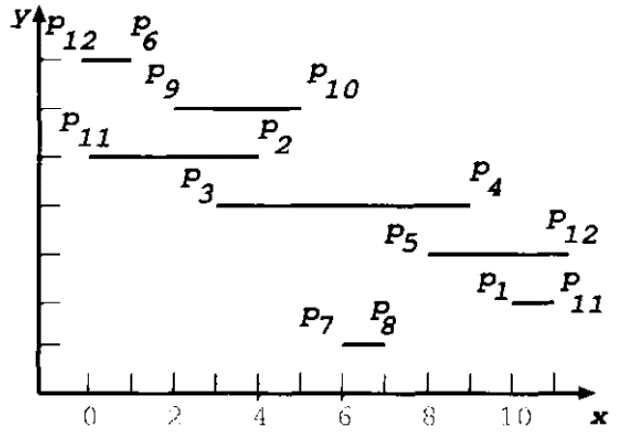
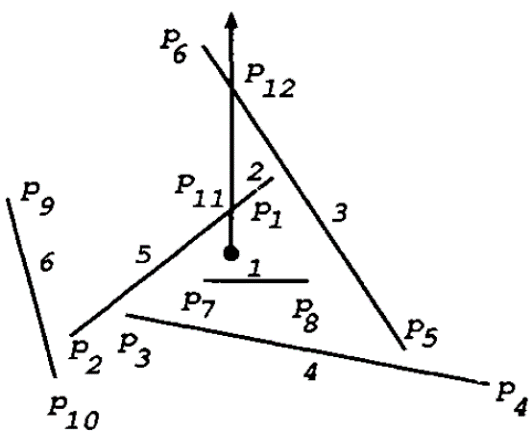
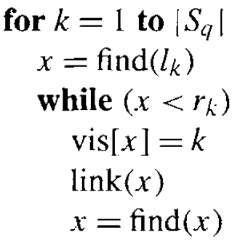
线性集合并计算可视多边形

计算可见多边形的算法利用了之前计算出的顶点极角序和边的偏序关系，其将多边形顶点的极角序转化为x坐标（如下左图中P11和P12两点均在竖直的射线上，其x坐标均为0；之后顶点的极角顺序为P6、P9、P3、P2依次向后，因此其x坐标分别为1、2、3、4等等），并将多边形中的边关于输入点的偏序关系转化为排序的序号值并进一步转化为y坐标（P7P8=1、P1P11=2、P5P12=3以此类推），从而将原先如下左图中的求可见多边形问题转化为了右图中对于每个x坐标求其对应的y值最小的线段。这种转化保证了原先图形中线段的遮挡关系，即每个区间对应的最低的线段一定是原图中输入点可见的线段。



