1. unity游戏引擎介绍

Unity3D是由UnityTechnologies开发的一个轻松创建三维视频游戏、建筑可视实时三维化、实时三维动画等交互内容的、多个平台的综合型高级游戏开发工具，是一个全面整合的专业游戏引擎，通过Unity简单的用户平台界面，玩家可以完成任何工作。Unity类似Director、Blender Game Engine、Virtools和Torque Game Build等以交互的图形开发环境为首要方式的3D游戏引擎软件。NVIDIAPhysX内置物理引擎带给玩家生动的交互。实时三维图形混合音频流、视频流，Unity的游戏和应用可以用C#语言，UnityScripts（一种JavaScripts的扩展）或者Boo其编译器运行在Window和Mac OS X下，可跨平台发布游戏。Unity简单的用户界面是层级式的综合开发环境，具备可视化详细的属性编辑器和动态的游戏预览特性。由于其强大的综合编辑特性，因此Unity也被用来快速制作游戏或者开发游戏原型。

unity游戏引擎的优点：

1. 多功能的IDE环境：Unity3D引擎的设计思路是All In One的思路，即将所有的编辑器集于一身，例如，可以在Unity里面利用图片九宫格的方式对图片的拉伸范围进行修改，可以编辑物体的动画等等。兼容三维画面的=美化所需要的美术资源文件格式，内嵌了动画、粒子、寻路等子系统可以对各种图形进行渲染。
2. 可自我定制的IDE环境：通过编辑代码（继承Unity的Editor），可以对unity3D编辑器进行编辑，这样方便了开发者将重复性的工作转换为编辑器上简单的数值更改，有利于策划测试出自己想要的效果。
3. 跨平台发布：只要安装下载好对应的sdk和jdk，Unity3D可以将完成的作品发布到现如今多个主流平台：手游（iphone、Android、Windows Phone）、PC（Windows、Mac、Linux）、游戏机（FC、N64、Wii、WiiU、Switch、GBA、GBASP、NDS、3DS、PS1、PS2、PS3、PS4、PSP、PSV、PS4Pro、XBOX、XBOX360、XBOXOne、XBOXOneX）、网页游戏（基于多种主流浏览器）等等。
4. 基于组件的对象系统：基于组件的对象系统，是指系统里的对象模型采用基于组件的方式，即把所有需要提供给游戏对象的基础功能都独立成单独的组件模块（Component），一个具体的游戏对象可以将它需要的功能模块组合到一起使用。所有“功能”不再是父类中的接口，而是变成子对象的实力，为游戏对象提供服务。这样既保证了功能代码的可重用性，有增加了整个对象体系的模块化和灵活度。
5. 基于Mono的开发脚本：众所周知，C/C++是图形界的宠儿，但是由于C++一不小心就会带来内存和效率的问题，所以Unity通过Mono的桥接，使得C++图形引擎和带GC的内存安全语言进行结合，不仅介绍了安全隐患，也使得大家在编写跨平台代码的时候更加简单。

本次项目开发采用Unity3D的原因：

目前世面上最火热的3D游戏开发引擎有两种，一是我现在在用的Unity3D，二是虚幻4。但是因为虚幻4已经建立了一个超高的图像表现标准（符合大家对3A游戏的定义），有着复杂的例子模拟效果和先进的动态照明技术，给了开发者一个全新的开发平台。正因为如此，虚幻4开发出来的游戏对于CPU和GPU的要求比较高，在手机端来说，目前的主流手机很少可以流畅的跑虚幻4开发出来的游戏。因此，目前虚幻4更多的是用在开发电脑端的游戏上。而Unity3D相较于虚幻4来说，图像效果比不上，只能算是2A的效果，但对设备的配置要求低，因此在手游端特别受欢迎。

