## Navigator窗口的实现思路

### 方案一：

Navigator是软件的第二视图窗口，如下图左边Navigator窗口，在中心窗口显示矩形框选的部分

**功能需求：**

功能1：将整幅图以合适的大小完全显示在Navigator窗口

功能2：刚开始打开gds文件的时候，矩形初始大小会包含整幅图，从而使完整图在中心窗口显示

功能3：矩形会获取鼠标滚轮事件和鼠标移动事件，随鼠标滚动量的大小而不断变化，随鼠标移动而平移

功能4、当矩形大小不变，移动矩形，在中心窗口实时显示矩形框选的部分

功能5：当矩形大小不断变化，改变的是缩放比例，（即在整个中心窗口显示的一直是矩形框选的部分。当矩形变小，框选的部分很小，在中心窗口显示的是放大的效果；当矩形变大，框选的部分很大，在中心窗口显示的是缩小的效果）

功能6：在中心窗口实现缩放，对应Navigator窗口中矩形大小和位置相应也会变化。即：中心窗口与Navigator窗口同步

**思路：**

功能1解决思路：

在Navigator 窗口：用scene-view框架显示整幅图，scene显示整幅图且大小不变，怎样以合适窗口显示？（gds合适窗口的显示）

功能2解决思路：

view画矩形，矩形的大小需要根据scene中整幅图的大小确定，如何确定整幅图的大小？

功能3解决思路：

难点：目前在view还没有实现矩形随鼠标滚轮的大小变化而变化，鼠标的移动而平移

功能4解决思路：

前提：Navigator中整幅图大小已经确定，矩形大小已经确定，这样缩放比例已经确定（矩形大小/整幅图大小）

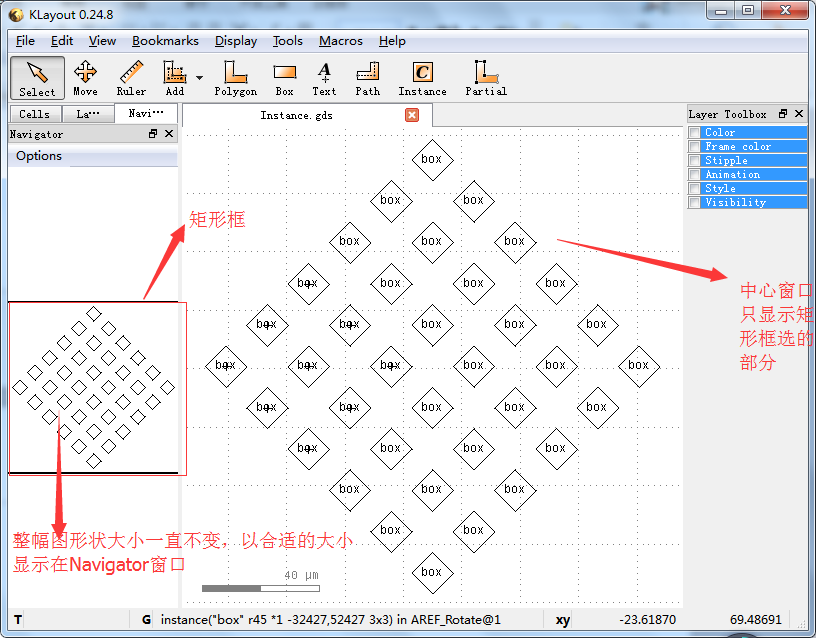
矩形的坐标(view坐标映射到scene坐标)，根据矩形坐标（scene坐标）和矩形大小，确定矩形框选的部分，使中心窗口显示对应的部分。

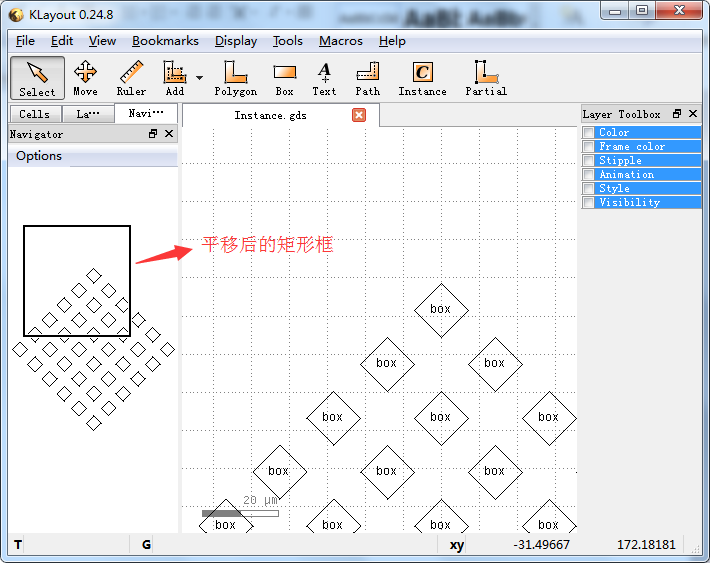
难点：在主窗口中显示部分图形。需要研究scene-view框架，看view是否提供可以显示scene指定的部分