进行转化后可以使用Gabow[3]中的线性集合并算法计算，在线性时间内得出点q的可见多边形。算法的伪代码如下所示：

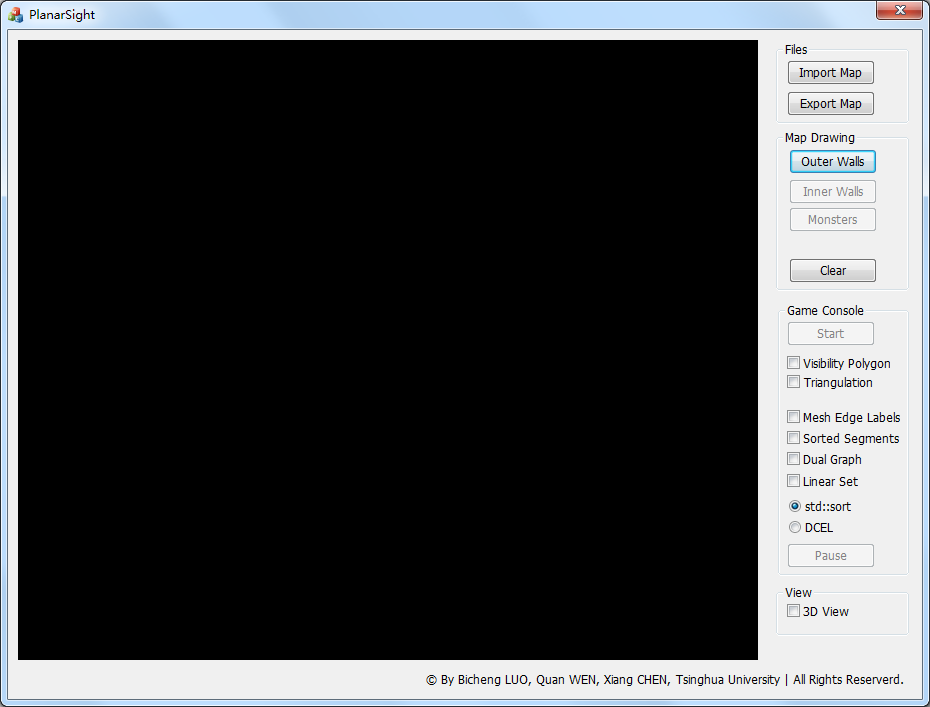


其本质为自底向上遍历每一条线段，为每一个x值标记其所对应的最低的线段的y值，并记录在vis数组中，最后将这些线段依次连在一起便得到了可见多边形。算法中涉及两个函数，find函数返回包含x的集合中的最大元素，link函数将包含x与包含x+1的集合合并起来。对于上面的例子，算法如下执行操作：首先遍历到y值为1和2的线段，即P7P8和P1P11，根据后边的link操作相应的x值6、7和10、11会分别组成集合，同时vis[6]=1，vis[10]=2；下一个循环中遍历到y值为3的线段P5P12，此时8和9会先组成集合，同时vis[8]和vis[9]会记为3，之后在while循环中x=9时link(9)操作会将集合{8, 9}与集合{10, 11}合并起来，再执行find时返回11会达到线段的右边界，循环也就终止了，这样vis[10]便保留了之前的值，即已组成集合（vis已被标记）的x值不会再次被访问，也就保证最后求出正确的线段y值。

上边的算法可以求出一个点关于多边形的整个可见区域，而我们的程序中需要求出某一视角内的可见多边形。因此我们根据视角的左右边界计算出其所在的x区间，通过输入点和两个边界求出其与所在x区间对应的最小y值线段的两个交点，之后按vis数组中的值依次连接相应线段便可得到一定视角范围内的可见多边形，这个过程同样可在线性时间内完成。

