# **从零点五开始做半个不能玩的小游戏（二）**

作者：[枸杞忧天](http://gad.qq.com/user/index?id=1075523)

链接：<http://gad.qq.com/article/detail/47073>

**好久不见。**

这回拖的有点久，一方面是因为很忙（懒），另一方面是写到一半的时候推翻了一次，所以没控制好时间...



闲话少叙，书接上文吧。

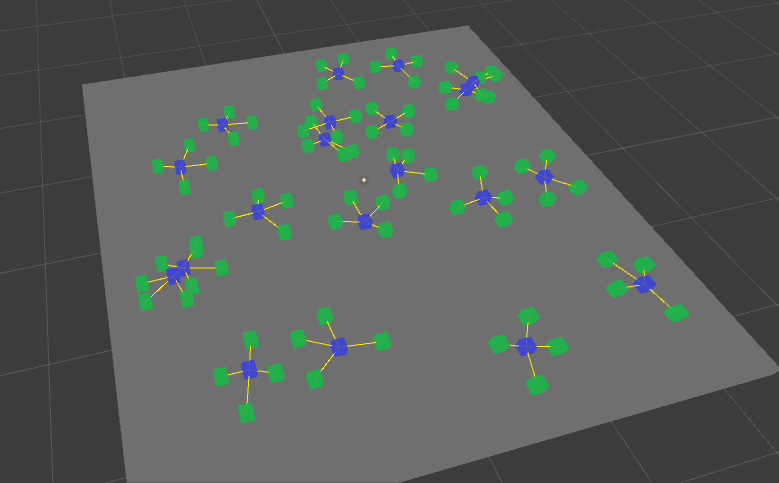
在上一回中，我们完成了：

1、创建项目。

2、添加世界管理器。

3、通过世界管理器生成及管理城市和野外。

好像大概是这个样子：



**本回目标**

1、加入英雄。

2、英雄可与周围的城市、野外进行互动。

3、建立简单的数值系统，驱动英雄决策。

上回中，为了快速看到效果，我们将城市和野外全部写到了**世界管理器**中，但这似乎不是一个很好的解决方法，这是因为：

随着项目的发展，每当需要为城市、野外添加更多功能时(如：城市建设、野外刷新等)，都无法从它们自身(城市、野外)出发去做处理，都需要通过世界管理器去配置数据，再通过逻辑将他们组合成城市、野外的样子。

长此以往，世界管理器将储存更多与其本身职责定位无关的数据(如：野外剩余资源量、英雄探索野外的进度数据等)。换位思考，如果你是世界管理器，你不烦么？你肯定会想：“介跟我有嘛关系？我只要管你们(城市、野外)就行了，你们自己的事儿难道自己不会管么？讨厌！死鬼！

