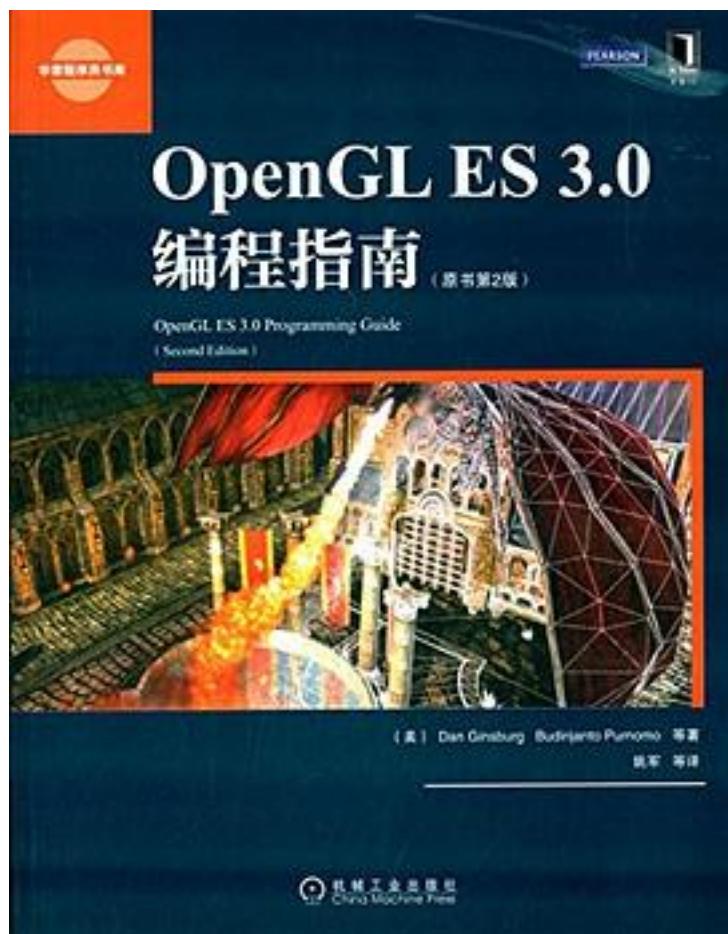


# OpenGL ES 3.0编程指南



[OpenGL ES 3.0编程指南\\_下载链接1](#)

著者:金斯伯格 (Dan Ginsburg)

出版者:机械工业出版社

出版时间:2015-2-1

装帧:平装

isbn:9787111489153

作者介绍:

Dan Ginsburg, Upsample

Software公司创始人，该软件公司提供3D图形和GPU计算的咨询服务。Dan参与撰写了多部著作，包括《OpenCL 编程指南》和《OpenGL Shading Language》（第3版）。在过去的工作中，Dan曾经开发OpenGL驱动程序、桌面和手持3D演示、GPU开发人员工具、3D医学显示和游戏。

Budirijanto

Purnomo, AMD公司高级软件架构师，领导跨越多个AMD软件栈的GPU调试和剖析技术的软件实现工作。他与多位AMD的软件和硬件架构师协作，定义用于调试和剖析GPU应用的未来硬件架构。他曾在国际会议上发表过许多与计算机图形技术相关的论文。

Dave

Shreiner, ARM公司图形与GPU计算部门主管，自从OpenGL诞生之日起就积极地参与到它的开发当中。他创建了OpenGL的第一个商业培训课程，并且拥有超过20年的OpenGL编程教学经验。

