

## 课后阅读 | 游戏开发工程师学习路径（上）

2018-08-18 蔡能

从0开始学游戏开发

[进入课程 >](#)



讲述：蔡能

时长 09:46 大小 4.48M



整个开发流程和内容我已经讲完了。你现在已经不是一个完全零基础的小白了。作为一个满心想要进入游戏开发行业的人，或者想要在游戏行业深耕，你可能要问了，后续的路，该怎么走呢？今天，我就来和你聊一聊这个内容。

我会通过这份看似很长的必读书单，先给你搭建一个学习路径，然后从底层知识和游戏开发这两方面分别讲讲，想成为游戏开发工程师，应该掌握哪些知识。

我在专栏里讲了很多具体的操作，作为初学者，把这些彻底掌握，其实已经差不多了。但是想要进阶，成为一名真正的游戏开发工程师，你还有很长的路要走。从业二十余年，我见过不少的年轻人，抱着对游戏的一腔热血，一头扎进游戏开发，想着自己有一天也能开发出一款爆款游戏，从此“走上人生巅峰”。

然而走着走着，发现要学的东西太多，无从下手，直接放弃；或者学了之后，因为一直写不出什么项目，也放弃了。就好像去少林寺学武功的人，方丈让他先捆着沙包登山、挑水，他以为这并不是学武术，因此就放弃了。

事实上游戏开发，甚至编程都是这样，你首先要学好基础知识，有了扎实的基础，才可能去做更多的事。道理我想你都懂，我就不多啰嗦啦。我们进入正题。

还记得，在专栏最开始的时候，我专门写过[游戏的分类](#)，按呈现效果，我们可以粗略分为 2D 游戏和 3D 游戏；按硬件，我们可以分为主机游戏、电脑游戏、移动游戏和电视机游戏。但是，如果从真正的开发角度，我会把游戏开发分为**原生游戏开发**和**网页游戏开发**。

为什么这么分呢？首先，我这个分类中，并没有把网络服务器端的内容放进来，而只是针对游戏客户端部分。我将它分为原生客户端（PC、手机非 HTML5 原生 App）和网页客户端（PC 端页游、HTML5 游戏、手机端 HTML5 游戏、微信小游戏）。

我们先从原生客户端讲起。

在专栏里，我通过 Pygame 做了示例，但是碍于你可能不仅是个游戏开发新手，还可能是一个编程新手，而我们专栏的侧重点在于帮你认识、了解游戏开发流程，知道游戏开发是怎么回事，对于底层的知识，我其实没有过多的去介绍。

我本身是写服务器端的，对于底层知识研究了很多，所以，我深知底层知识对于一个真正想要在游戏开发之路走得长远的人的重要性。学习的过程，我没有办法替你做，但是我想应该有那么一些经验还可以跟你讲一讲，不至于你在一些可以避免走弯路的地方，浪费掉许多时间。

开发原生客户端游戏需要非常系统的底层知识。这里的底层知识指的是计算机（移动设备）的硬件底层知识，比如内存、CPU、图形显卡等等。当然如果要学得全面，你还可以学习物理引擎、音频等等。

**想走捷径，去学 C++ 吧！**

如果你的学习时间并不充裕，或者非常迫切地想知道“底层是什么，底层做了点什么”，想相对快速地能学习到底层知识，那么我推荐你直接学习 C++。

首先从执行效率看，C++ 的执行效率仅次于汇编，约等于 C。如果要编写更底层的操作系统，因为语法糖太多，你不知道编译器在背后做了什么事情，C++ 语言将无法“直接”掌控底层设备。所以要写操作系统，需要对 C++ 编译器进行针对性改造。

我们再往上看，如果要编写更上层的应用，比如桌面应用、动态网页应用，这些是 C++ 经历过的“银弹”时代。90 年代到 2007 年左右，几乎电脑上所有应用都有 C++ 的身影，但是现在，桌面应用完全可以使用 C# 来编写。

网页端就更不用说了，PHP、JavaScript、Python 每一个都能比 C++ 做得更快、更完美。但是在游戏开发中，既需要执行速度，相对地，又需要开发速度，综合下来，只有 C++ 最适合。

所以，学习 C++，会让你对底层知识有了一个大致的了解，但是如果你没有更进阶地学习 C 或者汇编，那么你的知识面就停留在类、内存分配、对象等层面。

作为二十多年的 C++ 程序员，我这里有几个学习 C++ 的心得想跟你分享。

首先，C++ 和 C 语言并不是一门语言。它们之间的语法结构看起来虽然“相似”，但 C++ 相对于 C 更像是瑞士军刀和匕首之间的关系。它们都有刀的功能，而 C++ 除了刀的功能，还有更多的功能。

