



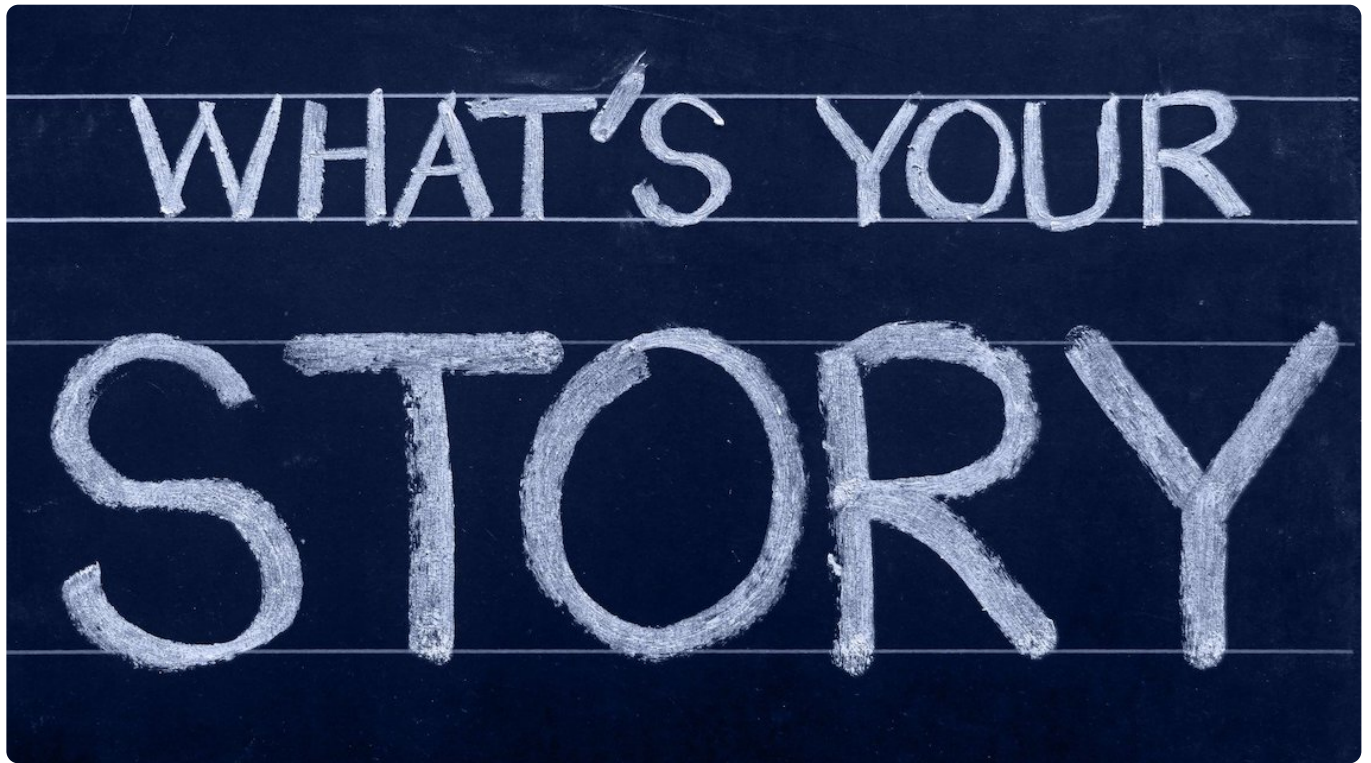
下载APP



用户故事 | 非前端开发，我为什么要学可视化？

2020-09-04 月影

跟月影学可视化

[进入课程 >](#)**讲述：月影+冯永吉**

时长 08:08 大小 7.47M



你好，我是月影。专栏已经更新过半啦，首先我想感谢仍然坚持认真学习你，很高兴见证你的成长。在这个过程中，我的课程对你有哪些帮助呢？你可以写在留言区分享给我。

今天，我想和你分享一篇罗同学的故事。罗同学每堂课都会非常认真地完成课后作业。在这里，我想对他说，很高兴能够看到你分享自己的学习故事，谢谢你能喜欢这个专栏。你是一名 Android 开发工程师，实际上我也写过 Android App，所以我们也算是半个同行。做技术的同学能够喜爱折腾技术是特别好的特质，尤其是在折腾技术的同时还能重视基础的积累，这就更难得了。这样的同学往往能成长到很高的高度，希望我的专栏能够让你在专业之路上走得更远。