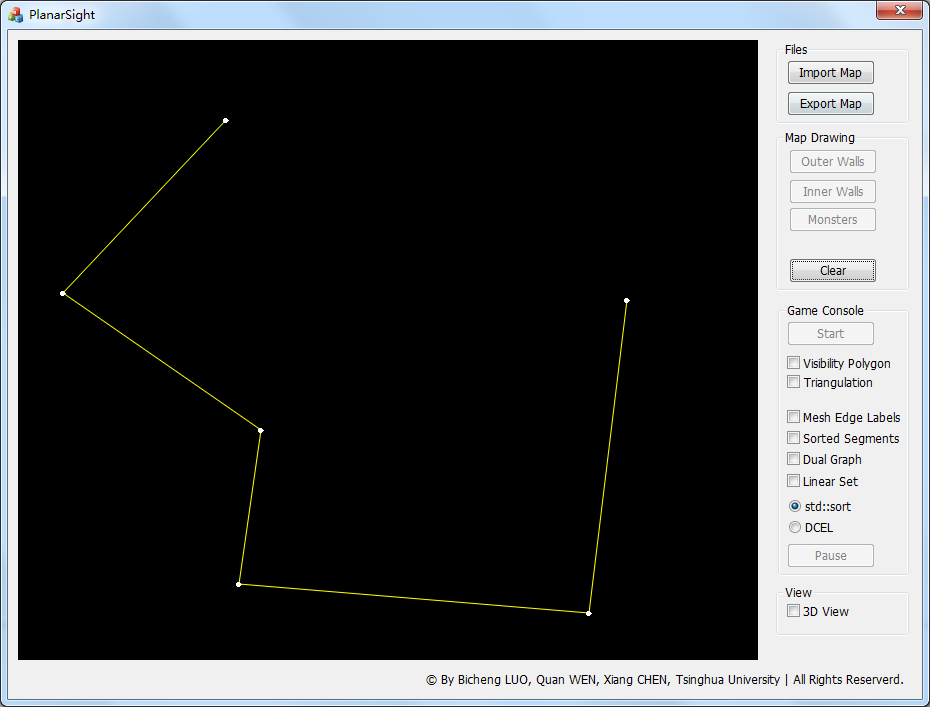
用户手册

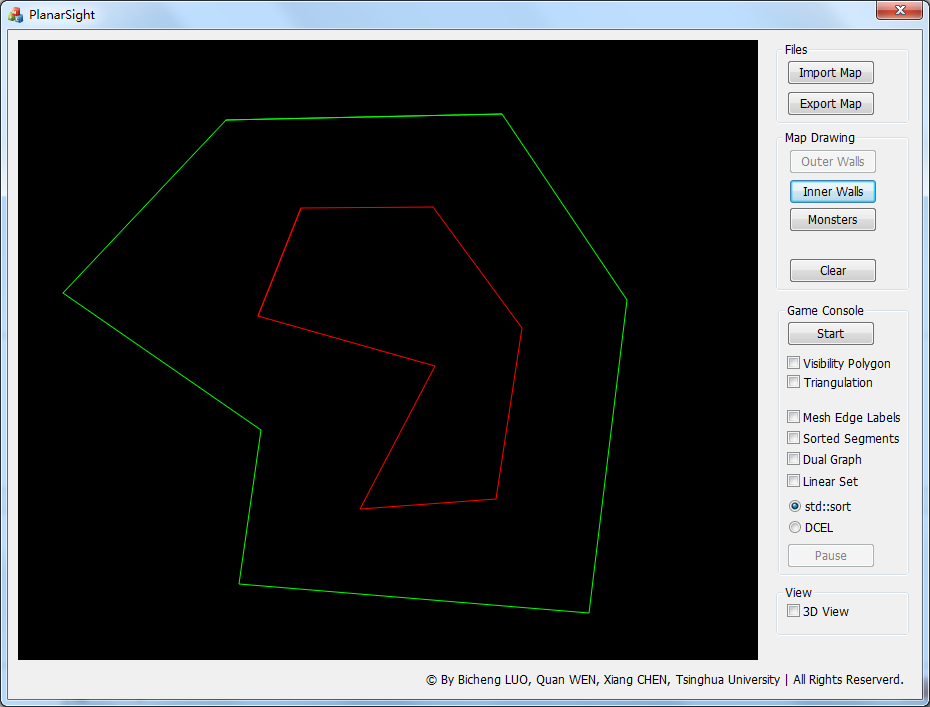
在Visual Studio中运行项目或直接运行可执行文件，可打开如下的程序主界面：