其次，C++ 是做减法而非加法。把 C++ 所有特性学完会耗费你极大的精力，注意是极大，不是大量。每一种特性背后都包裹着无数语法糖衣和编译器的“自动化”动作，使用 C++ 的人当中，有一部分人经常会“走火入魔”，专门研究一些奇技淫巧的技巧。

所以最好的方法是，**不要执着于语言特性**，因为两个不同的编译器，可能编译结果都会不同，比如一个会出错，一个会通过，但只要略作修改，就能运行。只要不影响具体最终软件的执行，就可以了。C++ 特性太多，选择一种或者两种特性来编写整个软件，才是发挥 C++ 最大功效的方法。

最后，大道至简，这对于整个产品、架构、编码，都是一句适用的箴言。当然对 C++ 也是如此，只有用最简略的方法来编写代码，才会让代码快速成型。C++ 比 C 多了太多快速搭建的语法功能。尽量利用它的优点（比如面向对象、模板等等），就能发挥出语言最大的特长。

## 走这个路径，基础更扎实！

如果你有较宽裕的时间学习编程，也希望学习到更多的底层知识，对计算机有一个全面的掌握和了解，那么我推荐你用以下的这个更全面的路径来学习。

以我的经验，我建议你先去学习一下 C 语言。为什么呢？因为 C 语言是汇编的封装，C 语言的一句语句，可以是汇编的几条指令，虽然每个 C 编译器最终生成的汇编指令可能不同，但大致方向是一致的。

其次，学习 C 语言能让你从基础的语言语法了解计算机编程是怎么回事。C 语言是 99.9% 的电子设备的基础语言（剩下的 0.01 是汇编）。

学习 C 语言有太多的入门书。在这里我推荐 [《C Prime Plus》](#)，绝对的 C 语言入门经典教材。[《C Programming Language》](#)，读完《C Prime Plus》之后再去读这本书，绝对会让你了解什么是高手所写的书。

看完了《C Programming Language》之后，你除了是 C 程序员之外，俨然是一个不错的 Linux 用户了。


学习完 C 语言后，你可以学习汇编语言。有了 C 语言的基础，对编程、地址操作、位操作这些基础，学习汇编就会相对容易。

估计看到汇编语言这几个字，很多人就望而却步了。但是，我想说，**没有比汇编更适合学习底层知识的语言了。**

毕竟在现实中，除非你编写的是芯片或者操作系统（现在编写芯片或者操作系统也大量使用 C 语言），只有占那 1% 的启动部分代码会使用汇编。这么看，使用汇编编程的人真的非常少，而且它本身也不适合编写大型项目。

你可能要问了，为什么你还推荐我从汇编开始学呢？学习汇编的意义在哪里？

我们先来看一段指令。

 复制代码

```
1  pushl %ebp
2
```

```
3  movl %esp,%ebp
4
5  subl $8,%esp
```

在汇编里，这些指令就代表着 CPU 的指令操作。在编写中级、高级语言，需要调试代码的时候，如果没有调试信息，最终都会落到汇编语言这里。这时候，汇编语言就是你的最后一道防线，学习了汇编，你就能快速、深入地定位程序的问题所在。

所以，**学习汇编的意义就是，你能在脑中完全掌握电脑的运行规律和运作逻辑**，比如内存寻址操作、寄存器操作、加减操作、CPU 指令等等。

学习汇编有一本书，是绝对的入门级好书，能够把深奥枯燥的机器知识讲得生动活泼，那就是王爽的[《汇编语言》](#)。

学习了汇编语言，也会对你的 C 语言知识有一个本质上的巩固。所以结合 C 和汇编，指针不再是头疼的问题，因为你已经知道在汇编中的地址是如何操作的。

C 语言看起来就好像把汇编进行了一次朴素的包装，你不再需要强制记忆寄存器和指令，定义一个变量就知道汇编语言是怎么做的，这对于你的 C 语言来说，是有绝对帮助意义的。

**一般人提倡的学习路径是由浅入深，如果你想一直在技术领域深耕下去，我建议你走一条由难到易的学习路径。**

因为，越高级的编程语言越接近人类的思维方式，这种思维会固化你的编程思维，让你在之后的学习过程中，无法更深层次地理解底层机器语言的沟通方式，学习起来就会比较困难。先学习底层知识，这样你在后续的高级语言的学习中，就属于“降维打击”，学习起来就相对轻松和快速。