好了，下面，我们一起来看看他的收获吧。

你好，我是罗乾林，是一名有着 5 年工作经验的 Android 应用开发工程师，坐标上海，很高兴能跟你分享我与可视化的故事。

讲起来还挺惭愧的，我的本职工作并不是前端开发，甚至我连 JavaScript 的语法都不怎么熟，但这并不影响我学习可视化专栏，反而除了图形学的知识以外，在阅读老师代码的过程中，我还学到了 JavaScript 的相关语法知识。因此，我希望结合我的经历，给同样非前端，正在学或者想学这门课的你一些帮助。

我为什么会学可视化？

首先，我想先说说，我为什么要学可视化，原因主要有两方面。

一方面，我的工作开发 Android 应用，在需要绘图的时候，我通常会使用内部提供的 Canvas，它底层使用的是 Skia 图形引擎，提供的绘图接口跟浏览器上的 Canvas 有些差别。这让我对原生的 Canvas 产生了好奇。

而且，平时我就是个比较喜欢折腾其他技术的人，比如 MFC 绘图的 DC、QT 的 QPainter、JavaFx 的 Canvas。这些环境都提供了一些类似的绘制基本图形的接口。但是，一直以来我只是在调用这些接口，并没有认真思考过这些绘图系统的核心是什么，对它们的理解也都不深入。

这就导致，我在绘制一个效果的时候，从一个提供了接口的绘图环境中，换到另一个没有提供接口的环境中就会不知所措。

另一方面，因为项目的原因我接触到了 OpenGL ES 绘图，而 OpenGL ES 只支持几种图元的绘制，很多图形的绘制都要自己实现。这里，我想先来简单说说这个项目。

这个项目简单来说就是开发一个自定义播放器，播放器运行在 Android 设备上。播放器从服务端获取布局文件，根据布局文件将界面划分为多个大小不等的区域，分别用来展示视频、图片、文本。其中，播放视频我们采用的方案是，将视频解码为一帧一帧的 RGB 数据并缩放到相应大小送入 Canvas 显示。

在开发过程中我发现，对一些稍大分辨率的视频文件，播放器播放起来非常卡顿。通过在网上查阅资料了解到，我可以将视频解码出 YUV 格式的数据直接送入 GPU 中展示，以此来提升播放性能。就在这个过程中，我接触到了 OpenGL ES。

刚接触一项新技术的时候，我们应该都是倾向于先写个“Hello World”找找感觉。于是，我就跟着网上的教程写了一个绘制三角形的 demo。写完后感觉整个人都不好了，因为绘制一个简单的三角形，我就写了快 90 行代码，中间涉及的 vertexbuffer 和 indexbuffer 更是看得人一头雾水，而且 OpenGL ES 提供的图元就只有点、线、三角形。这个时候，我就在想那要绘制一个圆该咋办呢？使用 OpenGL ES 完全没有我之前使用绘图系统提供的绘图接口丰富、易用。这更加深了我在可视化方面的困惑。

我与《跟月影学习可视化》专栏

因为我一直有用极客时间来学习，刚好在这个时候，我看到平台推出了“跟月影学可视化”这门课程，看了看介绍，发现就是在讲解图形学的知识，并且，OpenGL ES 和专栏中讲的 WebGL 几乎一样，这正是我想要的，所以我就立马买了下来。