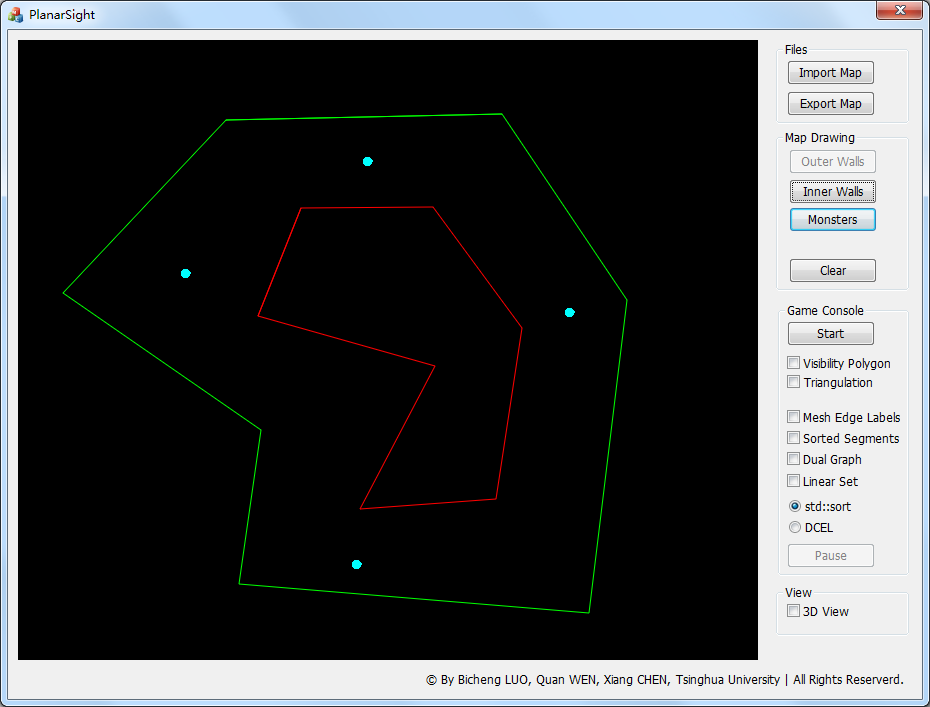
单击Import Map按钮可导入地图，单击Export Map可导出地图。

在绘制的部分，首先单击Outer Walls可绘制地图外环，依次用鼠标左键单击绘制区域形成点，最后单击鼠标右键结束。单击Inner Walls可绘制地图内环，绘制方法与外环相同，但内环需在外环之内。



