软装大师2D搭配：为装修设计、家具搭配提供操作的应用，主要是面向对家具装修设计有兴趣爱好的人群。

主要结构：



类文件结构：

每个模块都对应了一个模块类，在画布主体区中每个子元素都封装成了Product类，Product类本身具有可选择、放大和和缩小、旋转、拖动的功能，画布主体区中每个子元素都是一个独立的个体。库面板中需要数据加载，数据加载对应两个类，DataLoader和SubDataLoader，后者是前者的子类，具有分页的功能。菜单栏的弹出框和画布主体区“输入文字内容”的输入框以及工具栏中显示工具名字的功能都封装成了组件，这样做的目的是为了使index页面更具结构性。在画布主体区中对应有两个单独的模块：相关产品推荐（relProc）和相关图片推荐（relImg）。

几个重要的问题：

1.库面板中数据加载分页功能的实现？（库面板中有多个tab栏下面都具有分页的功能）  
问题：第一次点击tab栏时加载数据（当数据更新时也重新加载），只返回第一页数据和总的页数，初始化库面板列表数据和页码。只有在点击了2、3...页码时，才加载对应页面的数据。问题是：页面画布区是可全屏化的且浏览器窗口大小也是可以调整了，库面板的高度是可动态调整的，所以每页数据的个数是不一定的。

解决方法：第一次点击tab栏时加载所有的数据，获取当前库面板的高度，进而求出页码数，SubDataLoader文件中保存两个变量：一个是所有的数据totalData，一个是当前页码对应的数据context。没点击页码，我就更新context值，并重新编译。给页面注册点击事件，是找到当前tab栏内容下面的页码添加点击事件，只在第一次数据加载的时候注册点击事件，所以每个页码（包括上下页）的点击事件是不会重复注册的。

2.Product类中主要有四个方法：点击或者按下时被选中，图片四周被选中；只有在选中的情况下，才可以拖动；鼠标放在顶部控制点可旋转；鼠标在其它控制点上可放大和缩小。移动和旋转都是使用了transform3D变化，这个属性是一个累加的过程，每次赋值都会覆盖前次的赋值。

问题：每次移动后或者旋转后记录移动的距离（x,y）和旋转角度angle,使用自定义属性保存这三个变量。问题是这样做很麻烦，每次旋转前后移动前都得获取这三个自定义属性值，而且这样做导致计算画布每个子元素中心点很麻烦（整体放大和缩小的时候会使用到中心点）。

解决方法：每次移动后，在鼠标松开那一刻将移动的距离（x,y）累加left和top上，让transform3d(0,0,0)。每次旋转角度仍然使用自定义属性保存和获取。

3.Product类中有一个放大和缩小的方法，画布子元素四周有8个控制，移动八个控制点都可放大和缩小，但是中心点是不同的。比如左上角的控制点，子元素是以右下角为中心点进行放大缩小的。

问题：tranform中有个属性scale对物体进行放大和缩小，transform-origin可改变物体进行transform变换的中心点，默认值是物体的中心(center center)，如果要改变可以这样写tranform-origin: 100px 100px,表示以物体最左上角（0，0）分别向左和向右走100px。虽然可以改变中心点，但是会产生一个很大的问题，就是同rotate和translate3d一样都是属于transform一个属性，会累加会覆盖，必须也要用自定义属性记录这个值；而且缩放是会改变left和top值的，所以还得用自定义属性保存最初始的left和top的值。计算麻烦而且容易出错。

解决方法：模拟scale属性，放大和缩小无非就是改变了元素的四个属性：width, height, left, top。而且是按比如放大和缩小的。以最左上角为例，它的中心点在右下角，那么向上移动控制时，宽度增加，求出比例，进而求出高度。宽高的变化值就知道了。

4.工具栏有个成组和解组的功能？

成组的方法：新建一个li,新建一个ul,让ul插在li下面，成为二级元素，让需要成组的元素li插入ul中。新建li宽高和位置计算：遍历子级每个li 的每个控制点，找到最左边、最右边、最上边、最下边的控制点，进而就可以求出成组后的元素的width、height、top和left值，又能保证每个二级子元素li的在屏幕上的位置和旋转角度不便。而解组的实现恰好相反，它是遍历成组元素的每个子元素li,把它重新插到画布父级ul下，并把原来的成组元素li删掉。包括c元素和已经成组的AB再次成组，也是现将AB解组，再将A,B,C元素成组。

增加成组和解组功能后产生的问题：如果子级有一个旋转角度angleEle，此时父级也有一个旋转角度angle,那么解组后，要保持子级在屏幕上的位置不变，要做哪些变化？问题产生的原因：每个元素li旋转都是以自身中心为中心点旋转，但是当多个元素成组后，每个子元素是以整体的中心为中心点进行旋转的。成组前和成组后，子级元素li旋转的角度是相同的，只是成组后父级旋转，子级li跟着旋转，旋转的角度相同，当是相对成组前来说它的left和top值发生了变化，这个变化值是多少？

解决方法：根据每个子元素中心点移动的距离，旋转后，每个子级元素li中心点移动的距离，就是解组后每个子级元素left和top应该变化了的值。每个子级元素li中心点到成组父级元素中心点的距离是不会变的，旋转角度也已知。知道两条和它们的夹角，就可以求出第三条边，第三条边正是每个子元素旋转后中心点移动的距离。