开始学习专栏之后我发现，我前面遇到的这些问题，都是因为缺少图形学的基础知识。就像月影老师在 [第 5 讲](#)所说的“如果我们手中只有解决具体问题的工具，没有统一的方法论，那我们也无法一劳永逸地解决问题的根本”。

比如说，我在初中的时候，就学了点构成线、线构成面、面构成体，这些最基本的数学原理却一直不能在实践中进行应用。课程中的数学基础篇让我重新认识了数学的强大，不局限于某一图形系统提供的 API，图形学的基础知识才是根本。扩展到其他计算机知识都是这样，现在新技术层出不穷，让人眼花缭乱，而其最本质的知识都是不变的。

而且，随着学习的深入，我把之前有想过但并没有深入思考过的问题都发掘了出来，并且也都有了答案。比如说，判断点是否在区域内。我平时在绘图的时候，大部分是判断点是否在一些规则图形内部，即使碰到了一些不规则图形，也是使用现成的 API 判断，如 Android 提供了 Path、Region，浏览器 Canvas2D 的 isPointInPath 等。

接着，[第 8 讲](#)让我了解到，图形学领域有三角剖分这样成熟的技术来解决这类问题。第 14 讲和第 19 讲，又解开了我困惑已久的，如何使用着色器实现几何造型和动画的问题。

在平时的开发过程中，我也会简单使用仿射变换来做图形操作，适当地使用坐标变换，使我们在绘制一些图案时更加简单。而专栏中讲解的生成重复图案、分形图案及随机效果，更加让我体会到了数学知识的强大（离开学校后才意识到数学很有用）。

像学习其他软件编程的时候，我们在初期并不会觉得数学知识很重要，即使是做了几年的软件开发也未必会觉得数学知识在编程中帮助很大。而在图形学中很快就能让人意识到数学很重要，特别是线性代数，我们会经常使用矩阵变换、曲线方程、向量运算等数学原理，甚至还要用到物理学的知识，如实现动画涉及的缓动函数，实现自由落体等模拟现实世界的图形绘制。这也激发了我学习的兴趣，让我觉得这样的技术才是有门槛的。

关于学习专栏的一点建议

最后，我想给正在学习专栏的你一点建议，在绘图过程中有很多细节需要我们特别注意，如果只是听听老师的讲解，没有自己动手去尝试，那有很多细节就会留意不到，也就不会引起我们的重视。这门课程对我来说，学习起来确实不是很轻松，我相信也有很多人像我一样，即使我们现在学习起来有点困难，可只要我们把代码拉下来运行一下，调整下参数，再观察图形是否像我们预料的那样，也是能够加速对原理的理解，所以我的建议就是一定要多动手实践。

对底层技术的学习能让我们在技术方面更加地自由、自信，也是我们快速掌握新技术的内功。我认为计算机技术的核心是：操作系统、数据结构和算法、编译原理、计算机网络、计算机图形学。这也是我一直学习的方向。

提建议

更多课程推荐

程序员的数学基础课

在实战中重新理解数学

黄申

LinkedIn 资深数据科学家



涨价倒计时 🕒

今日秒杀 **¥79**, 9月11日涨价至 **¥129**

© 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 加餐二 | SpriteJS：我是如何设计一个可视化图形渲染引擎的？

下一篇 加餐三 | 轻松一刻：我想和你聊聊前端的未来

精选留言 (2)

写留言



李冬杰

2020-09-04

感同身受，以前对图形渲染即好奇又害怕，没系统化的知识作支撑，工具用起来都是提心吊胆，从开始学习图形学知识后，从一点一点细节和原理的学习中，逐渐找到自信。向罗同学学习

💬 1

👍 2



西瓜。

2020-09-08

我没有多少机会接触可视化项目，但是又想要深入学习这个方向，想找相关的岗位，可能缺少经验。对于这种情况，有没有什么建议？自主学习，做一些拿得出手的案例，再去找相关岗位？