我敢打赌，如果坚持这么做，过不了多久，世界管理器就会庞大、复杂到你再也不想打开它的程度，这也就是所谓的：**一胖毁所有**。



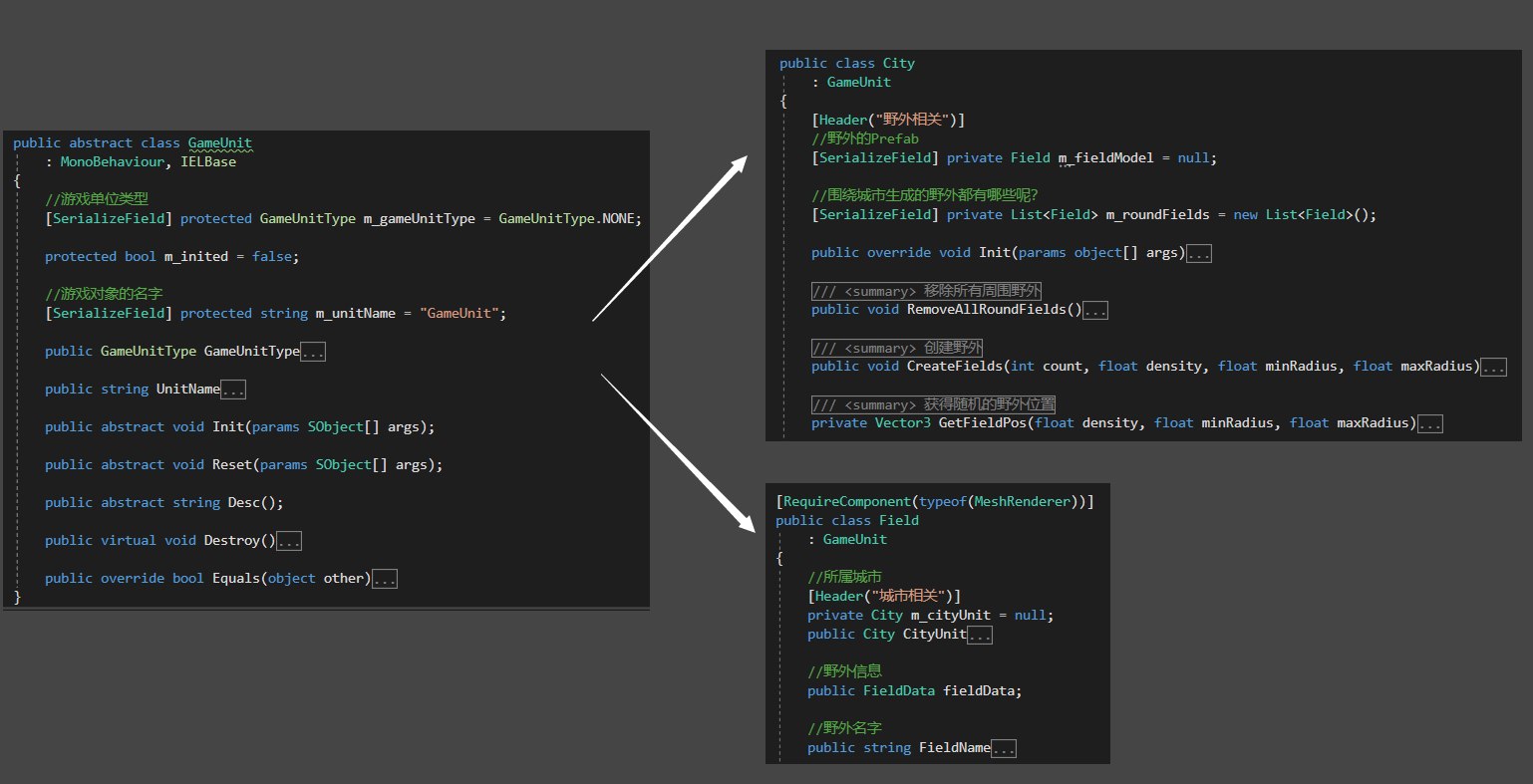
在此前那篇《Unity资源加载入门》中，我说过：

**无论你是否单身，在Unity的世界里，你都不愁找不到对象，因为一切都是对象。**

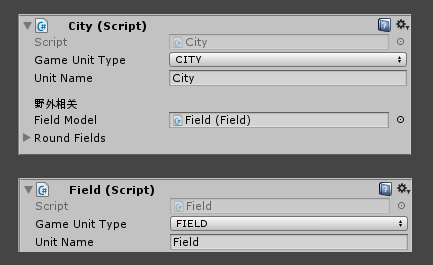
因此，在我们要加入英雄之前，先用对象的概念来稍稍调整一下项目，将城市、野外的逻辑从世界管理器中独立出来。