## 目录: 本书赞誉

译者序

序

前言

致谢

第1章 OpenGL ES 3.0简介 1

1.1 OpenGL ES 3.0 2

1.1.1 顶点着色器 3

1.1.2 图元装配 5

1.1.3 光栅化 5

1.1.4 片段着色器 6

1.1.5 逐片段操作 7

1.2 OpenGL ES 3.0新功能 8

1.2.1 纹理 8

1.2.2 着色器 10

1.2.3 几何形状 11

1.2.4 缓冲区对象 11

1.2.5 帧缓冲区 12

1.3 OpenGL ES 3.0和向后兼容性 12

1.4 EGL 13

1.4.1 使用OpenGL ES 3.0编程 14

1.4.2 库和包含文件 14

1.5 EGL命令语法 14

1.6 OpenGL ES命令语法 15

1.7 错误处理 16

1.8 基本状态管理 16

1.9 延伸阅读 17

第2章 你好，三角形：一个OpenGL ES 3.0示例 18

2.1 代码框架 18

2.2 示例下载位置 19

2.3 “你好，三角形”（Hello Triangle）示例 19

2.4 使用OpenGL ES 3.0框架 23

2.5 创建简单的顶点和片段着色器 24

2.6 编译和加载着色器 25

2.7 创建一个程序对象并链接着色器 26

2.8 设置视口和清除颜色缓冲区 27

2.9 加载几何形状和绘制图元 28

2.10 显示后台缓冲区	29
2.11 小结	29
第3章 EGL简介	30
3.1 与窗口系统通信	30
3.2 检查错误	31
3.3 初始化EGL	32
3.4 确定可用表面配置	32
3.5 查询EGLConfig属性	33
3.6 让EGL选择配置	35
3.7 创建屏幕上的渲染区域：EGL窗口	37
3.8 创建屏幕外渲染区域：EGL Pbuffer	39
3.9 创建一个渲染上下文	42
3.10 指定某个EGLContext为当前上下文	43
3.11 结合所有EGL知识	44
3.12 同步渲染	46
3.13 小结	46
第4章 着色器和程序	47
4.1 着色器和程序	47
4.1.1 创建和编译一个着色器	48
4.1.2 创建和链接程序	51
4.2 统一变量和属性	54
4.2.1 获取和设置统一变量	55
4.2.2 统一变量缓冲区对象	60
4.2.3 获取和设置属性	64
4.3 着色器编译器	64
4.4 程序二进制码	65
4.5 小结	66
第5章 OpenGL ES着色语言	67
5.1 OpenGL ES着色语言基础知识	67
5.2 着色器版本规范	68
5.3 变量和变量类型	68
5.4 变量构造器	69
5.5 向量和矩阵分量	70
5.6 常量	71
5.7 结构	71
5.8 数组	72
5.9 运算符	72
5.10 函数	73
5.11 内建函数	74
5.12 控制流语句	74
5.13 统一变量	75
5.14 统一变量块	75
5.15 顶点和片段着色器输入/输出	77
5.16 插值限定符	79
5.17 预处理器和指令	80
5.18 统一变量和插值器打包	81
5.19 精度限定符	82
5.20 不变性	83
5.21 小结	85
第6章 顶点属性、顶点数组和缓冲区对象	87
6.1 指定顶点属性数据	87
6.1.1 常量顶点属性	88
6.1.2 顶点数组	88
6.2 在顶点着色器中声明顶点属性变量	95

6.3 顶点缓冲区对象	98
6.4 顶点数组对象	106
6.5 映射缓冲区对象	109
6.6 复制缓冲区对象	112
6.7 小结	113
第7章 图元装配和光栅化	114
7.1 图元	114
7.1.1 三角形	114
7.1.2 直线	115
7.1.3 点精灵	116
7.2 绘制图元	117
7.2.1 图元重启	119
7.2.2 驱动顶点	119
7.2.3 几何形状实例化	120
7.2.4 性能提示	122
7.3 图元装配	124
7.3.1 坐标系统	124
7.3.2 透视分割	126
7.3.3 视口变换	126
7.4 光栅化	127
7.4.1 剔除	127
7.4.2 多边形偏移	128
7.5 遮挡查询	130
7.6 小结	131
第8章 顶点着色器	132
8.1 顶点着色器概述	133
8.1.1 顶点着色器内建变量	133
8.1.2 精度限定符	135
8.1.3 顶点着色器中的统一变量限制数量	136
8.2 顶点着色器示例	138
8.2.1 矩阵变换	139
8.2.2 顶点着色器中的照明	141
8.3 生成纹理坐标	145
8.4 顶点蒙皮	146
8.5 变换反馈	150
8.6 顶点纹理	151
8.7 将OpenGL ES 1.1顶点管线作为ES 3.0顶点着色器	152
8.8 小结	159
第9章 纹理	160
9.1 纹理基础	160
9.1.1 2D纹理	161
9.1.2 立方图纹理	161
9.1.3 3D纹理	162
9.1.4 2D纹理数组	163
9.1.5 纹理对象和纹理的加载	163
9.1.6 纹理过滤和mip贴图	167
9.1.7 自动mip贴图生成	171
9.1.8 纹理坐标包装	172
9.1.9 纹理调配	173
9.1.10 纹理细节级别	173
9.1.11 深度纹理对比（百分比渐进过滤）	174
9.1.12 纹理格式	174
9.1.13 在着色器中使用纹理	180
9.1.14 使用立方图纹理的示例	182

