软件设计文档

1. **技术选型理由**

1.1开发环境及工具

Win10，VS, coco2d-x，c++

技术：面向对象

1.2 理由：

随着游戏动漫行业的迅速发展，游戏开发的引擎也随着市场变化而更新，相比其他游戏引擎，cocos2d-x有着独特的优势。

首先，cocos2d-x提供了全套的引擎和开发工具，涵盖从前期的游戏设计、资源设置、开发调试，打包上线全套的解决方案。Cocos2d重点优化了工作流，规范了整个游戏开发流程，降低沟通成本，提高开发效率。

Cocos2d-x也不断的优化游戏性能，保证高帧速率下可以获得更华丽的效果。

Cocos2d-x使用c++程序开发，跨平台运行于ios和android中，确保一次制作，全平台支持。

除此之外，cocos2d-x开放了强大的拓展功能并推出了CoCos Store,提供插件、资源、工具、素材能丰富的资源，为用户开发提供了更大的灵活性。

1. **架构设计**

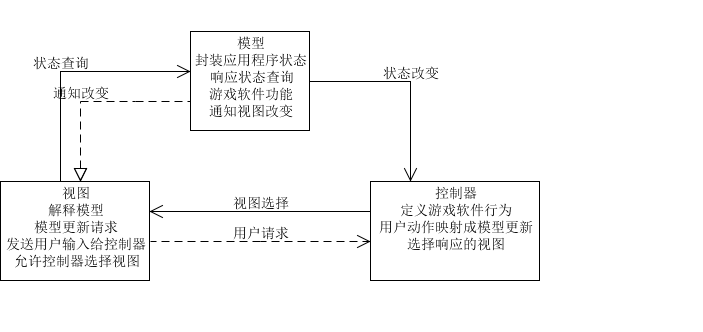
我们这个游戏软件基于MVC架构，主要分为模型，视图和控制器，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面和用户交互的同时，不需要重新编写业务逻辑，增强了应用的稳定性，易修改性和易复用性。

模型在游戏中代表的是事件驱动的有限状态机，模型会接受控制层发送的消息，并根据消息来更改自己内部的数据，然后把内部数据改变这一消息发送给View，通知它更新。

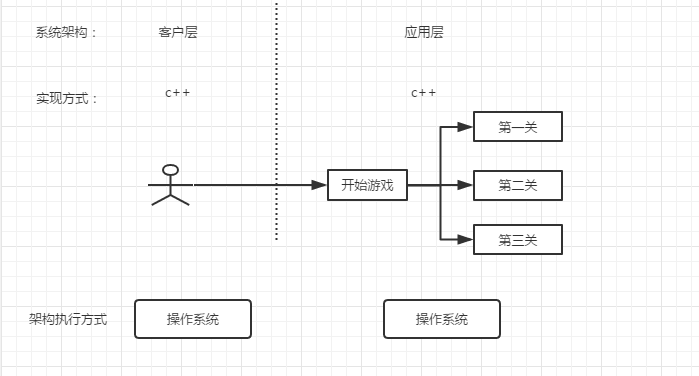
视图在游戏中代表的是模型消息的接受者，一般负责界面的显示。

控制器在游戏中负责将用户的触摸事件转化为逻辑事件，同时要对用户触摸事件中的信息进行正规化，并且通知变更。

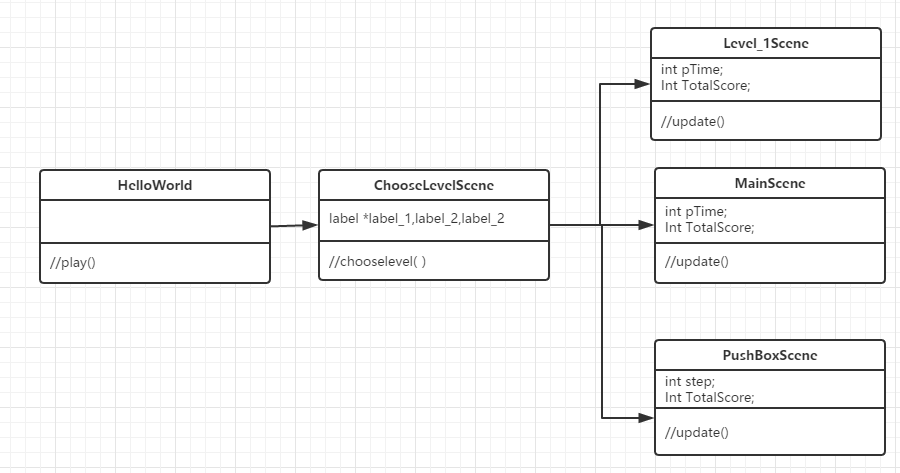
MVC架构图如下：



我们游戏共有三关，界面清晰明了，点击开始游戏自动进入第一关，如果成功过关自动进入第二关，成功过关进入第三关。如下图所示：



下面是游戏软件的一个简单类图：



1. **模块划分**

输入输出模块：COCOS2d-x本身提供的UI空间都有自身的输入响应机制，例如我们使用Touch对时间进行捕获和处理。

消息事件系统模块：考虑到游戏框架总体的拓展性，我们完全使用事件驱动模型来设计开发，将客户端事件的触发时机和具体处理逻辑彻底分隔开。游戏的各个模块，仅需要监听和实现其关心的消息事件，而无需关心事件何时被触发，降低了总体的耦合度。游戏中UI面板的隐藏/显示、事件响应、游戏流程的切换、游戏角色状态迁移等完全通过事件驱动方式开发。

UI系统：COCOS2d-x提供了一下UI控件，我们的UI并没有使用非常复杂的布局系统，所以我们大部分都使用的是COCOS2d-x提供的控件。

数据管理模块：戏的客户端面临大量数据，包括资源数据，道具数据，关卡数据，我们在此模块分出了数据层，主要对静态资源进行统一化的管理。