**重构城市、野外**

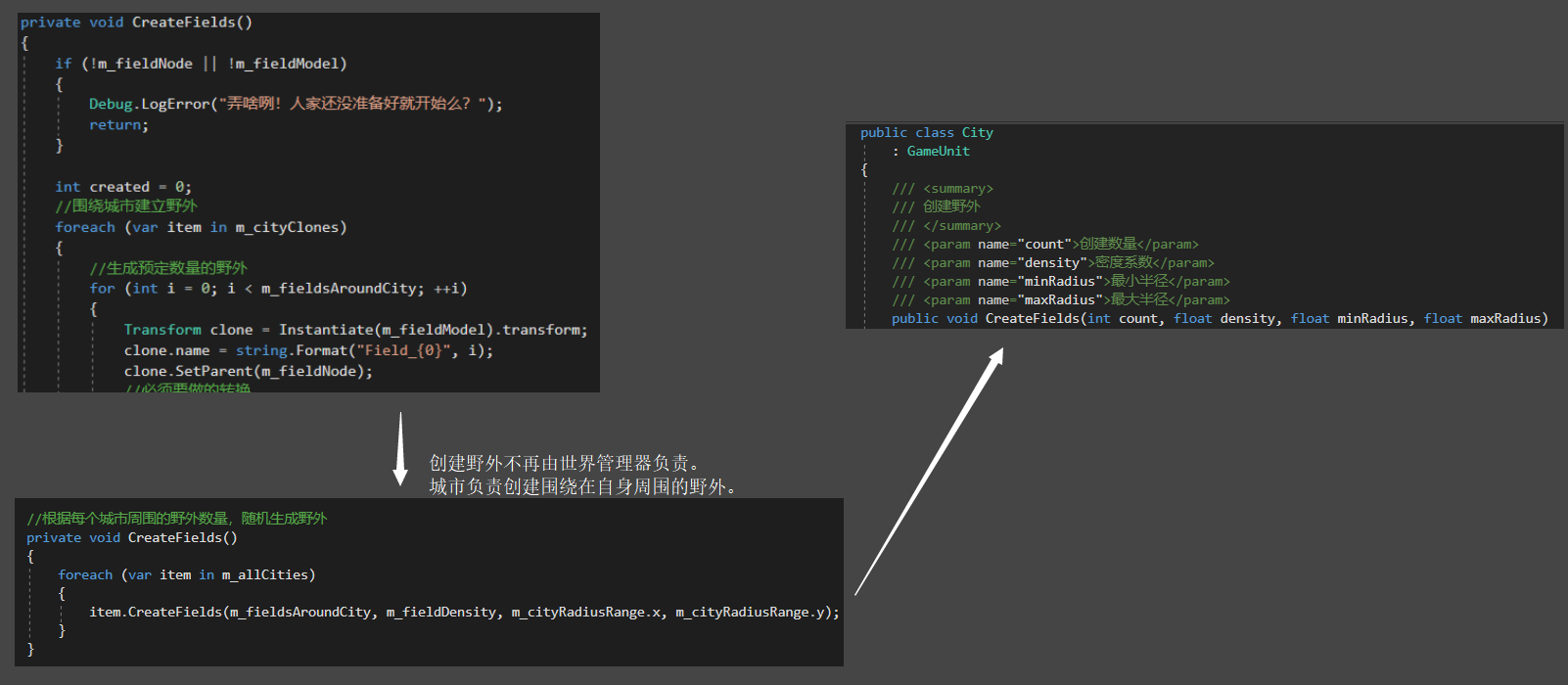
介于目前对于城市、野外的功能需求还是很简单，我们就先弄成这样吧：



建立脚本



修改City、Field的Prefab



简而言之就是将曾经属于城市的功能(创建周围的野外)放到了City中去做，而世界管理器仅生成城市并随机位置。

**加入英雄**

首先，让我们先脑补一下英雄应该是什么样子：

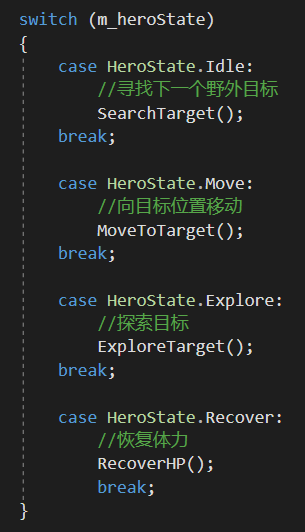