9.1.15 加载3D纹理和2D纹理数组	184
9.2 压缩纹理	186
9.3 纹理子图像规范	188
9.4 从颜色缓冲区复制纹理数据	191
9.5 采样器对象	193
9.6 不可变纹理	195
9.7 像素解包缓冲区对象	196
9.8 小结	197
第10章 片段着色器	198
10.1 固定功能片段着色器	199
10.2 片段着色器概述	200
10.2.1 内建特殊变量	201
10.2.2 内建常量	202
10.2.3 精度限定符	202
10.3 用着色器实现固定功能技术	203
10.3.1 多重纹理	203
10.3.2 雾化	204
10.3.3 Alpha测试（使用Discard）	207
10.3.4 用户裁剪平面	208
10.4 小结	209
第11章 片段操作	210
11.1 缓冲区	211
11.1.1 请求更多缓冲区	211
11.1.2 清除缓冲区	212
11.1.3 用掩码控制帧缓冲区的写入	213
11.2 片段测试和操作	214
11.2.1 使用剪裁测试	214
11.2.2 模板缓冲区测试	215
11.3 混合	220
11.4 抖动	221
11.5 多重采样抗锯齿	222
11.6 在帧缓冲区读取和写入像素	223
11.7 多重渲染目标	226
11.8 小结	229
第12章 帧缓冲区对象	230
12.1 为什么使用帧缓冲区对象	230
12.2 帧缓冲区和渲染缓冲区对象	231
12.2.1 选择渲染缓冲区与纹理作为帧缓冲区附着的对比	232
12.2.2 帧缓冲区对象与EGL表面的对比	232
12.3 创建帧缓冲区和渲染缓冲区对象	233
12.4 使用帧缓冲区对象	234
12.4.1 多重采样渲染缓冲区	235
12.4.2 渲染缓冲区格式	235
12.5 使用帧缓冲区对象	237
12.5.1 连接渲染缓冲区作为帧缓冲区附着	238
12.5.2 连接一个2D纹理作为帧缓冲区附着	238
12.5.3 连接3D纹理的一个图像作为帧缓冲区附着	240
12.5.4 检查帧缓冲区完整性	241
12.6 帧缓冲区位块传送	242
12.7 帧缓冲区失效	243
12.8 删除帧缓冲区和渲染缓冲区对象	244
12.9 删除用作帧缓冲区附着的渲染缓冲区对象	245
12.10 示例	246
12.11 性能提示和技巧	251

12.12 小结 251

第13章 同步对象和栅栏 252

13.1 刷新和结束 252

13.2 为什么使用同步对象 253

13.3 创建和删除同步对象 253

13.4 等待和向同步对象发送信号 253

13.5 示例 254

13.6 小结 255

第14章 OpenGL ES 3.0高级编程 256

14.1 逐片段照明 256

14.1.1 使用法线贴图的照明 257

14.1.2 照明着色器 258

14.1.3 照明方程式 260

14.2 环境贴图 261

14.3 使用点精灵的粒子系统 264

14.3.1 粒子系统设置 264

14.3.2 粒子系统顶点着色器 265

14.3.3 粒子系统片段着色器 266

14.4 使用变换反馈的粒子系统 268

14.4.1 粒子系统渲染算法 269

14.4.2 使用变换反馈发射粒子 269

14.4.3 渲染粒子 273

14.5 图像后处理 274

14.5.1 渲染到纹理设置 274

14.5.2 模糊片段着色器 274

14.5.3 眩光 275

14.6 投影纹理 276

14.6.1 投影纹理基础 277

14.6.2 投影纹理所用的矩阵 278

14.6.3 投影聚光灯着色器 278

14.7 使用3D纹理的噪声 281

14.7.1 生成噪声 281

14.7.2 使用噪声 285

14.8 过程纹理 286

14.8.1 过程纹理示例 287

14.8.2 过程纹理的抗锯齿 289

14.8.3 关于过程纹理的延伸阅读 291

14.9 用顶点纹理读取渲染地形 291

14.9.1 生成一个正方形的地形网格 292

14.9.2 在顶点着色器中计算顶点法线并读取高度值 293

14.9.3 大型地形渲染的延伸阅读 294

14.10 使用深度纹理的阴影 294

14.10.1 从光源位置渲染到深度纹理 294

14.10.2 从眼睛位置用深度纹理渲染 297

14.11 小结 299

第15章 状态查询 300

15.1 OpenGL ES 3.0实现字符串查询 300

15.2 查询OpenGL ES实现决定的限制 301

15.3 查询OpenGL ES状态 303

15.4 提示 306

15.5 实体名称查询 307

15.6 不可编程操作控制和查询 307

15.7 着色器和程序状态查询 308

15.8 顶点属性查询 309

15.9 纹理状态查询	310
15.10 采样器查询	310
15.11 异步对象查询	311
15.12 同步对象查询	311
15.13 顶点缓冲区查询	312
15.14 渲染缓冲区和帧缓冲区状态查询	312
15.15 小结	313
第16章 OpenGL ES平台	314
16.1 在包含Visual Studio的Microsoft Windows上构建	314
16.2 在Ubuntu Linux上构建	316
16.3 在Android 4.3+ NDK (C++) 上构建	317
16.3.1 先决条件	317
16.3.2 用Android NDK构建示例代码	318
16.4 在Android 4.3+ SDK上构建 (Java)	318
16.5 在iOS 7上构建	319
16.5.1 先决条件	319
16.5.2 用Xcode 5构建示例代码	319
16.6 小结	320
附录A GL_HALF_FLOAT	321
附录B 内建函数	325
附录C ES框架API	338
• • • • • (收起)	

