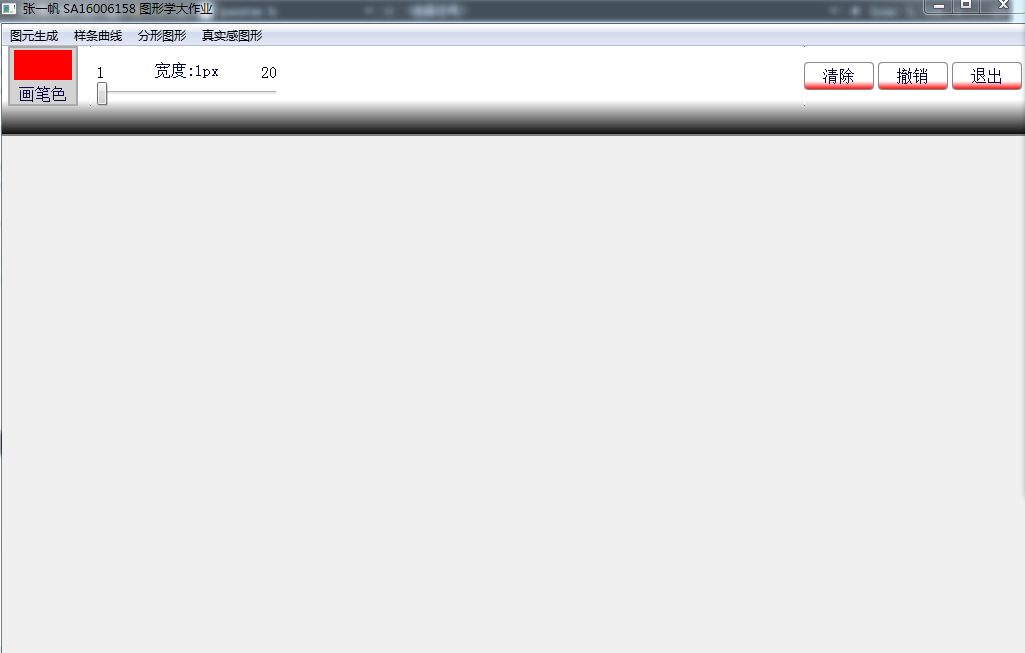
**计算机图形学**

**平台介绍：**

本次工程是一个Qt Quick Project，在QT Creator5.7平台上完成，采用[QML 与 C++ 混合编程，](http://blog.csdn.net/foruok/article/details/32698603)QML 构建界面， C++ 实现非界面的业务逻辑和复杂运算。这是目前一种比较流行的界面设计与算法实现方式。(可用QT Creator打开zyf\_graphic.pro文件）

**实验程序界面如下：**

****

界面介绍：菜单栏四个部分分别是我们实验的4部分要求，内容为：图元生成（直线，椭圆（圆是特殊的椭圆，在我们的编程中可以调节长短轴半径）），样条曲线（Bezier 曲线、 B 样条曲线），分形图形（Koch 曲线，蕨类植物），真实感图形（球体），下方的画笔色代表了我们画出图形的颜色（默认是黑色，建议在生成球体时换为其他颜色（如红色），这样能够明显看出阴影的变化）。然后滑动条是可以调节绘制出的笔的粗细（即线条宽度），右边的清除，撤销，退出按钮分别表示清屏，退回到上一步，退出界面操作。

