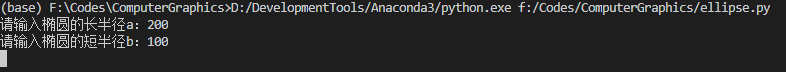
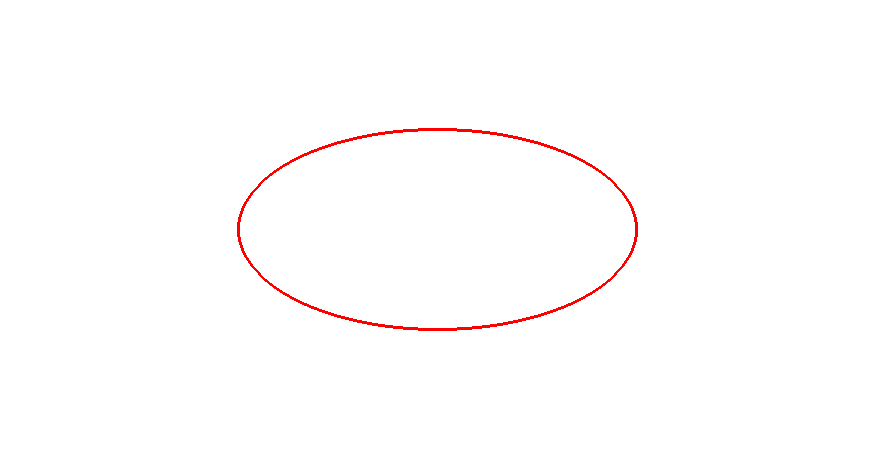
计算机图形学上机实验报告

