**张子豪**

2021-12-19 10:8:59

【问】

请您回忆一下最近或者之前学习某个工具、软件或技术的经历，从听说这个技术开始，到真正学会用它去完成一些开发工作，整个过程是怎样的？

给您提供两个两个工具啊，第1个是我在学习就拍上了。

【答】

给您提供两个工具。一个是我在最开始阶段学习python爬虫的时候呢，会在教程里面看到python可以做爬虫，那么我就去了解了，了解了一下爬虫是什么东西，接下来呢，就去根据教程所提供的一点小小的示例去实现一个简单的功能，但是呢它仅限于这个示例内部的提供的这个功能，然后对于实际应用场景呢，还是不太契合的，到后来嗯最近在工作中会用到一些爬虫的一些需求吧，然后我重新开始认真学习了一下。我在学习的过程呢，主要就是从需求的处罚，先把需求整理清楚了，再把这个需求的抽象成一些实现的这些功能点，然后根据功能点再去搜索。所以主要借助的还是搜索，就是百度或者是谷歌这种搜索的东西，那么主要是因为这个爬虫技术本身它提供的这些框架或者是这些库，都是比较标准，而且是没有特别多的难度，所以说直接通过搜索呀，或者是从示例上直接去获取它的用法，还是比较简单，所以说经过大概也就一两周的时间就可以基本的能够满足的工作需求了啊。所以这个是比较简单的一个案例。然后第2个呢是在学习，因为我的工作是跟游戏相关的，然后我在第2个阶段学习计算机图形学，或者叫图形编程的一个例子吧。图形编程里面涉及到对CPU的编程，对CPU的编程它可以抽象为几套API。它实现的是同样的功能，但是它的实现方式是不一样的，那么我在最开始学习HLSL，就是微软提供的一套对GPU的编程语言的时候，发现它缺少比较友好的新手的文档，比如说新手在缺少一定基础的情况下呢，学习他这个文档是特别费劲的，也就很难去入门，而且它的文档缺少示例，也就是说他只给你提供了基础的语法，其他的在上层次或者逻辑层面的应用呢，还是需要你自己去算。然后在整个的互联网上啊，或者是各个技术社区里面对于这个hl SL这一门CPU语言的应用啊或者是教程的话不是特别多，而且也是相对来说有一定门槛，那么在这种情况下呢，嗯，考虑到这种难度的原因，所以说当时我就换了另外一门的API，叫open GL。Open GL它提供了比较友好的，而且是对新手友好的入门的课程，那门槛就比较低，这门课程他从零开始，从回执一个黑色的窗口到绘制一个很简单的三角形，到最后可以导入自己的模型，然后显示一个相对来说比较完整的一个游戏场景，从0到有的这样一个过程来说，会对新手提供了一个比较好的教学。然后我通过这一轮教学呢，就基本了解了图形学的一些概念和用法，那么可以给自己简单搭建了一个游戏的引擎，或者是游戏的开发工具，然后对整个图形学呢也有一个初步的了解和认识，那么接下来在工作里面呢，也运用了这些我从open GL这个教程和相关的一些书籍里面学到的知识，能够把它真正的运用在日常的工作里面或者是开发中。大概也是这些，我不知道您对这边的细节有哪些疑问。

【问】

学这个open GL的时候除了看他的教程，那会看文档吗？他会提供文档吗？

【答】

嗯，这个文档算是他的教程一部分，那么文档可能是它属于示例类型的，或者是入门类型的，其他还有官方的一些语法的文档或者是官方的说明，他也有，但是我看的很少，因为我刚才提到的第1个hl SL，就是open GL的官方文档部分，和hl SL的语法说明部分是比较相像的，都是对于新手不是特别友好的，所以说当时看的最多的还是open GL的这个入门的过程。

