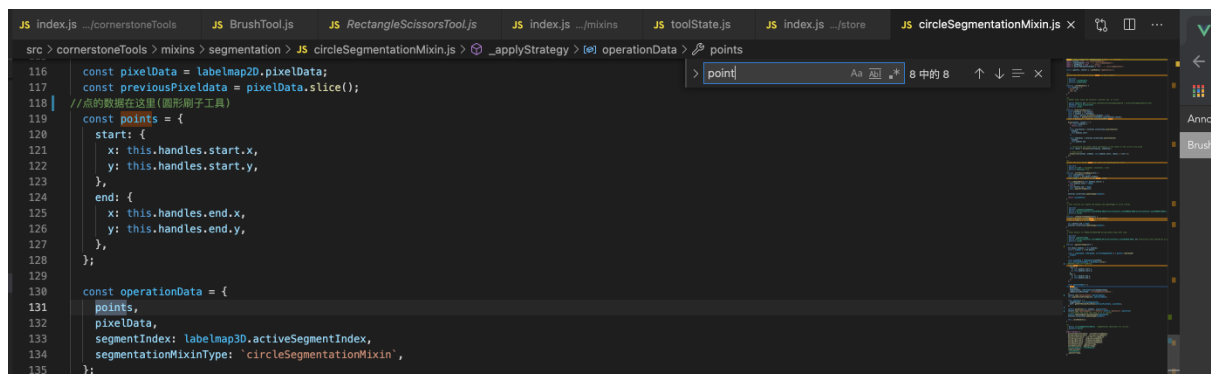
```
const loadToolStateButton = document.getElementById('loadToolState');
let toolNames = ['FreehandRoi', 'CircleRoi', 'EllipticalRoi', 'RectangleRoi'];
```

而对于以下的刷子工具



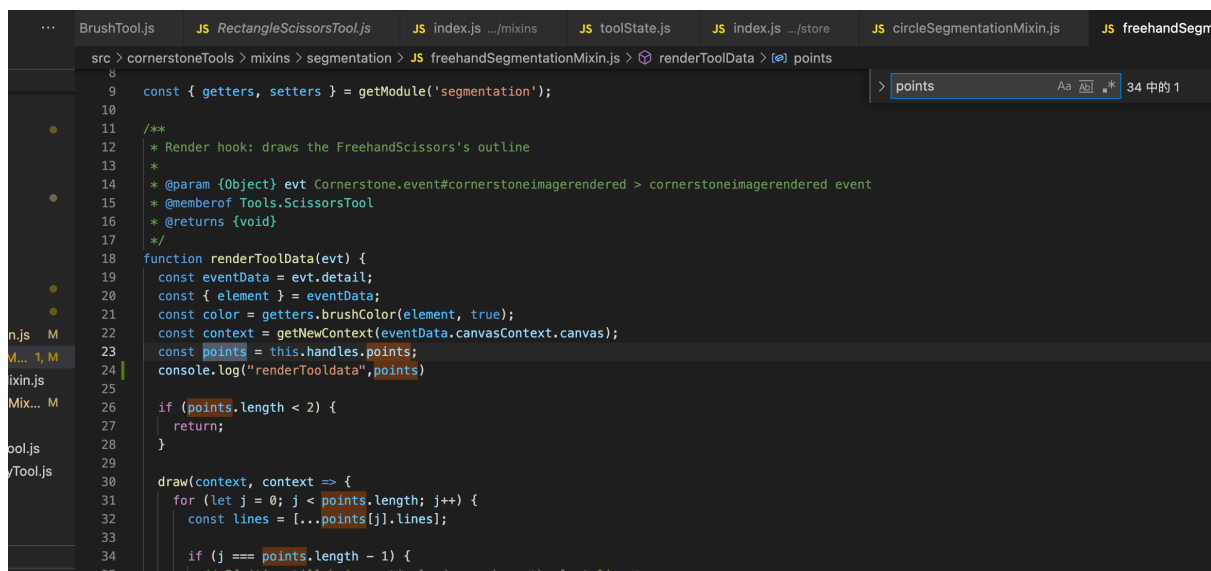
圆形

参数保存在points



多边形

points 保存在this.handles参数里面



长方形

也在this.handles里面

```
Tool.js JS index.js .../mixins JS toolState.js JS index.js .../store JS circleSegmentationMixin.js JS freehandSegmentationMixin.js JS rectangleSegmentationMixin.js x cems-ms
src > cornerstoneTools > mixins > segmentation > JS rectangleSegmentationMixin.js > _applyStrategy
68
69   } = evt.detail;
70   this.handles.end = image;
71   external.cornerstone.updateImage(element);
72 }
73
74 /**
75  * Event handler for MOUSE_UP/TOUCH_END during handle drag event loop.
76  *
77  * @private
78  * @method _applyStrategy
79  * @param {(CornerstoneTools.event#MOUSE_UP|CornerstoneTools.event#TOUCH_END)} evt Interaction event emitted by an enabledElement
80  * @returns {void}
81  */
82 function _applyStrategy(evt) {
83   // console.log("_applyStrategy(evt)")
84   evt.detail.handles = this.handles;
85   console.log("长方形刷子工具数据", this.handles)
86 }
```

刷子工具

pointsArray

```
JS BrushTool.js x JS index.js .../mixins JS toolState.js JS index.js .../store JS circleSegmentationMixin.js JS renderBrushMixin.js
src > cornerstoneTools > tools > segmentation > JS BrushTool.js > BrushTool > _paint
38
39 */
40 _paint(evt) {
41   const { configuration } = segmentationModule;
42   const eventData = evt.detail;
43   const element = eventData.element;
44   const { rows, columns } = eventData.image;
45   const { x, y } = eventData.currentPoints.image;
46
47   if (x < 0 || x > columns || y < 0 || y > rows) {
48     return;
49   }
50
51   const radius = configuration.radius;
52   const pointerArray = getCircle(radius, rows, columns, x, y);
53   console.log("brushTool pointerArray", pointerArray)
54   // console.log("brushTool eventData", evt.detail, "pointArray", pointerArray)
55   const { labelmap2D, labelmap3D, shouldErase } = this.paintEventData;
56
57   // Draw / Erase the active color.
58   drawBrushPixels(
59     pointerArray,
60     labelmap2D.pixelData,
61     labelmap3D.activeSegmentIndex,
62     columns,
63     shouldErase
64   )
65 }
```

问题就是这些数据都保存在工具的内部，不好拿出来

在处理svg类型工具时 只要往toolsstate里面添加数据 再updateimage一下,就可以加载图片了

```

525 });
526
527 loadToolStateButton.addEventListener('click', event => {
528   Array.from(toolStates).forEach(tooldata => {
529     console.log("遍历查看当前工具的数据", tooldata);
530     if (tooldata) {
531       cornerstoneTools.clearToolState(element, tooldata.tooltype);
532       console.log("load ", tooldata.toolstate);
533       tooldata.toolstate.data.forEach((data) => {
534         cornerstoneTools.addToolState(element, tooldata.tooltype, data);
535       })
536       window.cornerstone.updateImage(element);
537       // alert('loaded');
538     } else {
539       alert('Nothing has been saved!');
540     }
541   });
542
543 });

```

这个目录下有一个刷子工具的demo，我没有看到类似这样的接口，感觉这里的刷子工具都是一层一层的，好像都在cornerstone内部，刷子工具的数据不知道是不是有一个全部可以访问的存储的地方。

