#### CrossApp简介

CrossApp是一款完全免费、开源、跨平台的移动app开发引擎。CrossApp以C++作为开发语言，图形渲染基于OpenGL ES 2.0，采用MVC框架模式。使用CrossApp开发的应用程序支持导出到各大主流移动平台，真正实现“一次编码，多处运行”的跨平台开发技术。

CrossApp主要由[9miao.com](http://9miao.com/)自研，9秒社团官方制定了CrossApp的基本架构，确立了CrossApp的基本雏形，而其后的版本也将由来自各方的开发精英自愿组成的9秒社团常务贡献委员会共同进行更新和维护。

支持所有主流平台：

CrossApp支持各大主流开发平台，轻松实现将应用程序移植到各个平台。

开源引擎所有内容并提供技术支持：

CrossApp完全开源，并且完全免费，开发人员在实际开发过程中遇到任何有关CrossApp的问题，都会由CrossApp官方技术团队免费提供技术支持。

丰富的UI控件：

CrossApp封装大量功能丰富的UI控件，极大提升开发效率，并且不断整合社区开发人员提供的优秀控件资源。

配套编辑器：

CrossApp为开发者免费提供相关编辑器，并由9秒技术团队不断更新和维护。

#### 主要功能与特色

**主要功能**

1. **跨平台**

**CrossApp暂时主要支持目前最主流的移动平台IOS和Android，后续更新版本将完善更多平台的支持。**

**CrossApp的平台无关性，使得相对传统的iOS和Android原生应用开发，CrossApp更能节省开发的周期和成本，App开发完成后只需要经过较为简单的移植工作，就能轻松实现在不同平台上应用。**

1. **整合部分第三方库**

**最初版本的CrossApp整合一些第三方库，例如常见的数据解析库jsoncpp、tinyxml、http等，同时CrossApp也通过第三方库spidermonkey转换，使之可以直接使用js脚本编写程序。**

1. **基于OpenGL ES 2.0**

**CrossApp的图形渲染使用的是Open GL ES 2.0，渲染效率高，可以使移动设备的GPU发挥到最佳效果。**

**引擎特色**

1. **开源**

**开源是9miao.com的一贯坚持的理念，CrossApp也是开源的移动App开发引擎。由于CrossApp是基于最宽松的MIT开源协议，所以开发者可以完全免费、毫无顾虑的使用CrossApp制作任何商业项目，并且在开发过程中遇到的有关CrossApp的相关问题，9秒社团技术团队负责相关技术难题的解答并给予技术支持（官方网站CrossApp专区回复问题，回复率100%）。**

**CrossApp会进行持续的版本更新，后续的版本更新与维护由9秒社团常务贡献委员会（后面简称“9贡委”）与9秒社团官方共同完成，开发者可以申请加入9贡委，为CrossApp的进一步完善做出自己的贡献。提交的版本将由9秒社团官方审核，审核完成后将整合进下一版的CrossApp版本。**

**CrossApp工作Git ：**<http://git.oschina.net/9miao/CrossApp>

**CrossApp GitHub ：**<https://github.com/9miao/CrossApp>

1. **丰富的UI控件**

**CrossApp的设计宗旨在于为移动应用开发者提供快速、高效的开发解决方案。在此基础上，CrossApp封装了大量的UI控件，各类控件的功能十分丰富，开发者可以直接使用这些控件进行应用的开发。这些控件基本满足应用开发需求中的大部分功能，也免去了自己封装UI控件花费的大量时间，进一步提升开发的效率、节省开发的成本。同时CrossApp也会整合部分由第三方开发者提供的优秀控件资源，更大的丰富CrossApp的UI控件。**

1. **提供CrossApp配套编辑器**

**高效快速的开发模式少不了不同编辑器的支持，最初版本的CrossApp没有提供引擎相关编辑器，但在2014年7月份的更新中将推出编辑器，辅助开发者进行更快速便捷的移动app程序的开发。**

#### 与其他开发平台比较

下表为当前移动平台开发App所使用的主要技术比较：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 技术  说明 | Mobile Web | Hybrid App | Interpreted App | Cross Compiling | Native App |
| 代表产品 | HTML5/JQuery Mobile | PhoneGap | Titanium | CrossApp | IOS/Android |
| 跨平台性能 | 强 | 强 | 中 | 强 | 低 |
| 离线能力 | 无 | 有 | 有 | 有 | 有 |
| 入门要求 | 低 | 低 | 低 | 低 | 高 |
| 功能 | 弱 | 中 | 中 | 强 | 强 |

通过上表分析我们可以得出：

1. Mobile Web

以HTML5和JQuery为代表的web开发技术是以纯浏览器为基础的，所以没有离线能力可言，更无法充分发挥各平台的功能和特性。

1. Hybrid App

Hybrid App即混合模式应用程序，是指介于web和native直接的app，具备一定Native原生App优秀的用户体验和Web App跨平台的优势。但由于Hybrid仍旧以前端技术为基础，无法对内存和系统资源进行有效的管理。

1. Interpreted App

以前端技术为基础，同样不能有效对内存和系统资源进行管理。

1. Native App

原生开发以IOS和Android为代表，虽然原生开发能够实现最佳的用户体验和高优化，但开发的成本较高。而且因为原生开发几乎不支持跨平台特性，相对于跨平台技术来说，带来的实际效益更低。

综上所述，还是Native App和Cross compiling的综合效率更高，开发者需要根据实际情况，考虑各方面因素来选择合适的开发解决方案。

#### 开发模式介绍

CrossApp采用的开发模式是各平台共享一套代码，采用交叉编译的方式实现不同平台上的运行。CrossApp是基于MVC模式的，面向对象的目标就是设计出低耦合、高聚合的程序，MVC模式提供了一种很好的解决方案。这里的MVC即模型层(Model)、视图层(View)、控制层(Controller)，三者各司其职，共同维持一个完整的应用程序。

在CrossApp中的Model层也就是我们各种数据原型、业务逻辑和算法，App在功能上完全依赖Model层的实现；View层顾名思义就是展现给用户的UI界面，而Controller层的职责就是把Model和View两个相互关联不大的层衔接起来，将用户的操作传递给Molel层，在控制View层表现出来。CrossApp通过CAView和丰富的UI控件来实现View层的功能，通过CAViewController及其子类来共同完成Controller层，实现Model和View之间的通信。