1、爱野外，不探索，毋宁死。

2、会自主选择野外前往并进行探索。

3、探索结束后根据情况考虑返回城市恢复战斗力。

4、恢复后开始一个新的探索。

让我们用一个简单的状态机来描述一下：



让我们先把工作简单分成两部分：

**1、快速实现状态机，跑通流程。**

**2、设置一个简单的数值体系：英雄数值，野外数值。**

**快速实现状态机**

先定下一些简单的规则：

1、生成的野外都有一定的资源量用于英雄探索。

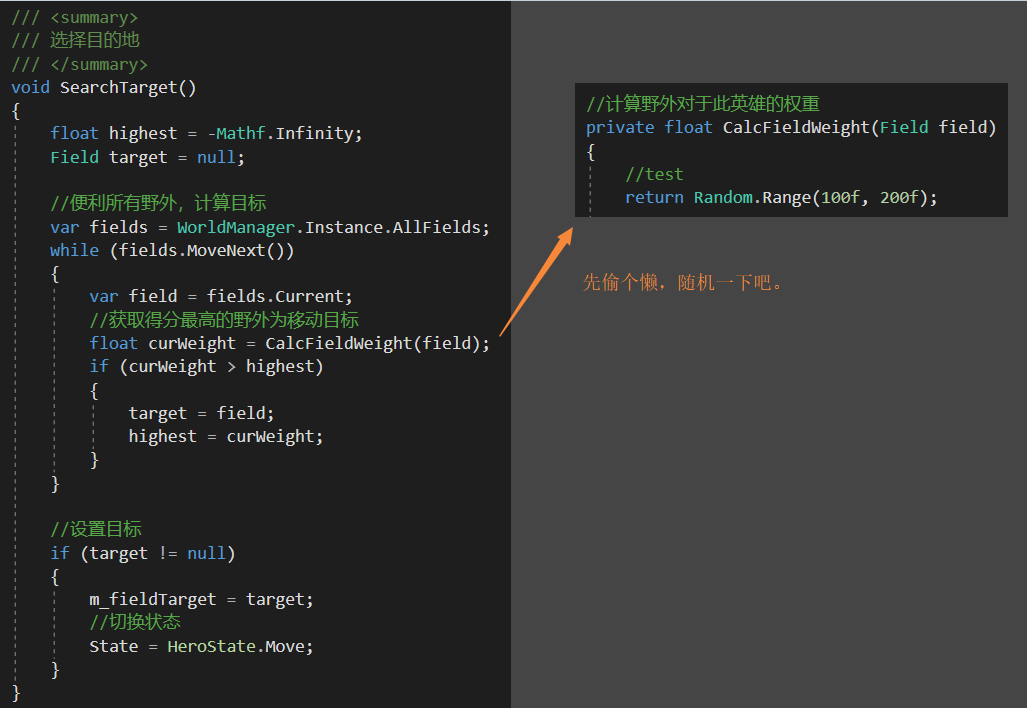