1. 椭圆的扫描转换算法
2. 编程语言：python
3. 输入：椭圆的长半径和段半径
4. 输出：画出的椭圆

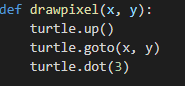
输出效果如图所示：



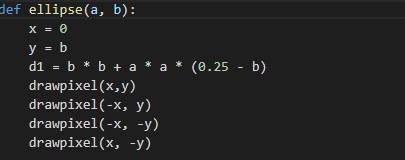


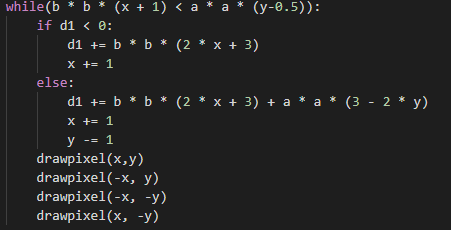
1. 代码截图：

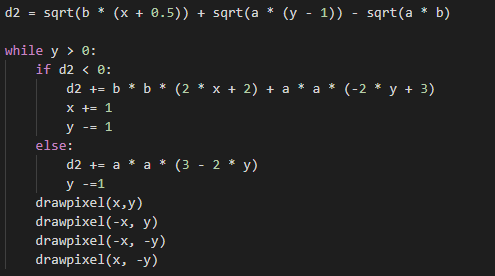
(1).画点函数：



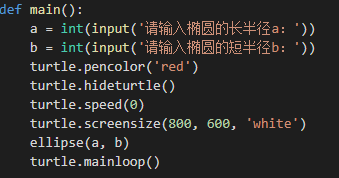
(2).画椭圆的函数





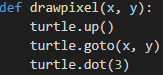


(3).主函数

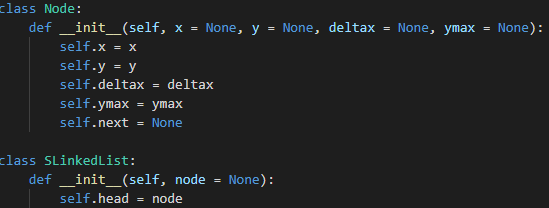


1. 六边形的扫描线填充算法
2. 编程语言：python
3. 输入：按顺序输入六边形的六个顶点
4. 输出：填充好的六边形
5. 代码截图：

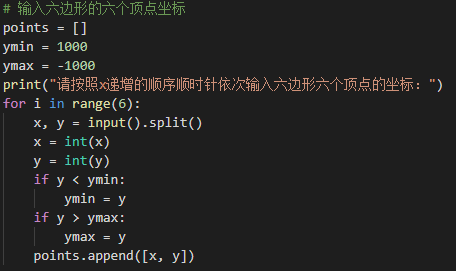
（1）、画图函数

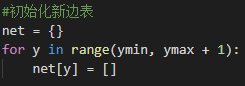


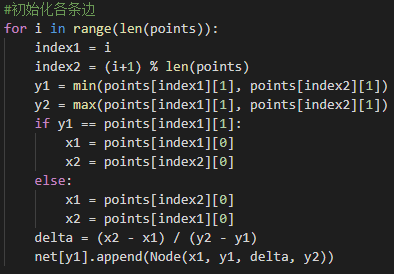
（2）、定义链表以及链表中结点的数据结构

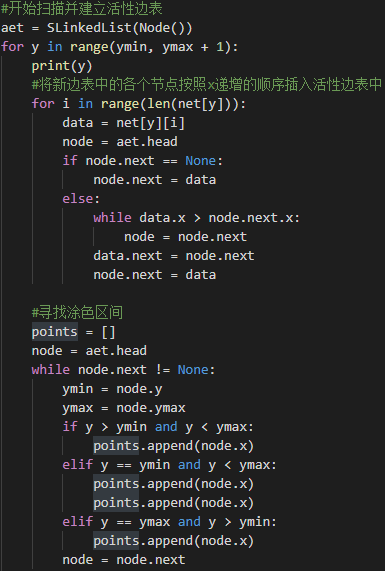


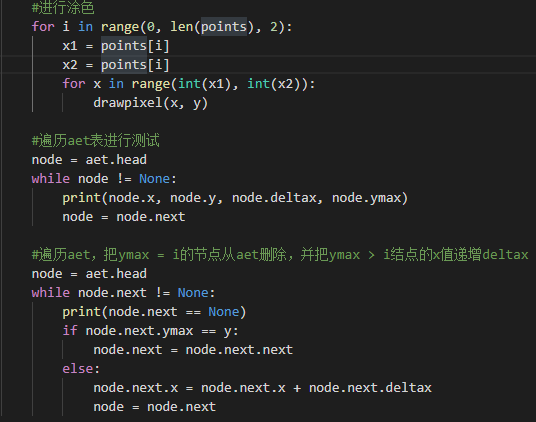
（3）、扫描线填充算法的核心代码



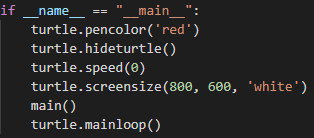








（4）、主函数代码



注：此处的main函数实为进行扫描线填充的函数

1. 程序运行结果截图：



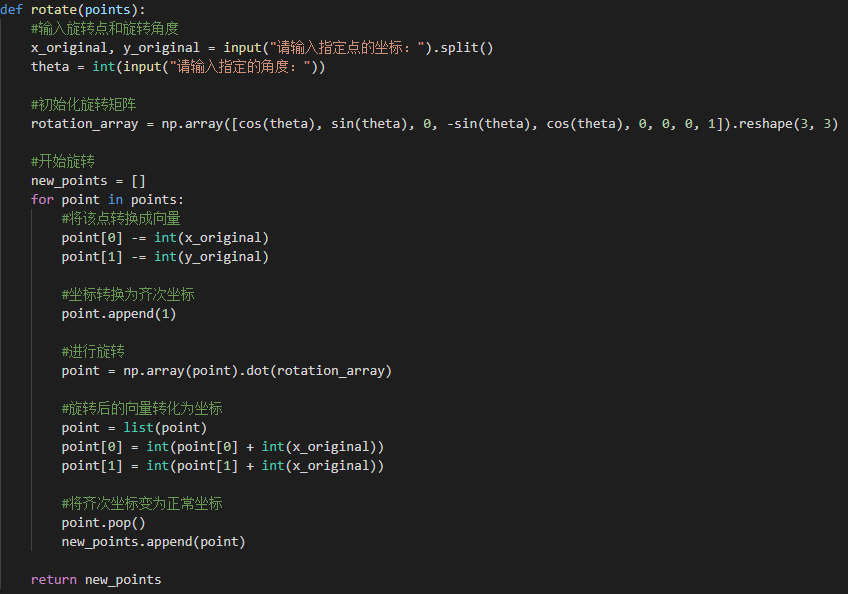


1. 六边形的旋转
2. 编程语言：python
3. 输入：按顺序输入六边形的六个顶点
4. 输出：旋转前的六边形以及旋转后的六边形
5. 代码截图：

（1）、画图函数：

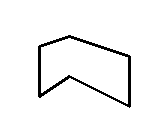


（2）、旋转函数：

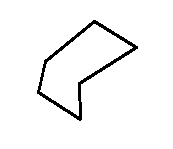


（3）、运行结果截图

旋转前：

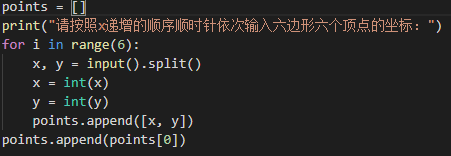


旋转后：

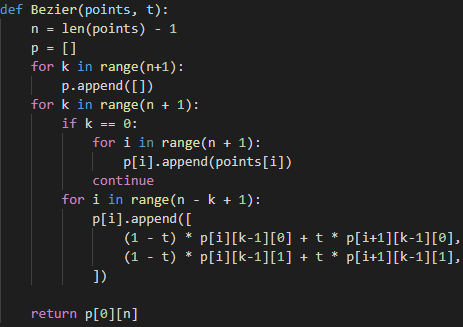


1. 六边形的Bezier曲线
2. 编程语言：python
3. 输入：按顺序输入六边形的各个顶点
4. 输出：原六边形（黑色）和生成的Bezier曲线（红色）
5. 代码：

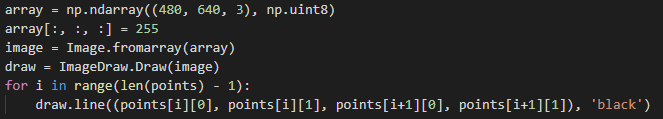
(1)、输入点的坐标



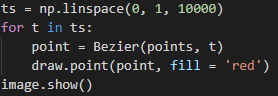
(2)、根据参数t求Bezier曲线上参数为t的点



(3)、新建画板并绘制六边形



(4)、取n个0到1之间的参数t，根据这些参数找到Bezier曲线的点，多个点最后集成变为Bezier曲线



1. 输出结果

结果如下图所示：

