

○：这里我们假设已经有了一个绘图函数 `setPixel(x,y,color)`,这个函数的功能是将屏幕的(x,y)处像素颜色设置为 color(在 vesa 显示模式下，直接设置显存中的某个字节就可以改变相应像素的颜色，不过这需要直接写显存，我试过可以写成 bootloader 或者其他方式，windows 下面可以使用 GDI,GDI+,directX2D,easyx 等画点)，因为现在处理器运算速度比较快，浮点数处理速度也很快，所以本文主要是将 3D 图形学入门的数学原理，对代码的优化暂时就不作出来。通常显存映射会把一段连续的内存映射到屏幕的每一个像素，假设屏幕宽高为 $W \times H$ ，显存起始地址为 GM ，每个像素占三个字节用于保存 RGB，那么在屏幕上任意一点(x,y) ($0 \leq x < W$, $0 \leq y < H$) 的内存地址为 $p = GM + (x \times W + y) \times 3$ ，这时候只要对从 p 开始的三个字节赋值即可设置对应像素的 RGB 值(我认为显卡会在下次刷新的时候将内存的变化反应到屏幕上)，当然有的时候为了内存对齐，一行像素不一定占用 $W \times 3$ 字节，可能会在后面补充几个字节，这时候就要根据具体情况具体计算了，本示例代码使用 EasyX 作为底层图形库(见 Graphics.cpp 和 Graphics.h)，有兴趣的读者可以使用 GTK 或者 QT 等任意方式将其实现其他平台。

前面几张先讲述 2D 图形学的一些基本技术(绘制直线、填充多边形、裁剪等基础算法)，后面将会使用这些基础算法扩充完成 3D 图形学入门教程。