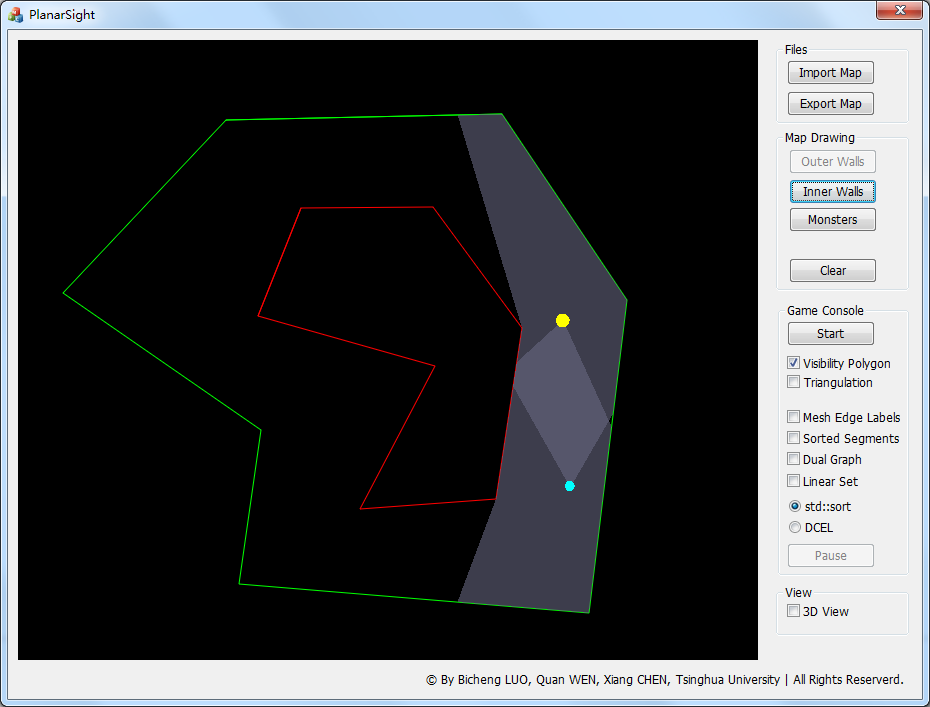


单击monsters可在地图中添加怪物，可用鼠标左键单击绘图区的多边形内绘制，最后单击鼠标右键结束。

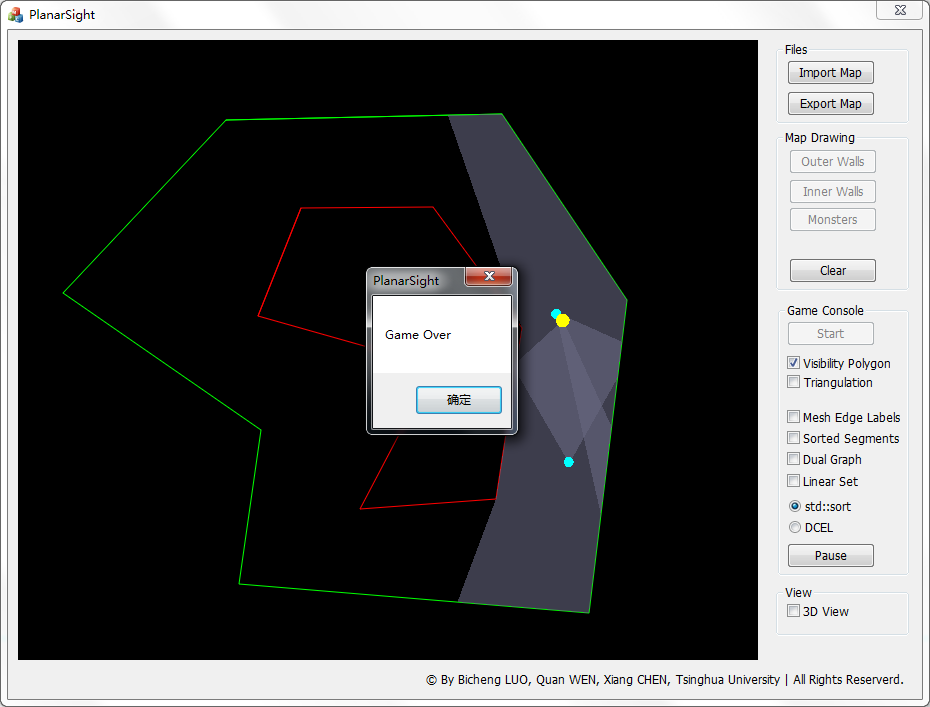


单击Clear可以清空整个绘图区。

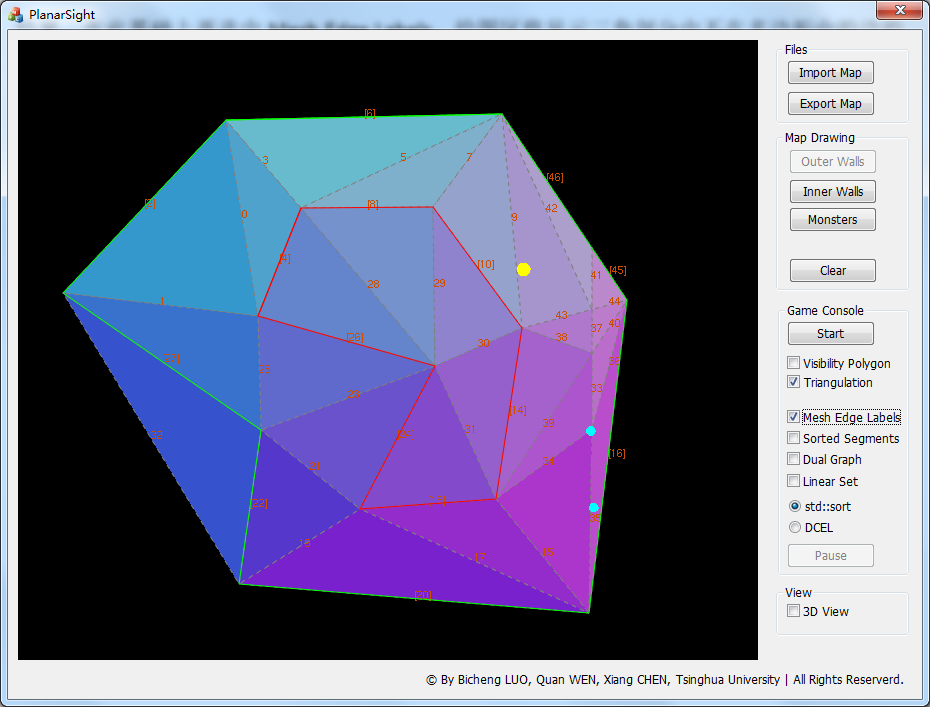
单击Start后游戏开始，怪物会在迷宫中随机行走，玩家则会随机生成在迷宫中的某个位置，可用鼠标单击表示玩家的圆点拖动玩家行走躲避怪物的追击。选中Visibility Polygon可显示玩家和怪物的可视范围，同时玩家只能看到可视范围内以及距离自身一定距离之内的怪物，怪物则会加速追击出现在其视野内的玩家。



