

# Windows 游戏编程学习路线

郑华

2015 年 11 月 16 日

一个游戏程序员的学习资料

## 1 编程部分

### 1. 熟练掌握

- -C++,STL
- -WindowsAPI
- -windows 编程
- -数据结构和常用算法
- -设计模式

### 2. 熟悉使用

- -计算机图形学
- -熟悉 DirectX 或 OpenGL
- -研究任意一款开源图形引擎
- -熟悉网络编程
- -lua/python 脚本语言

## 2 三维图形学

搞三维图形学首先还是要扎扎实实的先看解析几何、线性代数、计算几何的教材，后面的习题一个都不能少。国内数学书还是蛮好的。苏步青大师的《计算几何》称得上具有世界级水准，可惜中国 CAD 的宏图被盗版给击垮了。现在是我们接过接力棒的时候了。It's time!

《Computer Graphics Geometrical Tools》《计算机图形学几何工具算法详解》算法很多，纰漏处也不少。

《3D Math Primer for Graphics and Game Development》浅易，可作为三维数学的“速食”。

《Mathematics for 3D Game Programming & Computer Graphics》第二版比上面那本深入一些，证明推理的数学气也浓一些，可作为专业的数学书与编程实践一个过渡的桥梁吧。内容涉猎也广，射线追踪，光照计算，可视裁剪，碰撞检测，多边形技术，阴影算法，刚体物理，流体水波，数值方法，曲线曲面，还真够丰富。

《Vector Game Math Processors》想学 MMX,SSE 吗，那就看它吧，不过从基础讲起的，要耐心哦。

### 3 DirectX

《Introduction to 3D Game Programming with DirectX 9.0》DirectX 入门的龙书，作者自己写的简单示例框架，后面我干脆用 State 模式，把所有例子绑到一块儿去了。

《Beginning Direct3D Game Programming》作者取得律师学位后变成了游戏程序员，真是怪也哉。本书虽定位为入门级书，内容颇有独特可取之处。它用到的示例框架是 DXSDK Sample Framework，而不是现在通行的 DXUT。要想编译有两种办法吧，一是自己改写成用 DXUT 的。二是找旧的 Sample Framework。我又懒得为了一个示例框架下载整个早期版本的 DirectX，后面在 Nvidia SDK 9.5 中发现了。

《Advanced Animation with DirectX》DirectX 高级动画技术。骨骼系统，渐变关键帧动画，偶人技术，表情变形，粒子系统，布料柔体，动态材质，不一而足。我常常在想，从三维创作软件导出的种种效果，变成一堆 text 或 binary，先加密压缩打包再解包解压解密，再用游戏程序重建一个 Lite 动画系统，游戏程序员也真是辛苦。

### 4 OpenGL

《NeHe OpenGL Tutorials》虽是网络教程，不比正式的书迹，本来学 OpenGL 就不过是看百来条 C 函数文档的工夫吧，如果图形学基础知识扎实的话。

《OpenGL Shading Language》OpenGL 支持最新显卡技术要靠修修补补的插件扩展，所以还要配合《Nvidia OpenGL Extension Specifications》来看为上。

《Focus on 3D Models》《Focus on 3D Terrain Programming》《Focus on Curves and Surfaces》顾名思义，三本专论，虽然都很不深，但要对未知三维模型格式作反向工程前，研读 Geomipmapping 地形算法论文前，CAD 前，还是要看看它们为上，如果没从别处得过到基础的话。

### 5 脚本

先看《Game Scripting Mastery》等自己了解了虚拟机的构造，可以设计出简单的脚本解释执行系统了。再去查 Python , Lua , Ruby 的手册吧，会事半功倍。

《Programming Role Playing Games with DirectX 8.0》一边教学一边用 DirectX 写出了一个 GameCore 库，初具引擎雏形。

《Isometric Game Programming with DirectX 7.0》三维也是建立在二维的基础上，这就是这本书现在还值得看的原因。

《Visual C++ 网络游戏建模与实现》联众的程序员写的，功力很扎实，讲棋牌类游戏编程，特别讲了 UML 建模和 Rotional Rose。

《Object-Oriented Game Development》套用某人的话：“I like this book.”

## 6 Shader

要入门可先看《Shaders for Game Programmers and Artists》讲在 RenderMonkey 中用 HLSL 高级着色语言写 Shader. 再看

《Direct3D ShaderX : Vertex and Pixel Shander Tips and Tricks》用汇编着色语言，纯银赤金。

## 7 三大宝库

《Game Programming Gems》我只见到 1-6 本，据说第 7、8 本也出来了？附带的源代码常有 bug，不过瑕不掩瑜，这套世界顶级游戏程序员每年一度的技术文集，涉及游戏开发的各个方面，我觉得富有开发经验的人更能在其中找到共鸣。

《Graphics Gems》全五本图形学编程 Bible，看了这套书你会明白计算机领域的科学家和工程师区别之所在。科学家总是说，这个东西在理论上可行。工程师会说，要使问题在  $\log N$  的时限内解决我只能忍痛割爱，舍繁趋简。