除此之外，如果你已经是一名程序员，你应该很能体会，你的工作将会越来越忙，几乎没有时间学习新的知识，而学习能力是一名程序员必备的基本能力。

接下来，你还是需要学 C++ 语言了。具体原因其实我刚才也说了，游戏引擎绝大部分使用 C++ 编写，也有一小部分是使用 C 语言或者汇编编写的，比如 Allegro。



如果你要学习 C++ 语言，学苑出版社出版的 HerbertSchildt 写的《C++ 从入门到精通》，这本书一定不能错过，它是我的 C++ 启蒙书。唯一可惜的是，这本书年头有些久了，我的书还没有找到，但是我至今对书的内容都记忆犹新。这本书后面的小习题，绝对能让你在学完这本书之后，对 C++ 了解的非常透彻和深刻。

另外，[《深度探索C++ 对象模型》](#) [《C++ Programming Language》](#) [《C++ 语言的设计与演化》](#)也是必看书籍。前一本能让你了解 C++ 对象模型，从底层了解 C++ 的机制。后两本出自 C++ 之父 Bjarne Stroustrup 之手。

从 C++ 开始设计到现在，为何会如此演化和设计，语言的背后又是怎样的故事，除了语言的基础知识，也会有一些小小的故事。另外，如果你的知识面和允许，我建议你最好去看《C++ Programming Language》的原版，因为只有原版是原汁原味，不会有被翻译偏差影响的。

现在，我们已经将一幢大楼的“地基”都搭建好了。

## 学完这些，要多久啊？

你可能会问了，从开始学习底层知识，到对知识有了解并且能顺利写出软件需要多久呢？因人而异，有人是计算机系的，有人是野路子，有人对计算机特别有天赋，有人虽然不聪明但是非常勤奋，每个人的学习曲线和学习路径以及最终学会的时间都是不一样的。

但是一般来说，要从 C 到汇编再到 C++，一直到能编写成熟的代码，平均下来需要 5~7 年的时间。我没有做过精细地统计，但是我想，按照大学四年的学习时间，从学习到实践，再加上毕业 1~3 年的实践深入，5~7 年的时间基本是差不多的。你觉得呢？

今天这些内容主要涉及游戏开发的底层知识。说起来很容易，做起来却一点都不容易。有了这些基本功，下一节，我会推荐一些游戏原理和机制相关的书籍。希望我的分享对你有帮助。

---

# 从0开始学游戏开发

你的游戏开发入门第一课

蔡能

原网易游戏引擎架构师  
资深游戏底层技术专家



新版升级：点击「👤 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 第36讲 | 如何制作游戏内容保存和缓存处理？

下一篇 课后阅读 | 游戏开发工程师学习路径（下）

## 精选留言 (6)

写留言



淡看烟雨\_

2018-08-27

1

我是IC行业的，做嵌入式系统底层开发十年了。本身有些文字底子，上学时候喜欢跑团，做设定，喜欢魔戒，也有些素描底子。现在三十三了，想尝试做游戏。其实心里也有些打鼓，有几个问题自己也没想明白，第一，我现在肯定没有年轻时候的自己学的快了，不知道能不能挺过初期陡峭的学习曲线。第二，现在年轻人们喜欢什么？我不知道，也可能大家都不知道。我能做出令一部分人喜欢的东西吗？第三，故事性强，必然占时间，走上...  
展开



风雨无阻

2019-02-28

1

老师，游戏中3d地图如何拆分成几小块呢？

展开 ▾



**Veel**

2019-02-28



你好我是游戏UI想转客户端开发..做一些个人项目，比如休闲的小游戏，棋牌，该是怎样的学习路径



**田森、很忙**

2018-09-04



如果想做棋牌类的游戏，类似斗地主这样。u3d + skynet 是不是学习曲线会平缓一些。本身是做网页后端的小程序员。

作者回复: 客户端推荐用h5的，服务器端最好用成熟的引擎，skynet属于个人作品，特殊性比较多，推荐用成熟公司引擎，比如nodejs，或者直接用Python写也可以



**zc**

2018-08-20



学苑出版社出版的 HerbertSchildt 写的《C++ 从入门到精通》，这本书确实买不到了，有可以替换的入门书吗？

作者回复: effective c++可以试试



**湛卢**

2018-08-18



别的语言转c++呢

展开 ▾

作者回复: 重新，系统的学比较好