【问】

入门的视频教程是吧？

【答】

不对，是嗯，是文字的教程。

【问】

文字的教程啊啊，文字的教程的话，应该也算是它的文档的一部分，嗯，那您可以简单的说一下这个教程它都有哪些部分组成的？就它一般都会讲什么东西？

【答】

从功能上来说，就是像刚才我也提到它实现了，你在这一套API可以实现的比大部分的功能，然后这个基础功能啊，可以画一个窗口出来，从什么都没有的窗口，到最后你可以做一个简单的游戏场景，还有一些比较复杂的一些特性啊，可以实现这样的功能，那么每一篇文档里面他会提供你学习这篇文档首先你必备的基础知识，也就是说你如果没有这个基础，你就看不懂这篇文档，他会明确的告诉你，我觉得这一点是比较好的。然后其次呢，它会在每一步它所涉及的API或者是提供的一些术语，他会对其进行解释，这一点是特别好的。在阅读其他的文档的时候也会涉及到这种问题啊，就是你这个术语，术语规定不规范的话，那么给我们学习的人来说可能就会增加一些我们没有必要承受的学习成本啊，那么这第2点。然后第3点呢，就是他会在这一篇文档中，在做到某一个阶段的时候，他会给你做一点总结，总结之余还会给你提供额外的一点点示例，就说对这个阶段的小示例，所以说他整个一篇文档是一个大的示例，但是它会在每一个阶段提供一点点小示例辅助你去强化你的记忆，然后另外会做一些扩展，我会给你解释你这个术语，我从这个字面意思给你解释清楚了以后呢，那我会从原理上给你拓展一下，当然你不看这个解释也可以啊，看你这个术语文字解释就够了，但是他会给你做一些拓展，然后帮助你从底层去理解这些他提供的这个术语的意思。然后接下来之后呢是会，那我觉得做得最好的一点就是说他的示例，因为他示例大概分了，按照目录结构可能分了三十几篇，那么在每一篇之间呢，它是有很强的联系性的，就是说他的这个逻辑是没有发生断层的，或者说很少发生断层，你学了上一篇以后就可以直接进入下一篇，嗯，大概是这几个方面我是觉得挺好的。

【问】

嗯，那您可以把这个学习Open GL这个教程的过程分一下阶段吗？就分成

学习的前期中期和后期，然后请您分别说一下各个阶段，您学这个教程的一个特点吧，学习的特点，学习的目标，就分一下阶段。

【答】

你是说从我个人学习的角度来说，对吧？

【问】

对对。

【答】

前期来说，首先是知道这个叫什么，就是知道有open Gl这样一套API，或者一套框架，然后获取到有教程这样一个信息的时候呢，会接触到很多他所给你提供的那种准备知识，或者说，你需要首先掌握一些只是。那么在学习这些知识的过程中发现也是有一定困难的，但是呢，把这些知识完成学习以后，就可以真正的进入他的正式学习了，那么第1个过程呢，其实是不属于他这个教程的给你提供的这个内容里面，还是需要你自己去学习完成的。那么完成了这个阶段了以后呢，这个阶段大概花了几天时间吧，也不太多，然后在第2个阶段也就是说正式开始学习的时候呢，首先要了解一下open Gl这套API他的构成，或者说它写作的逻辑，那这一套比较抽象的这一层概念呢，其实是open GL这个教程有一点点欠缺的，他对于里面的一些使用逻辑并没有过多的给你介绍，而是直接给你按照他的代码，你就可以直接挑出来一个运行的结果，但是对于这个使用逻辑没有做过多介绍。就比如说里面有一个使用的术语叫绑定，那个绑定这个词呢，其实是翻译过来的，那么在英文里面是叫bind，但通过绑定和bind是没有办法很好的理解它真正的使用功能，后来通过大量的查阅中英文的这些问题啊，或者是博客，才发现它是有一种特定的功能，如果你没有很好了解这个功能的话，可能对于整个API的使用还是会有一定的困难。那么这个是我集中举的一个例子。那在第2个阶段呢，或者叫初识open GL的阶段，会遇到这些在逻辑层面或者使用层面的一些问题，那么把这些问题都排查掉了以后呢，就可以很顺利的进行到open GL的正式的学习阶段，就是他给你提供哪些，我可以首先我也会预料到它在这个框架内，它处于什么样一个地位，就是在抽象层面处于什么样一个层次，那么在很好的理解到这一个方面以后呢我就可以比较顺利的进行学习了啊，这是第3个层面。第2个层面耗费的成本还是比较大的。然后第3个层面的话，因为open GL这个教程提供的内容还是比较丰富的，然后基本可以覆盖到学习的需求，所以说我觉得大概分这三个层面，差不多。