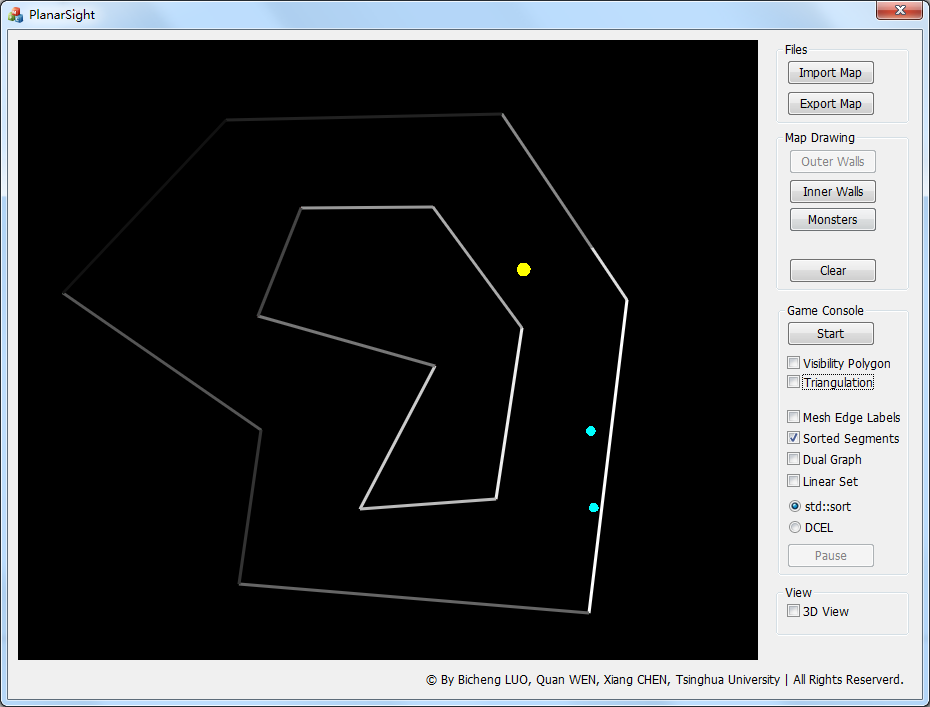
若玩家被怪物发现且怪物接近玩家到一定距离之内，则游戏结束。



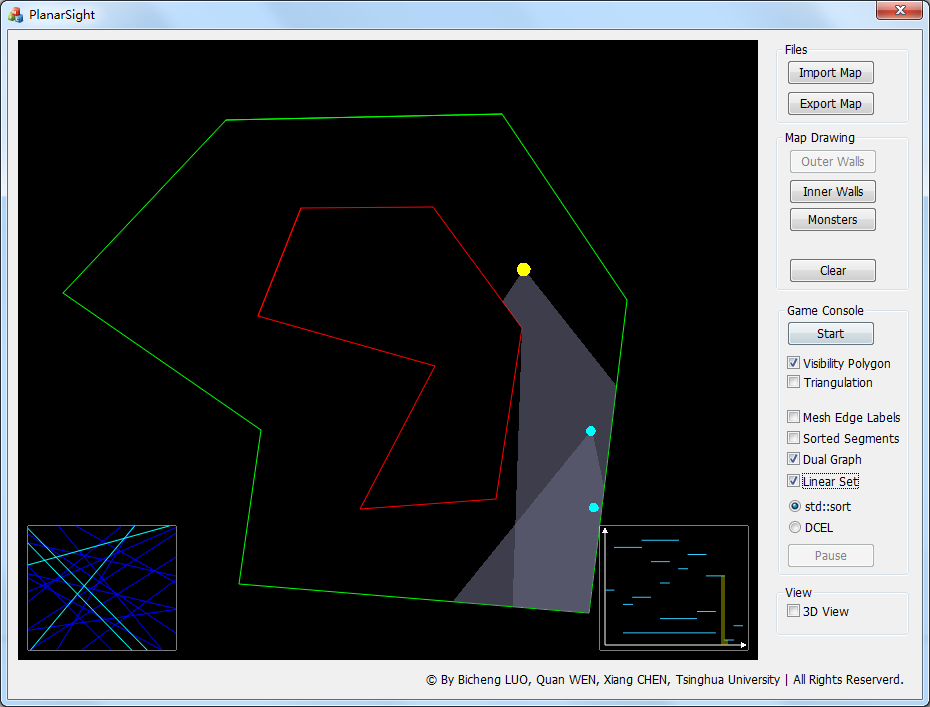
下面介绍工具栏中的一些其他功能。选中Triangulation，绘图区将显示三角剖分的中间结果。在此基础上再选中Mesh Edge Labels，绘图区将显示三角剖分中不在多边形中的边的标号。



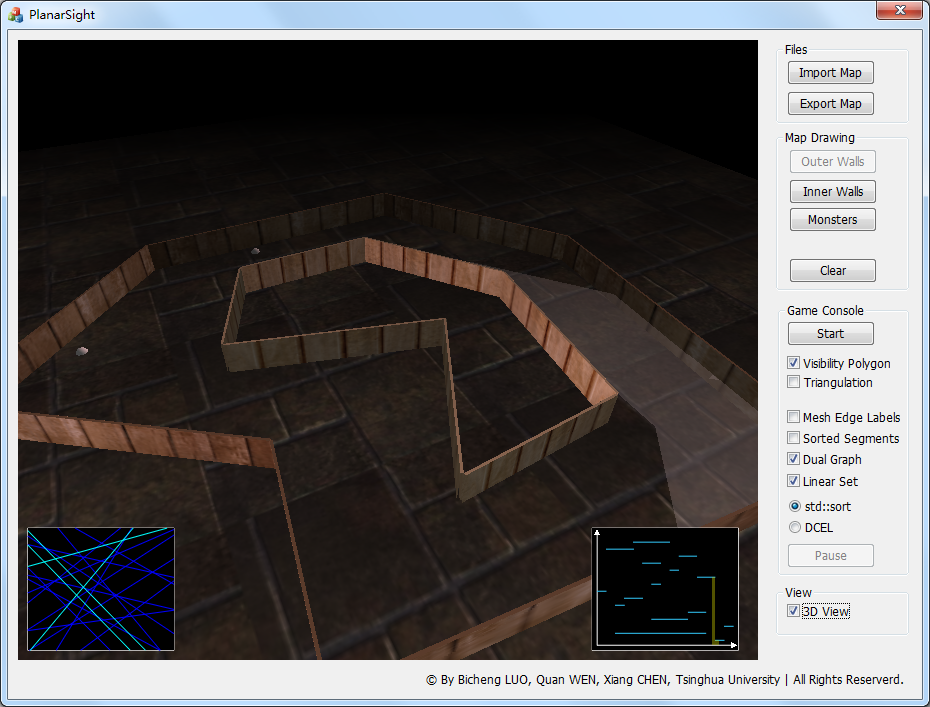
选中Sorted Segments，绘图区将显示多边形中线段偏序关系的排序结果，距离输入点越近的边越亮，反之越暗。



选中Dual Graph，绘图区将显示在使用DCEL求极角序的算法中DCEL中使用的对偶图。选中Linear Set，绘图区将显示线性集合并算法中对顶点极角序和线段偏序关系进行转化的结果图。



在std::sort和DCEL间切换可以选择使用不同的算法计算顶点极角排序结果。选中3D View则可观看地图的三维效果。



最后是一些复杂地图的运行效果：