**实验效果预览：**

****

关于怎样生成图形的操作说明，会在下面每一部分给出。

一.图元的生成

1. 直线生成算法——Bresenham 画线算法

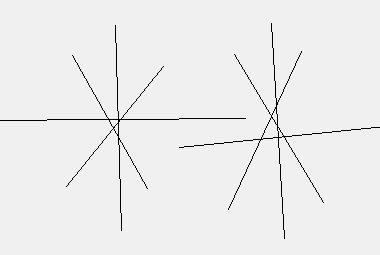
本实验实现了生成任意斜率，任意位置的直线。算法如下所示：







生成直线如下：



操作指南：进入图元生成，选择直线，进入绘图区域。选择起始点，按住鼠标左键，拖动就行。松开左键即这条直线生成了。当然还可以继续绘制，操作如上。

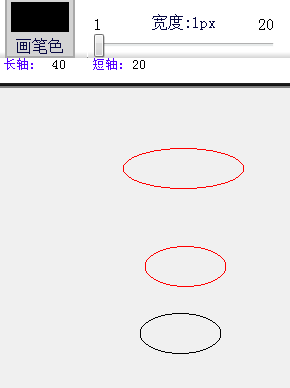
2. 椭圆的生成

本实验可以通过手动设置长短轴，画出各异的椭圆。采用中点算法，如下所示：



利用中点算法有效的避开了乘法除法以及浮点运算，生成的椭圆效果理想。

生成椭圆如下：

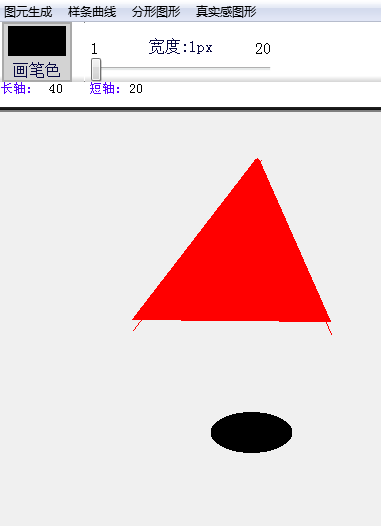


操作指南：进入图元生成，选择椭圆，在任意位置按下鼠标左键即可生成椭圆，可连续生成多个（改颜色后可直接生成）。可看见这里有长短轴，可在数字地方进行修改，实现你想要的目的。如修改长短轴后，则需要重新按上面操作才能画出椭圆。

3. 区域填充

本实验实现了画出三角形和椭圆，然后进行填充的功能。对于填充三角形使用的是扫描线填充算法，基本思想是用水平扫描线从上到下（或从下到上）扫描由多条首尾相连的线段构成的多边形(这里是三角形），每根扫描线与多边形的某些边产生一系列交点。将这些交点按照x坐标排序，将排序后的点两两成对，作为线段的两个端点，以所填的颜色画水平直线。多边形被扫描完毕后，颜色填充也就完成了；对于填充椭圆，采用的漫水法，基本思想是漫水法由一个给定内点开始，沿该内点所在的扫描线先向右扫描，直到边界，得到最右边的点；然后再沿该内点所在扫描线向左扫描，直到边界，得到该扫描线与边界相交的最左边的点；将最左面点的上下两个像素中还未被填充的入栈； 然后将栈顶元素出栈，重复上述填充过程，直到栈为空。

填充图形如下：



操作指南：按照之前直线绘制步骤，绘制出一个三角形区域，然后选择区域填充，在绘图区域点击鼠标左键即可；同理，填充椭圆区域，先绘制出一个椭圆，然后选择区域填充，在椭圆内部按左键即可。

二.样条曲线的生成

1. Bezier曲线的生成

本实验Bezier曲线的生成采用Casteljan递归算法，算法原理如下所示：



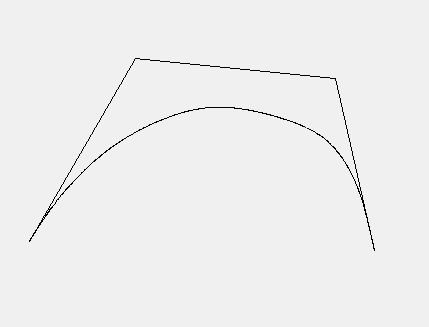
其中为前半部点，为下半部点

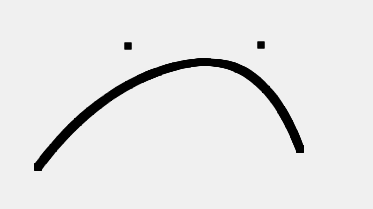
实验步骤：

1. 确定4个控制点
2. 判断4个点是否为8连通，如果是则绘制这4个点
3. 如果不是8连通，则继续进行递归运算casteljan()，casteljan()