【问】

第1个阶段的话相当于是刚刚起步，就是刚刚有一个大概的了解，还没有去深入了解。然后第2个层面的话就是您开始去读他的文档了，但是也会遇到一些逻辑上技术上的问题，还有些看不懂，然后就回去查其他的博客啊之类其他的资料来读懂。然后第3个层面就是应该能够应用了是吧？

【答】

对，是的，第3个层面这样说可以是流畅的学习她这套文档，然后如果说应用的话呢还是在最后一个层面吧，就是等把这一套文档学的差不多了以后，确实可以主动去应用了。

【问】

嗯，那您可以说一下应用是怎么应用的嘛，是会参照文档里面给的去做一些功能开发是吧？

【答】

应用，嗯，因为open g2这一套API他有一些所谓函数或方法呢，它是比较难记，对CPU编程比较难记。所以说在用的时候还是需要参考他的用法。比如说它的参数啊，或者是跟其他的函数怎么配合呀这一套，如果使用时间不长的话，确实还是需要参考一下。

【问】

嗯嗯，那他也会给一些示例吧，这个文档里面。

【答】

对，是的，整个这一篇文档啊，就是一个巨大的示例，那么里面还会分小的示例，就是对某一个函数会单独给你一点代码，告诉你它是怎么使用，有什么效果。

【问】

那您去应用的时候会参照他示例给的代码吗？

【答】

嗯，对。

【问】

嗯，那你觉得这是利好用吗？

【答】

嗯，大部分来讲还是可以的，那么小部分其实也由于刚才我说第2个阶段涉及到的原因，就说他这个框架的使用逻辑当时还不是特别清楚的情况下，可能会有一些困难。

【问】

嗯嗯，好的，嗯，好下一个问题是文档的类型，您刚才提到了学教程，然后您平时用的文档里面除了教程还有其他的文档吗？

【答】

嗯，教程来说应该是属于官方提供的啊，那么除了官方提供的以外呢，刚才提供的博客呀，或者是这种问答的，都是属于用户生产的，那么这一部分也是提供了很大帮助的，但是呢，也会给你带来一些误会，因为有的时候可能你非官方生产的内容，它的正确性不太能保证。这是第2种类型。那么其他的可能就是直接查阅API。从API函数名的角度，他给你一个最直接最简单的解释，可能算是另外一种吧，但是你把他定为官方的功能文档我觉得也可以。然后再一个就是实践性质的。也可能算是用户生产的。为了完成某一项比较大或者比较复杂的功能，那么他会出一篇各种各样的用法的实践或者说经验总结。

【问】

嗯，那除了这个还有其他的吗？

【答】

其他的。我觉得大概可以就分这几类吧。

【问】

下面一个是关于信息来源的问题，就是您刚才提到的这些文档啊，资料啊，都是从哪得到的呢，从哪来的？

【答】

首先是通过搜索引擎，百度或者谷歌，然后再一个呢是有一些个人的分享里面，博客里面可以提到，给你整理了一系列的入门的材料。这是第2个。然后其他的呢就是它的官方文档，那么就比如说open GL，我知道open GL这个API可能是通过其他的朋友的同事口中，然后我再去访问他的官网，然后发现他有这样一套教程，大概这三个途径吧。

