4缓存：程序从CPU的内存复制数据到OpenGL ES的缓存。GPU访问缓存中的数据。从而节省内存开销。

5帧缓存：接受渲染结果的缓冲区。

22 demo OpenGLES\_Ch2\_1 三角形

32 demo OpenGLES\_Ch2\_2

50映射：程序需要指定怎么对齐纹理和顶点，以便让GPU知道每个片元的颜色由哪些纹素决定。U、V是映射X、Y到S、T的映射坐标

51取样：每个顶点的U和V坐标会附加到每个顶点在视口坐标中的最终位置。然后GPU会根据计算出来的每个片元的U、V位置从绑定的纹理中选择纹素。

51 glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR)出现多个纹素对应一个片元，从相配的多个纹素中取样颜色，然后使用线性内插法来混合这些颜色以得到片元的颜色。

GL\_NEAREST则取与片元的U、V最接近的纹素的颜色

52 glTexParameteri(GL\_TEXTURE,GL\_TEXTURE\_WRAP\_S, GL\_CLAMP\_TO\_EDGE)当U大于S或小于0时，取样纹理边缘的纹素。 GL\_REPEAT则重复

52 MIP贴图：为纹理存储多个细节级别的技术

52 demo OpenGLES\_Ch3\_1

53将纹理图像加载到纹理缓存中GLKTextureLoader textureWithCGImage: 可在options里加GLKTextureLoaderOriginBottomLeft抵消垂直翻转

GLKTextureInfo会自动调用glTextParameteri为创建的纹理缓存设置取样和循环模式

62 demo OpenGLES\_Ch3\_3

64开启混合,glEnable(GL\_BLEND)

设置混合函数glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA)其中GL\_SRC\_ALPHA用于让源片元的透明度元素挨个与其他的片元颜色相乘。GL\_ONE\_MINUS\_SRC\_ALPHA用于让源片元的透明度元素(1.0)与在帧缓存内正被更新的颜色元素相乘。如果片元透明度是1，则替代帧缓存内对应的像素颜色，否则一部分会被添加到帧缓存内对应的像素颜色的一部分来产生混合的结果

最终颜色Redfinal=Alphafrag\*Redfrag+(1.0-Alphafrag)\*Redframe buffer

64 demo OpenGLES\_Ch3\_4

68 demo OpenGLES\_Ch3\_6着色器语言 没有详解

70检查是否支持纹理压缩glGetString(GL\_EXTENSIONS) 是否含GL\_IMG\_texture\_compression\_pvrtc

75右手法则：四指指向x，向y弯曲，大拇指指向的方向就是z。同样适用于向量积 或者：让右手拇指指向x轴的正方向，食指指向y轴的正方向，掌心即z方向

79 光源位置：.light0.position是GLKVector4，如果第4个元素非零，就是点光源，否则，则前三个元素表示光的方向向量

80 demo OpenGLES\_Ch4\_1

87 demo OpenGLES\_Ch4\_2灯光烘培进纹理，这里只适用静态，即片元分段阴影在图片上就是死的

90深度缓存view.drawableDepthFormat=GLKViewDrawableDepthFormat16

glDepthFunc(GL\_LESS)片元的深度值小于深度缓存中该片元的位置所保存的值时才会替换像素颜色缓存的内容

demo OpenGLES\_Ch5\_1 OpenGLES\_Ch5\_2

96设置变换矩阵抵消拉伸self.baseEffect.transform.projectionMatrix

97 demo OpenGLES\_Ch5\_3 用UIView而不是GLKView

101透视GLKMatrix4MakeFrustum，近大远小

103 度数转弧度数GLKMathDegreesToRadians

105 堆栈函数GLKMatrixStack push和pop之间写入动态变化的量

105 demo OpenGLES\_Ch5\_5

109 定义视域GLKMatrix4MakeOrtho

111 demo OpenGLES\_Ch6\_1 GLKMatrix4MakeLookAt计算视点变换矩阵，返回一个model-view矩阵，这个矩阵会对齐从眼睛的位置到看向的位置之间的矢量与当前视域的中心线。参数:前3个是眼睛位置，中间3个是观察者正在看向的xyz位置。

116 demo OpenGLES\_Ch6\_2随着时间改变网格顶点的Y坐标来产生网格动画

118 为了应用索引顶点，需要创建元素数组缓存glBindBuffer(GL\_ELEMENT\_ARRAY\_BUFFER,indexBuffer)

122 demo OpenGLES\_Ch6\_3聚光灯

self.baseEffect.lightingType: GLKLightingTypePerVertex会计算几何图像中的每个顶点的光线值，并在顶点之间插值，容易出现锯齿。设置GLKLightingTypePerPixel会产生一个更高质量的渲染结果，但折需要为每个片元重新计算光线效果，比每顶点计算耗费更多的GPU计算值

127 demo OpenGLES\_Ch6\_4旋转的纹理

128 demo OpenGLES\_Ch6\_5纹理中按时间部分截取形成的动画

132一些建模工具

135 demo OpenGLES\_Ch7\_1 加载COLLADAViewer导出的.modelplist文件

144 demo OpenGLES\_Ch7\_2 关节:由一个位置和一个变换矩阵组成，从一个关节到下一个关节之间的位移代表了一个不可见的骨头。关节与关节之间存在相对位置

148 demo OpenGLES\_Ch7\_3 蒙皮：设置每个顶点的关节索引和权重的过程

151 demo OpenGLES\_Ch8\_1天空盒：把渲染场景包裹在一个六面都是纹理图片的立方体中 用到了GLKSkyboxEffect类

157着色器语言Shading Language

varying声明可选的可变变量(输入变量)，可用于保存从顶点着色器获得的属性 attribute声明属性

159栅格化：转换几何图形数据为帧缓存内的颜色像素的渲染步骤

160片元中的纹理函数textureCube

164点精灵 demo OpenGLES\_Ch8\_3

170 公告牌 demo OpenGLES\_Ch8\_4

179基于视平截体的剔除 demo OpenGLES\_Ch9\_1

195地形网格 demo OpenGLES\_Ch10\_1

222 demo OpenGLES\_Ch10\_2