初始控制点数目为N+1时，Bezier样条曲线的阶次为N。

Bezier曲线如下：





操作指南：进入样条曲线，选择Bezier曲线，然后在绘图区域可以直接标点绘制曲线，也可拖动按住鼠标左键，拖动，把生成的直线显现出来，形成Bezier曲线。

1. B样条曲线的生成

B 样条的多项式次数与控制点无关， 允许局部调整，但构造相比于 Bezier 要复杂。  
 对于 n+1 个控制点生成的 d-1 次 B 样条，可以定义为如下：



其中是输入的一组N+1个控制点，d为B样条混合函数的阶次，混合函数是一个d-1次多项式。

B-样条如下：



操作指南：进入样条曲线，选择B-样条，其他操作步骤与生成Bezier曲线一致。

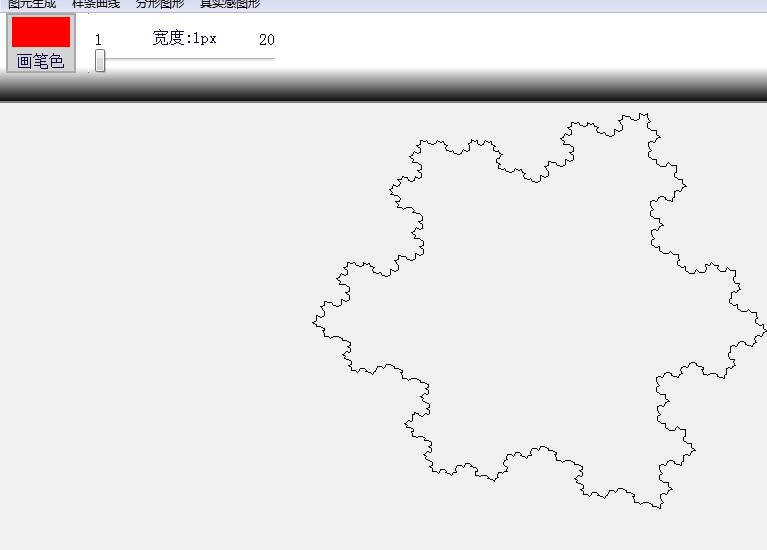
1. 分形图形生成

在这一部分，我选择生成的koch曲线以及蕨类植物(fern)。

1. koch曲线

基本思想如下：使用三角形作为初始图元，然后初始图元中的每个直线段使用 4 个相等的线段替代，替代方式为：取 P0P1 线段三等分点 P10， P11，第一段为 P0P10，第二段为 P10P11 绕 P10 逆时针旋转 60 度，获得点 P13，即第二段为 P10P13，第三段为 P13P11，第四段为 P11P1，每条线段在作为初始图元继续迭代，可以获得不  
同阶次的 Koch 曲线。

koch曲线如下：



操作指南：进入分形图形，选择koch,在绘图区域选择起始位置，按住左键，绘制等边三角形，松开后，连续点击鼠标，就能得到不同阶数的Koch曲线。

2.fern(蕨类植物)

蕨类植物是由一组仿射变换迭代产生的分形，迭代公式如下：





A,B,C,D可看做是缩放和旋转参数，E,F是平移参数。每次迭代以不同概率进行不同的仿射变换。



fern如下：



操作指南：进入分形图形，选择fern，在绘图区域点击鼠标左键即可生成想要图形，可连续生成。

1. 真实感图形生成

真实感图形的生成，包括三维几何造型的计算，投影，消隐（消去隐藏面），光照模型，纹理映射等。

几何造型就是设计想要生成的形状，比如球体、圆柱、立方体等。

投影是从三维到二维平面的一个映射，投影有多种方式（比如平行投影和透视投影），可以用旋转、平移、伸缩的组合和正投影来实现多种投影。

消隐是消除在视线方向看不到的线或面，可以用画家算法、Z-Buffer算法等。对于一些简单图形也可以利用视线向量与平面法向量之间的关系来实现消隐。

光照模型是模拟光照射在物体表面的效果，包括反射（漫反射和镜面反射）、透射（透明）等。光照因素有：环境光，对光源（平行或点光源）的漫反射和镜面反射（暂不考虑透明），其中镜面反射又和物体本身的材质有关。而光源的亮度也会随着传播距离的增加而减小。