【问】

您说的博客是指一些像csdn这样的博客吗？

【答】

嗯，对，就目前来说可能csdn或者博客园或者几乎啊这种多一些。CSDN质量比较差，而且抄袭也比较严重，看的不是特别多，但是有的时候小问题还是会很参考一下。

【问】

那就是主要还是那个知乎还有博客园是吧？

【答】

可能知乎稍微多一点，知乎的接受度，就是对我自己观察来说可能是总结性还是稍微好一点。

【问】

嗯，好的，嗯，您还提到了搜索引擎，就是也会用搜索引擎去搜资料，那您可以举一些例子吗？就是您是怎么搜的，怎么去构建这个搜索的表达式的？

【答】

一般来说两方面。一个是需求，就说我想实现什么样的功能，那么我就会搜，就比如说python实现爬虫，就是python空格爬虫，就是最简单的，另外一方面可能就把你所遇到的问题来描述一下，我倒是觉得这个可能大家都没有太大的区别吧，在搜索方面嗯。

【问】

好的，嗯，下面一个问题是关于阅读方式的，就是您打开一篇文档以后你一般是怎么阅读的呢？

【答】

嗯，如果有目录的话，首先会看目录，看完目录以后呢，会去大概阅读了一下这个文档内部的标题，或者说从视觉的方面首先看一看整个文档的结构。如果我想要的是一个具体的问题，很具体的一个问题的解决方案的话，那我就直接在文档内部去搜索这个关键词，看这个关键词出现在哪里，如果在这个出现关键词路线的附近能够解决的话，那我其他的内容就不会再看了，除非是我遇到了其他的，会回过头去看他前面所说的一些背景或者原理之类的。那么如果是学习整个一篇文档它所表达的一个概念或者是原理的话，还是会先从结构来说，先从结构出发，然后再去一点点通读一下。

【问】

嗯，好的，嗯，那您在开发过程中应该也会遇到一些问题吧，一些bug或者是技术性的问题，那您在遇到问题的时候是怎么解决的呢？

【答】

遇到问题，如果我自己对这一套我使用的技术或者框架有足够了解的程度的话，还是会优先考虑自己排查一下，从逻辑层面或者是这个API的调用层面去排查一下这个问题。如果是新应用的这样一个技术呢，那我还是会首先尽快的去搜索或者是请教其他人，这样的话避免因为自己的这个误解啊，或者是比如使用不熟悉带来的额外的时间成本。

【问】

搜索的话就是通过搜索引擎来搜索一些别人的回答解答，是吧？

【答】

嗯嗯对。

【问】

那有会返回去查文档来看文档里面有没有解决方案的这种情况吗？

【答】

如果搜索引擎搜索到的结果不能满足的话我会再去看文档，因为搜索引擎是最快的，然后各种技术里面的官方文档啊，因为它的组织程度和结构性，或者是查阅的这个便捷性还是不一样，差异还是挺大的。所以说，在我的这个习惯里面，官方的文档就放在第2位，那就是说他便不太方便去查，你查的话也未必能查得到。

【问】

嗯，所以就优先用搜索引擎来搜。

【答】

对，也比较快。

【问】

那搜索引擎您百度谷歌都会用到吧？

【答】

对，谷歌用的多一点点。

【问】

嗯好的，下面一个问题就是关于什么样的文档才是好的开发者文档，您可以自己根据自己的情况来回答一下。可以从几个方面分几个方面来回答，比如内容啊，它的组织结构啊，以及它的这个交互设计啊，还有它的这个更新程度等等，您可以大概的说一下。