《GPU Gems》出了二本 Nvidia 公司召集图形学 Gurus 写的，等到看懂的那一天，我也有心情跑去 Siggraph 国际图形学大会上投文章碰运气。

## 8 游戏引擎编程

《3D Game Engine Programming》是 ZFXEngine 引擎的设计思路阐释，很平实，有太多惊喜。

《3D Game Engine Design》数学物理的理论知识讲解较多，本来这样就够了，还能期待更多吗？

## 9 人工智能

《AI Techniques for Game Programming》讲遗传算法，人工神经网络，主要用到位数组，图算法。书的原型是根据作者发表到 GameDev.net 论坛上的内容整理出来的，还比较切中实际。

《AI Game Programming Wisdom》相当于 AI 编程的 Gems。

《PC 游戏编程 (人机博弈)》以象棋程序为蓝本，介绍了很多种搜索算法，除了常见的极大极小值算法及其改进-负极大值算法，还有深度优先搜索以外。更提供了多种改进算法，如：Alpha-Beta, Fail-soft alpha-beta, Aspiration Search, Minimal Window Search, Zobrist Hash, Iterative Deepening, History Heuristic, Killer Heuristic, SSS\*, DUAL\*, MFD and more. 琳琅满目，实属难得。

## 10 反外挂

《加密与解密 (第二版)》看雪论坛站长段钢破解序列号与反外挂有关系么？不过，世上哪两件事情之间又没有关系呢？

《UML Distilled》Martin Fowler 很多人直到看了这本书才真正学懂 UML。

Martin Fowler 是真正的大师，从早期的分析模式，到这本 UML 精粹，革命性的重构都是他提出的，后来又写了企业模式一书。现在领导一个软件开发咨询公司，去年 JavaOne 中国大会他作为专家来华了吧。个人网站：[MartinFowler.com](http://MartinFowler.com)

## 11 设计模式三剑客

《Design Patterns Elements of Reusable Object-Oriented Software》

《Design Patterns Explained》

《Head First Design Patterns》

重构三板斧：

《Refactoring : Improving the Design of Existing Code》

《Refactoring to Patterns》

《Refactoring Workbook》

软件工程：

《Extreme Programming Explained : Embrace Change》第二版其中 Simplicity 的 Value 真是振聋发聩，这就是我什么都喜欢轻量级的原因。

《Agile Software Development Principles, Patterns, and Practices》敏捷真是炒得够火的，连企业都有敏捷一说，不过大师是不会这么 advertising 的。《Code Complete》第二版名著。

数学：

《数学，确定性的丧失》M. 克莱因原来数学也只不过是人类的发明与臆造，用不着供入神殿，想起历史上那么多不食人间烟火的科学家（多半是数学家），自以为发现了宇宙运作的奥秘，是时候走下神坛了。

物理：

《普通物理学》第一册 += 《Physics for Game Developers》物理我想就到此为此吧，再复杂我可要用 Newton Engine, ODE 了，等待物理卡 PPU 普及的那天，就可充分发挥 PhysX 的功效了，看过最新的《细胞分裂》游戏 Demo 演示，成千上万个 Box 疯狂 Collide，骨灰级玩家该一边摸钱包一边流口水了。

## 12 开源代码

**Irrlicht** 著名的鬼火引擎，从两年前第一眼看到它，这个轻量级的三维图形引擎，就喜欢上了它。源代码优雅，高效，且不故弄玄虚。值得每个 C++ 程序员一读，并不限于图形编程者。它的周边中也有不少轻量级的东西。如 Lightfeather 扩展引擎，ICE、IrrlichtRPG、IrrWizard. 还有 IrrEdit、IrrKlang、IrrXML 可用。（可能是为了效率原因，很多开源作者往往喜欢自己写 XML 解析库，如以上的 IrrXML 库，即使有现成的 tinyXML 库可用。这真会让 tomcat 里面塞 Axis，Axis 里面塞 JUDDI，弄得像俄罗斯套娃玩具的 Java Web Service Coder 们汗颜。）

**OGRE** 排名第一的开源图形引擎，当然规模是很大的，周边也很多。除了以 C# 写就的 OgreStudio ,ofusion 嵌入 3DS MAX 作为 WYSWYG 式的三维编辑器也是棒棒的，特别是其几个场景、地形插件值得研究。以至于《Pro OGRE 3D Programming》一书专论其用法。搜狐的《天龙八部》游戏就是以其作为图形引擎，当然还另外开发了引擎插块啦。我早知道 OGRE 开发组中有一个中国人谢程序员，他以前做了很多年的传统软件编程。有一次天龙八部游戏的图形模块的出错信息中包含了一串某程序员的工作目录，有一个文件夹名即是谢程序员的英文名，我据此推断谢程序员即是搜狐北京的主程。看来中国对开源事业还是有所贡献的嘛，王开源哥哥的努力看来不会白费！不过我侦测的手法也有些像网站数据库爆库了，非君子之所为作。