[OpenGL ES 3.0编程指南 下载链接1](#)

## 标签

OpenGL

图形学

游戏开发

计算机图形学

翻译渣，史上之最。

编程

C++

代码

## 评论

opengl es

入门最好的一本书。多说一句，opengl本身的api简直巨库，宏大而至于让入门者难寻章法。es是opengl的子集，api脉络清晰扼要，如果有想要学习opengl的同学，想必以后我都会推荐此书给他入门，更能提起兴趣。

---

后半部分翻译有地方语句都不连贯，前面还可以。需对照英文原版看比较好，以后买书看到翻译有等这个字的，练习先看看再入手。

---

这本书如果仔细看，能拓展到不少便利知识点和优化技巧，对于ES3.0较于ES2.0的新特性也有较多的介绍，读书笔记：[https://github.com/MartinRGB/GLES30\\_ProgrammingGuide\\_NDK/wiki](https://github.com/MartinRGB/GLES30_ProgrammingGuide_NDK/wiki)

---

英文版还是不错的，中文版的就不要买了，翻译的太勉强，看的非常难受

---

中文版翻译得不好，很多词语翻译过来第一次都搞不懂在说什么。

但是这本书本身是非常值得一读的，尤其当你对GLES

不是特别熟悉的时候。但是这本书深度不够，没有把GLES常用的高级技巧讲明白。

作为一本介绍入门书，是非常不错的！

---

翻译一般

---

最近在看渲染相关内容，当工具书查阅很详尽，很棒//PAUSE  
大概读一半了吧//感觉还不错

不推荐，翻译太渣！

说实话，翻译这本书的人压根没有用OpenGL写过代码吧

很长时间没完整的看完一本技术书了，之前虽做了两年Unity，学些Shader，但缺乏对GLSL和HLSL的了解，现在中国手游热，自己现在也在从事移动端游戏引擎的相关工作，对GLES的了解也是必须的。这书虽然不十分出彩，但知识点都涵盖了，也并不全是罗列API。是有不小收获的，当你了解了这些基础图形API的时候再回头看Unity的一些所谓的新功能无非是在底层API基础上包装了一下。书中涉及到的一些技术还并没有吃透，没有实际应用过，还要继续学习。 --2016年4月20日于北京

原书给3，翻译给1，，翻译错误，不通顺，字漏打，粗心大意，，  
原书也有错误，，并且不全面，，

学习webgl资料太少，只能看opengl es，好在差别不大。

灾难性的翻译，读着很累...

见过的最垃圾的机翻

中文版翻译质量实在不敢恭维，大部分的专有名词也不标注英文名称，就算知道作者说的是什么看起来还是要在脑海中翻译一下，并不建议拿这本书的中文版入门

[OpenGL ES 3.0编程指南 下载链接1](#)

书评

论入门学习， DirectX比OpenGL友善得多，版本简单明了，装好SDK开Visual Studio就可以跑，文档案例齐全，省心省事。OpenGL有多麻烦？网上一搜，被告知需要glew；想在不同系统下快速搭建开发环境？再去找找glfw。搜个教程，又要求glut。开始写代码了，OpenGL里面没有内置数学库，...

---

很长时间没完整的看完一本技术书了，之前虽做了两年Unity，学些Shader，但缺乏对GLSL和HLSL的了解，现在中国手游热，自己现在也在从事移动端游戏引擎的相关工作，对GLES的了解也是必须的。这书虽然不十分出彩，但知识点都涵盖了，也并不全是罗列API。是有不小收获的，当你了解...

---

我就说一处： glShaderSource 最后一个参数 length，看看这货怎么翻译的  
“如果length为NULL，着色器字符串将被认定为空。”啥意思？再看官方原版 “If length is NULL, each string is assumed to be null terminated.”  
请不好糟蹋好书了好吗？

---

第十一章， fragment operations stencil buffer testing 的例子 test0  
的case的注释是不是写错了 应该是 $(0x7 \& 0x3) < (0x1 \& 0x3)$  写成了 $(0x7 \& 0x3) < (0x1 \& 0x7)$

---

[OpenGL ES 3.0编程指南 下载链接1](#)