产生的问题二：成组元素后的放大和缩小问题。画布里每个元素li的宽高是不确定的，它是根据从库面板中拖过来的图片的大小生成的li，所以当元素li大小变化是，下面的元素不是跟着一起动态调整的。成组后的元素li下面对应有多个图片（多个子级Li）,每个子级li又是怎么跟着一起变化？

解决方法：子级li的宽高增量可以根据父级li的宽度增量\*子级的宽高比上父级的宽高，但是子级left和top值又怎么变呢？因为子级的left值是相对于父级的宽度的，最大的值也就是父级的宽度，所以那个比值就是子级的left和top比上父级的宽高。

成组是个很多的问题，除了产生这三个问题外，还会导致工具中复制、水平镜像和垂直镜像以及下边的整体放大和缩小都收到了组成的影响。下面说道它们再介绍。

5.工具栏中的上一层、下一层、底层、顶层，实现它们的方法很简单，但是有个关键点是要保证画布里的每个元素li的zIndex都是唯一的。所以我在cvsSet模块中定义了一个公开的变量zIndex，每当我拖拽一个图片到画布中时，我就让zIndex+1,以及复制操作也会涉及到zIndex的改变。如果复制的是成组元素，那么我遍历成组元素的每个Li，让它们按zIndex排序，并依次给它们复制cvsSet.zIndex++，这样解组后就能保证画布里每个元素li的zIndex都是唯一。

6.镜像翻转，以水平翻转为例，我是直接改变.img的transform值，每次翻转都增加180度。如果是成组元素，同样是中心点的位置，每次翻转不应该以自己图片的中心为中心点，而应该是以成组父级li的中心为中心。

解决方法：我不改变中心点，因为元素是可以移动的，所以中心点也一直再变，所以如果是成组元素我改变的就不是.img的transform值，而是成组元素下面的ul的transform值，因为它的宽高都是和成组元素li的宽高相同的。如果成组元素发生了镜像翻转，解组又怎么保持子级元素的位置和角度都不变呢。

解决方法：父级li元素发生了镜像翻转，子级元素li也是反正了同样的镜像翻转，除此之外，它相对于子级的left和top值肯定发生了变化，如果是水平翻转，就相当于从左边到了右边。

怎么求left和top值的变化？每个子级元素都是相对于父级的中心线翻转的，left和top值就是子级的中心点到父级中心点距离的两倍。

7.工具栏左下角有个用于控制画布整体放大和缩小的小方块。要实现以下的要求：（1）当我移动这个小滑块的时候，是画布所有元素一起放大和缩小的；（2）放大和缩小这一过程是以画布屏幕中心为中心点的；（3）新拖过来的图片，如果画布此时是缩小到.8，那么新加入的图片一开始就要缩小到.8，而且始终要以鼠标松开那个点为新建li的中心点；（4）整体放大缩小后，画布的每个元素仍然具有自己的属性（可移动、旋转、放大和缩小），而且它们的相对位置是不变的；（5）放大缩小后，画布始终是铺满整个屏幕的。

基于以上要求，要实现这个功能，不能在画布ul上使用scale的样式。原因：中心点的问题（每个li拖过来的位置是不一定的，得计算每个li的transform-origin，但是每个元素li本身也是可以旋转的，所以不应该改变transform-origin的值；如果使用scale对画布进行放大和缩小，显然画布不能保证始终铺满整个屏幕。

解决方法：这里仍然是通过改变width, height, left 和top的值来模拟是scale,与单个元素放大和缩小不同的是，一旦移动了小滑块，那么画布里的所有元素width, height, left 和top都得跟着变，而且它们都是以屏幕中心为中心点进行变化。移动小滑块，可以得到画布应该缩放的比例cvsSet.scale,每个li元素的宽高变化乘以这个比例就可以了，但是left和top又改变了多少呢？实际上left和top的变化值就是每个元素自身的中心点到画布的中心点的差值，乘以那个比例的变化值，也就是1-cvsSet.scale。因为cvsSet.scale这个值是累加变化的，就是说第一缩小到.8，第二次缩小到.7，从第一次到二次值变化实际上是0.1,但是计算的时候是.7，所以必须定义一个变量pos来保存元素li的宽高、left和top、中心点的位置，在加入元素或者复制元素或者成组元素时，得初始化这个自定义属性pos,而在每次移动或者自身放大和缩小时（凡是涉及到中心点改变的操作）都得更新这个pos自定义属性。如果在画布已经处于放大或者缩小的状态下，新拖拽图片到画布中，宽高仍然是按那个比例变化，但是要始终保证新建的li的中心点为鼠标松开的那个点为中心点，所以还需要在pos中保存另一个值scale，复制为当前cvsSet.scale，这样后来加入的元素，再次缩放时，是变化0.1，而其它元素仍然是变化了0.7。这样做的另一个好处是，即使我每次移动元素，更新scale,相当于从scale开始缩放，这样就不会出现一闪的现象。如果是成组元素，那么每个成组下面的子元素都按照父级的那个scale值去缩放就可以了。

9.撤销操作？

使用sessionStorage来保存每次画布ul 下li发生操作后，ul的html()值，给每个画布ul设置一个唯一的id号，并自定义一个版本号的属性，当前画布li元素操作后，就sessionStorage保存ul的内容，key值就是ul的id号+版本号，然后让版本号加1。撤销时获取当前画布ul的id值好版本号，得到key值，取出ul的内容，重新插入ul下。