综上所诉，由于本次项目是手机游戏，因此我采用了Unity3D游戏引擎进行开发。

开发环境 Visual Studio 2017

Visual Studio 2017是微软公司推出的开发环境，是目前最流行的Window平台应用程序环境。Microsoft Visual Studio 2017支持C#、C++、VB多种语言，可以快速实现相应的功能。所写的目标代码适用于微软支持的所有平台，包括Microsoft Windows、.Net Framework、.Net Compact Framework和Microsoft Silverlight 及Windows Phone。特点是易于使用、高效、灵活。在本次项目中利用Visal Studio 2017开发Unity游戏相关的C#脚本。

1. 网络协议：Protobuf协议介绍：Protobuf是一种平台无关、语言无关、可扩展且轻便高效的序列化数据结构的协议，可以用于网络通信和数据存储。

Protobuf协议的优点：

1. 体积小：序列化后，数据大小可缩小约3倍
2. 序列化速度快：比XML快20-100倍
3. 传输速度快：因为体积小，传输起来带宽和速度会有优化
4. 插件：EasyTouch——虚拟摇杆控制

介绍：EasyTouch是android系统的辅助操控工具(类似iPhone AssistiveTouch)，EasyTouch漂浮在手机屏幕或任何APP之上, 您可以将它移动到屏幕任意边缘，也可以随心隐藏在通知栏，让它不影响您的游戏或视频体验。

优点：1、开源。可以自己改动代码，定制适合项目的虚拟移动摇杆之类的操作。

2、设置简单，只要通过简单的设置，就可以实现虚拟摇杆控制人物移动。

使用EasyTouch的原因：用unity开发游戏的时候，有时候想控制物体移动，但是发现做虚拟摇杆太麻烦了，于是采用EasyTouch，方便快捷。

1. 插件——DoTween用于动画

介绍：DOTween是一个用于Unity的快速、高效、完全类型安全的面向对象动画引擎，为c#用户进行了优化。

优点：，1、开源，可以根据个人喜好去更改和设置DoTween的使用的方式。

2、通过简单的设置，就能做出优秀的动画播放效果。

使用DoTween的原因：在本次项目中，DoTween用的最多的地方主要是ui的动画。一开始是使用协程来控制ui的动画播放效果，但是由于协程的控制麻烦，同时在优化方面没有DoTween做的好，所以最后还是采用了DoTween这个插件。

1. AssetBundle的介绍、原理和优点

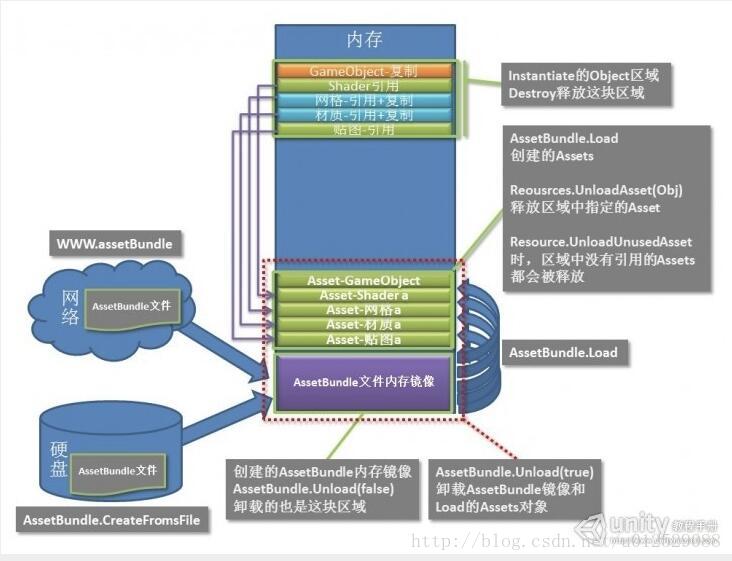
介绍：AssetBundle是一种使用LZMA压缩方式压缩的资源文件。AssetBundle主要包含两个部分：头部和数据段。

头部包含有关AssetBundle的信息，例如标识符，压缩类型和清单。清单是一个由对象名称键入的查找表。每个条目都提供一个字节索引，用于指示在AssetBundle的数据段中可以找到给定对象的位置。在大多数平台上，这个查找表被实现为一个平衡搜索树。具体来说，Windows和OSX派生的平台（包括iOS）采用红黑树。因此，构建清单所需的时间将随着AssetBundle内资产数量的增长而线性增加。数据段包含通过序列化AssetBundle中的资产生成的原始数据。