功能5解决思路：

难点：确定矩形大小变化的规律

**效果图：**





方案二:

读懂klayout源代码，直接使用klayout源代码对应Navigator功能。

已经找到Navigator主要源代码，由于涉及的文件太多，完全理解它的思路很困难，需要一段时间

资料查找问题：按理说navigator是画图软件一个很常见的功能，我搜了百度，一些博客，网站，没有找到相关的，目前我能借鉴的资料只有klayout源代码，只是klayout阅读起来比较困难

希望刘博士和刘导可以给一些指导，提供解决这个问题更好的方案。

不管是你的方案还是klayout的基本上是相同的，直接采用你的方案就可以了。这里面主要涉及到Navigator和scene-view视图的关联。该方案的实现很大程度与缩放方案相关。

Navigator和scene-view视图的关联功能描述。

Navigator和scene-view视图的关联与cell widge 与view的关联类似，在Navigator里面对于鼠标缩放和平移事件捕捉后，计算当前的捕捉矩形大小后转换成要显示的图形区域范围。然后再把该事件转发给相应的scene-view(也可以用自己定义的信号)，scene-view接收到该信号后根据要显示的图形区域显示图形。

实现的一些必要步骤

1. 在每次打开一个图形文件时，必须计算出该图形的区域范围。如果某些格式的文件中已经包含了区域范围，则只需要读取就可以了(大部分格式的文件应该包含了)。否则需要遍历整个场景计算出图形区域范围。这一步我想应该已经实现了，因为在scene-view显示的场景的时候应该使用了图形区域范围，否则不方便在加载文件时显示整个场景的。

**华丽合联(1816023692) 5/9/2017 10:42:43 PM都需要读取GDSII文件的主构元（最高层）以及其属下调用的所有从构元的边框（Frame）。**

**可以定义每个构元的Frame为所有坐标的（Xmin，Ymin）至（Xmax，Ymax）的Box（矩形框界）。注意搜索/计算每个构元的Frame时，一定有办法只用O(N)的计算量级的，千万不要用二阶的O(N^2)。**

**即每个子构元算出Frame，母构元根据子构元的Frame和附加基本图形占用的坐标可以算出扩大的Frame（边框）。逐次上推。最上层的主构元的Frame即是合适窗口显示的尺寸，以及整幅图的大小。**

**华丽合联(1816023692) 5/9/2017 10:47:07 PM**

**这个Frame计算，可以定义一个子函数，同时可以加在自定义的类体里面做为记录。在我们以后的扩充文件格式里面，一定要把Frame加进去的。Frame同时对以后的Boolean运算也非常有用。**

**一定要用定义Frame的方法去由下往上类推。**

**当构元显示层级不够深的时候，超过层级的构元不用显示具体图形，但是需要显示该构元的Frame和Name。**

1. 在scene数据结构中必须添加一个图形显示区域的矩形数据结构(Display Rect)，初始化为整个图形区域大小。如果Display Rect在缩放功能的实现中已经拥有并且跟着scene的缩放而变化，则直接使用该数据。
2. 打开Navigator时，参考scene-view的显示，居中显示出整个图形区域。整个图形显示区域具体占Navigator窗口多大范围，这个可以根据测试效果调整，可以从Navigator窗口90%开始。打开Navigator窗口是，主程序建立或者修改Navigator与scene-view的映射。然后记住这个显示的图形区域对应的客户区的坐标，这个区域与整个场景图象的显示区域一一对应。
3. 如果是新打开文件是已经存在Navigator窗口时，则Navigator窗口中的选择矩形包含整个区域图形。如果打开Navigator窗口时已经存在了scene-view，则根据当前的激活的scene-view的显示范围Display Rect和场景图形区域计算出缩放比例(可参考缩放方案)，然后按照Display Rect中心点对应着Navigator窗口的中心点的原则，根据缩放比例计算出Display Rect区域在Navigator窗口对应的矩形坐标，然后画出矩形。
4. 当鼠标缩放时，计算出缩放比例，根据缩放比例画出导航矩形的大小(计算缩放比例的方案参考之前的缩放方案)。然后给相应的scene view发送信号要求scene view更新图形显示。这个scene view信号槽实现可以利用原有的鼠标滚动平移信号处理，也可以单独。
5. 当鼠标移动时，根据鼠标的移动量计算出新的导航矩形的中心点，然后重新绘制。然后给相应的scene view发送信号要求scene view更新图形显示。
6. 当view切换不同的scene view时，则需要根据相应的显示区域重新绘制Navigator窗口中的图形。