**RakNet** 基于 UDI 的网络库，竟还支持声音传输，以后和 OpenVision 结合起来做个视聊程序试试。

**Blender** 声誉最盛的开源三维动画软件，竟还带一个游戏引擎。虽然操作以快捷键驱动，也就是说要背上百来个快捷键才能熟练使用。但是作为从商业代码变为开源之作，威胁三维商业巨头的轻骑兵，历经十年锤炼，代码达百万行，此代码只应天上有，人间哪得几回看，怎可不作为长期的源码参考？风魂二维图形库。云风大哥的成名之作。虽然不代表其最高水平（最高水平作为商业代码保存在广州网易互动的 SVN 里呢），但是也可以一仰风采了。圣剑英雄传二维 RPG。几个作者已成为成都锦天的主力程序员。锦天的老总从一百万发家，三年时间身价过亿，也是一代枭雄了。这份代码作为几年前的学生作品也算可以了，因为一个工程讲究的是四平八稳，并不一定要哪个模块多么出彩。反正我是没有时间写这么一个东东，连个美工都找不到，只能整天想着破解别人的资源。

**Boost** C++ 准标准库，我想更多的时候可以参考学习其源代码。

**Yake** 我遇到的最好的轻量级游戏框架了。在以前把一个工程中的图形引擎从 Irrlicht 换成 OGRE 的尝试中，遇到了它。OGRE 的周边工程在我看来都很庸肿，没有完善文档的情况下看起来和 Linux 内核差不多。不过这个 Yake 引擎倒是很喜欢。它以一个 FSM 有限状态机作为实时程序的调度核心，然后每个模块：物理、图形、网络、脚本、GUI、输入等等都提供一个接口，接口之下再提供到每种具体开源引擎的接口，然后再接具体引擎。通过这样层层抽象，此时你是接 Newton Engine,ODE 还是 PysX 都可以；是接 OGRE,Crystal Space 还是 Irrlicht 都可以；是接 RakNet 还是 LibCurl 都可以；是接 Python, Lua 还是 Ruby 都可以，是接 CEGUI 还是 others，是接 OIS 还是 others（呵呵，记不起来 others）都可以。所以 Yake 本质上不是 OGRE 的周边。虽然用 Neoengine 的人都倒向了它，但是现在版本还很早。特别是我认为，学习研究时一定要有这样的抽象之抽象，接口之接口的东西把思维从具体的绑定打开，而开发时抽象要有限度的，就像蔡学镛在《Java 夜未眠》中讲的，面向对象用得滥也会得 OOOO 症（面向对象过敏强迫症）。

**Quake Doom 系列** 据说很经典，卡马克这种开源的黑客精神就值得赞许。把商业源代码放出来，走自己的创新之路，让别人追去吧。不过 Quake 与 Unreal 引擎的三维编辑器是现在所有编辑器的鼻祖，看来要好好看看了。

**Nvidia SDK 9.X** 三维图形编程的大宝库，这些 Direct3D 与 OpenGL 的示例程序都是用来展示其最新的显卡技术的。硬件厂商往往对软件产品不甚在意，源代码给你看，东西给你用去吧，学完了还得买我的硬件。Intel 的编译器，PhysX 物理引擎大概也都是这样。Havok 会把它的 Havok 物理引擎免费给别人用吗？别说试用版，连个 Demo 都看不到。所以这套 SDK 的内容可比 MS DirectX SDK 里面那些入门级的示例酷多了，反正我是如获至宝，三月不知愁滋味。不过显卡要 so-so 哦。我的 GeForce 6600 有两三个跑不过去，差强人意。

## 13 网站

**www.CSDN.net** 程序员大本营吧，软文与“新技术秀”讨厌了点，blog 和社区是精华之所在。

**www.gamengines.com** 3D 游戏引擎网，专注于 3D 游戏引擎相关的资料和技术网站

**www.GameRes.com** 游戏程序员基地，文档库中还有点东西。投稿的接收者 Seabug 与圣剑英雄传的主程 Seabug 会是同一个人吗？一个在成都锦天担当技术重担的高手还有时间维护网站吗？我不得而知。

“何苦做游戏”网站 名字很个性，站长也是历尽几年前产业发展初期的艰难才出此名字。

**www.66rpg.com** 二维游戏图片资源很多，站长柳柳主推的 RPGMaker 软件也可以玩一玩吧，但对于专业开发者来说不可当真。

**www.GameDev.net** 论坛中有不少热心的国外高手在活动。

**www.SourceForge.net** 不用说了，世界最大的开源代码库，入金山怎可空手而返？看到国外那些学生项目动不动就像模像样的。（DirectX 的雏形就是英国的学生项目，在学校还被判为不合格。）

**www.koders.com** 源代码搜索引擎，支持正则表达式，google Lab 中也有。当你某种功能写不出来时，可以看一下开源代码怎么写的，当然不过是仅供参考，开源代码未必都有产品级的强度。说到 google，可看《Google Power Tools Bible》一书，你会发现，google 的众多产品原来也有这么多使用门道。