原理：AssetBundle打包的时候，你可以指定一个mainAsset，那么加载完之后就可以通过AssetBundle.mainAsset来获取到了。你也可以不指定mainAsset，直接打包一堆内容进去，然后加载后通过AssetBundle.LoadAsset指定名字的读取出来。

在资源之间，存在着依赖的关系。可以把资源拆分得很细，比如一个模型，由网格模型、材质、贴图构成，你可以把每一个小部分都拆开，各自打包成压缩文件。当Unity需要加载使用的时候，把该模型的所有依赖的小资源都加载起来，然后根据依赖关系组装，就变回了我们看到的资源了。

配图理解：



优点：1、可以直接在编辑器UI上设置操作

2、提供了更简便的脚本API

3、Unity本身会处理所有的依赖关系

4、生成了一种叫做manifest的文件，用于记录资源之间的依赖关系，并以链式结构记录，修改时只需修改链的其中一环

5、增量打包功能。

1. Json配置表的介绍

介绍：JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

1. PlayerPrefab的介绍、存储地方以及优点

介绍; PlayerPrefs是Unity内置的一个静态类，可以用于存储一些简单的数据类型.

存储的地方：1.在Mac OS X上存储在~/Library/PlayerPrefs文件夹，名为unity.[company name].[product name].plist,这里company和product名是在project Setting中设置的

2.在windows下，playerPrefs被存储在注册的HKCU\Software[company name][product name]键下，这里company和product名是在project setting中设置的。

3.在Android上，数据存储（持久化）在设备上。数据保存在SharedPreferences中。

1. MVC框架

介绍： MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，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及用户交互的同时，不需要重新编写业务逻辑。

优点：

耦合性低：

视图层和业务层分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动MVC的模型层即可。因为模型与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。

模型是自包含的，并且与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。如果把数据库从MySQL移植到Oracle，或者改变基于RDBMS数据源到LDAP，只需改变模型即可。一旦正确的实现了模型，不管数据来自数据库或是LDAP服务器，视图将会正确的显示它们。由于运用MVC的应用程序的三个部件是相互独立，改变其中一个不会影响其它两个，所以依据这种设计思想能构造良好的松耦合的构件。

重用性高：

随着技术的不断进步，需要用越来越多的方式来访问应用程序。MVC模式允许使用各种不同样式的视图来访问同一个服务器端的代码，因为多个视图能共享一个模型，它包括任何WEB（HTTP）浏览器或者无线浏览器（wap），比如，用户可以通过电脑也可通过手机来订购某样产品，虽然订购的方式不一样，但处理订购产品的方式是一样的。由于模型返回的数据没有进行格式化，所以同样的构件能被不同的界面使用。例如，很多数据可能用HTML来表示，但是也有可能用WAP来表示，而这些表示所需要的命令是改变视图层的实现方式，而控制层和模型层无需做任何改变。由于已经将数据和业务规则从表示层分开，所以可以最大化的重用代码了。模型也有状态管理和数据持久性处理的功能，例如，基于会话的购物车和电子商务过程也能被Flash网站或者无线联网的应用程序所重用。

生命周期成本低：

MVC使开发和维护用户接口的技术含量降低。

部署快：

使用MVC模式使开发时间得到相当大的缩减。

可维护性高：

分离视图层和业务逻辑层也使得WEB应用更易于维护和修改。

有利软件工程化管理：

由于不同的层各司其职，每一层不同的应用具有某些相同的特征，有利于通过工程化、工具化管理程序代码。控制器也提供了一个好处，就是可以使用控制器来联接不同的模型和视图去完成用户的需求，这样控制器可以为构造应用程序提供强有力的手段。给定一些可重用的模型和视图，控制器可以根据用户的需求选择模型进行处理，然后选择视图将处理结果显示给用户。