【答】

因为我这个部门呢，他也是维护了几个场景，那么这个场景也是需要写文档的，然后其实我之前在这方面也简单做了一点点功课。那么我觉得什么样是一个好文档，首先呢，最原始的出发点是它需要满足用户的需求，如果从开发者的角度来写的话，有可能跟用户的实际使用场景那是不契合的，你的产品和文档，虽然说都是开发者来做的话，我觉得是有这样的问题，所以说我们一般写文档会首先去调查一下这个用户来去用我们这个产品，它最原始的需求是什么。我们写文档，把你最原始的需求覆盖到了就可以了，其他的我就算是我们想写的，我们也不会去花时间成本去写，那么这是第1个。就是首先那个文档要符合用户的最原始使用这个产品的需求。然后第2个是内容上面的，要把文档分好几种类型，就说你想提供哪几种类型的文档给用户，分别是什么作用，比如说我提供官方的API的文档，那就是提供给用户从搜索的角度来去查阅。如果是第2个入门或者是教程类似形式的文档，完全提供给新手或者说没有基础的用户，让他去一步一步按照你所设定的这些东西去学习。然后内容的另外一个方面呢，就是官方文档的组织方式，我觉得这个也是比较重要的一个方面。在一系列的官方文档中，你的整体的结构或者是整体的层次是不是足够有逻辑性，用户能不能通过你的目录结构很快的去找到他想要的东西。然后你文档之间是不是有的逻辑是不是比较紧密？文章里面的所涉及到逻辑跟其他的文档的逻辑是不是，在内容上面或者在逻辑上面是不是一致的。因为之前经常碰到，不同的作者在写各自的文档的时候，他们的行为风格是大相径庭的。这个问题也是，嗯，我觉得是需要在内容方面做比较多的考虑。然后另外一个在文档上可能还需要考虑视觉效果，因为你的排版和布局还有格式，行文的格式也是比较重要的。就是同样一些内容，我们这边其实做过一些调查，比如说，嗯，在同样的内容你用不同的比如说HTML去写，或者说你用那markdown去写，对不同写作技能的作者来说，它的最后的效果还是不一样。我觉得大概是这几点吧。

【问】

嗯嗯，好的，那有有关一个问题是这个文档的语言的问题，就是想问一下您平时是读中文文档比较多还是英文文档比较多啊？

【答】

中文比较多，但是英文如果是有价值的还是会自己去阅读的。

【问】

那你读英文读起来觉得困难吗？

【答】

会有一点点困难。我自己学习的效率肯定是没有中文高的，有的时候会去用软件先翻译一遍，如果觉得哪里看起来不太对劲，那我就是自己去读原文，我自己是这样的。

【问】

那你有读过那种就是它原文是英文的文档，然后有人把它翻译成中文，你有读过这种翻译的文档吗？

【答】

我提到的这个open GL的入门文档，或者说教材，他其实就是这样的，原文是英文的，然后国内有一些开发者翻译得。这个阅读起来还可以，基本上来说都是比较准确的，然后如果是有一些翻译的书籍呢，英文原文的书籍翻译过来以后就读过一些比较差的，就是会对你的阅读或者是学习构成一定的影响。

【问】

嗯嗯好的，哦，那我再简单问一个问题吧，也是我们的一个研究问题就是，嗯您觉得开发者，他的学习和工作跟别的职业比起来有什么特点吗？有什么不一样的地方吗？

【答】

如果这个开发者是单纯指程序开发的话，那么它最大的特点我觉得，我觉得就是需要比较比较严密的逻辑思维，这个是比较重要的，不管是阅读，不管是自己写代码还是阅读其他人的代码，就是逻辑层面是需要严格，否则的话整个你的开发还是很难有长期来说的价值，我是这样觉得的。那么对于其他的比如说你设计人员或者是产品人员来说，当然逻辑也很重要，但是因为不是跟机器打交道，所以说可能不需要那么严格，我觉得是这样。

【问】

嗯，那这个学习的方式呢？有什么特点吗？

【答】

学习的方式按我的理解呢，那就是你如果接到了什么任务啊，或者接到了什么需求，然后你要去尽快的把这个需求转变为你想实现的功能，然后再从功能去出发，再找你自己的学习的需求，我觉得是需要经历这样几段过程对吧，然后在学习上面呢，一个是需要快速的开发出来，所以说你在学习上啊，学习的时候也需要提高效率。然后第2个是要保证质量，所以在里面学习的时候，需要选择正确的学习方法或者是学习的一些材料吧。