纹理映射是把定义在二维平面上的图形映射到三维图形上，提供较强的真实感。其实现过程为：将二维平面的图形映射到三维立体图形的曲面上，然后再投影到二维平面上进行显示。

具体生成步骤如下：

（1）实验中几何造型是简单的球体

（2）由球的参数模型生成集合造型表面的法向量N，为。

（3)消隐：消隐是用自消隐方法（用视线向量和图形表面法向量的内积来判断，大于0则可见，否则不可见）。

（4）投影：将可见面从世界坐标系映射到观察坐标系

（5）光照模型：实验中选取的是 Phone 模型，即包括泛光、漫反射光和镜面反射光，  
 具体模型计算公式如下：

𝐼 = 𝑘𝑎𝐼𝑎 + 𝑘𝑑𝐼0𝑐𝑜𝑠𝜃 + 𝑘𝑠𝐼0𝑐𝑜𝑠𝑛𝜑

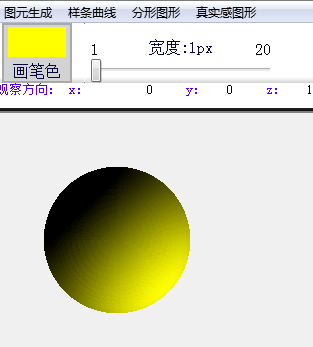
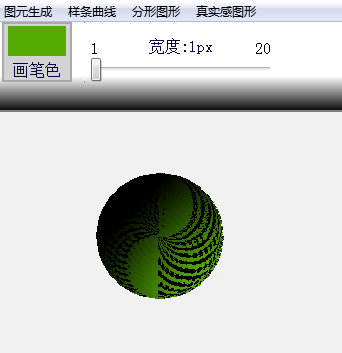
其中𝑘𝑎𝐼𝑎是泛光部分， 𝑘𝑑是漫反射的反射系数，与物体材质有关， 𝜃是表面法向量和光线的夹角， 𝑘𝑠是镜面反射光的反射系数， 𝜑是视线与反射光线之间的夹角，具体参数设置参照程序，另外程序中允许自定义观察实现方向和光线方向。

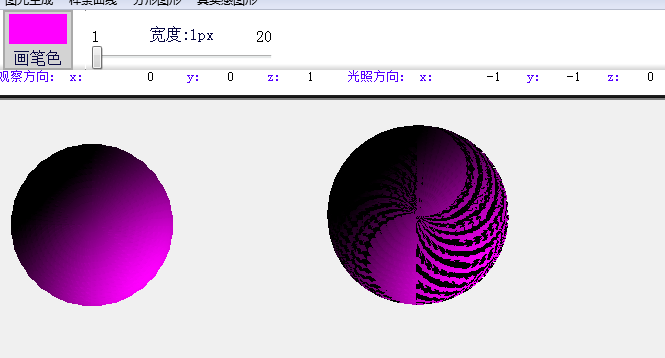
（6）画面绘制：将光强与相应的RGB三个通道直接相乘即可。此时要做一个线性变换，拉伸到【0,255】之间。

（7）纹理映射：在物体几何造型建立时将纹理模式映射到对象的表面。实验中通过修改对象表面光强度值的过程来定义纹理模式。

真实感图形如下：

生成球体 球体纹理映射



操作指南：首先一定要更改画笔的颜色（如红色），这样便于观察效果。进入真实感图形，选择球体，然后在绘图区域按住鼠标左键，轻轻拖动，拖动长度不要太长，松开即可看见绘制的球体。同理，可以绘制出纹理映射图形。