2、野外的资源量被探索到0后，将消失。

3、英雄探索中会损失生命值。

4、英雄生命值减少为0后会选择距离最近的城市进行恢复。

5、英雄生命值恢复后会继续探索野外。

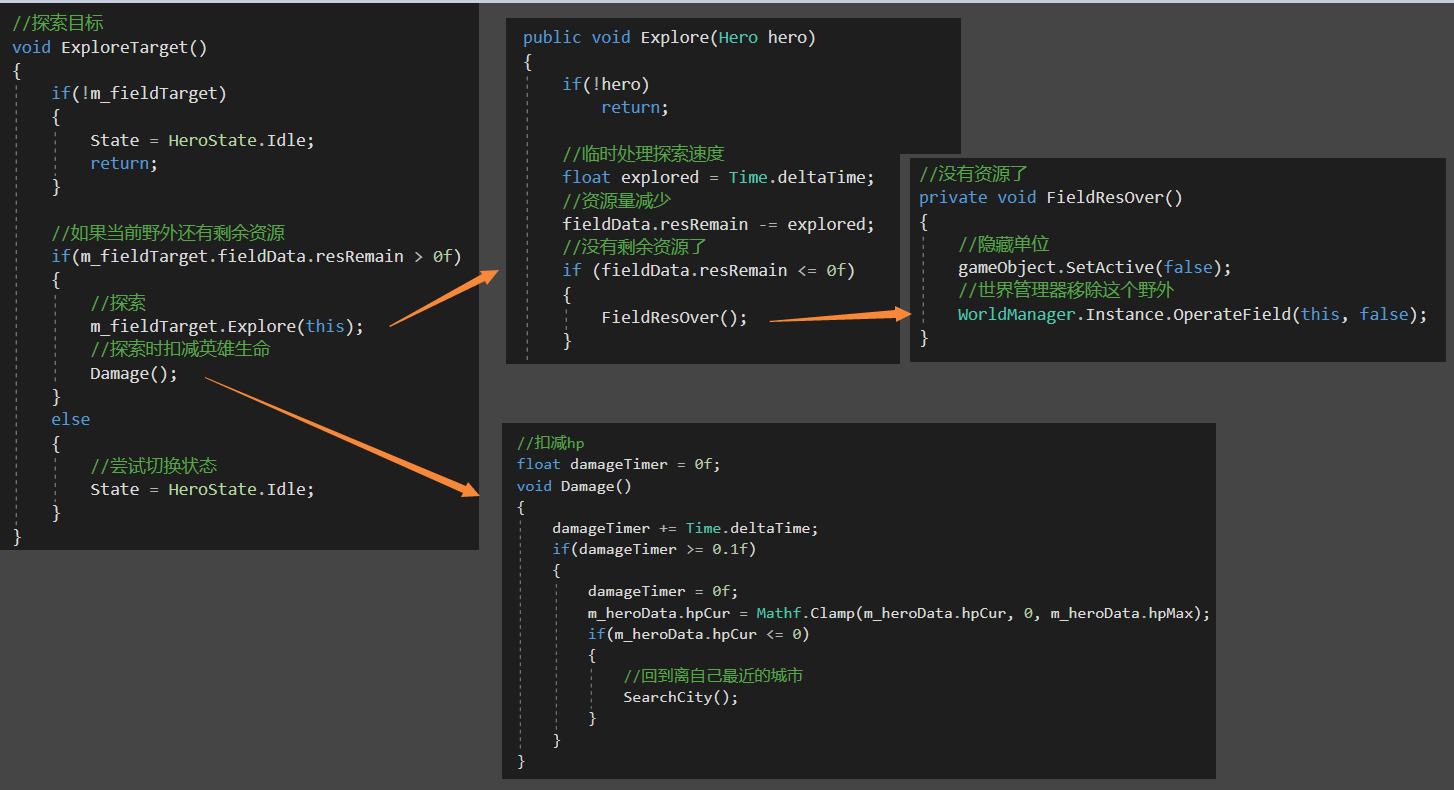
为了快速跑通流程，我们先将麻烦的部分简单处理，如：**英雄随机选择下一个探索的野外；每个野外的资源量设置为固定值等。**



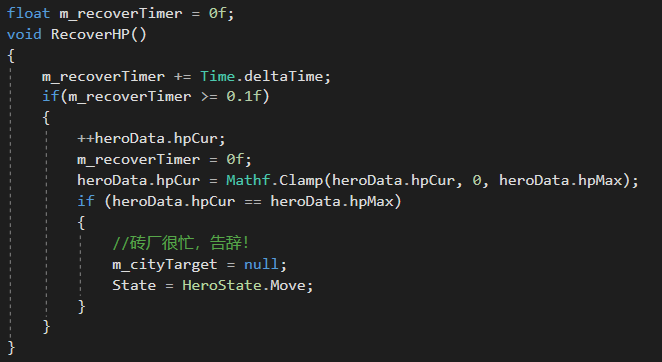
英雄确定目的地



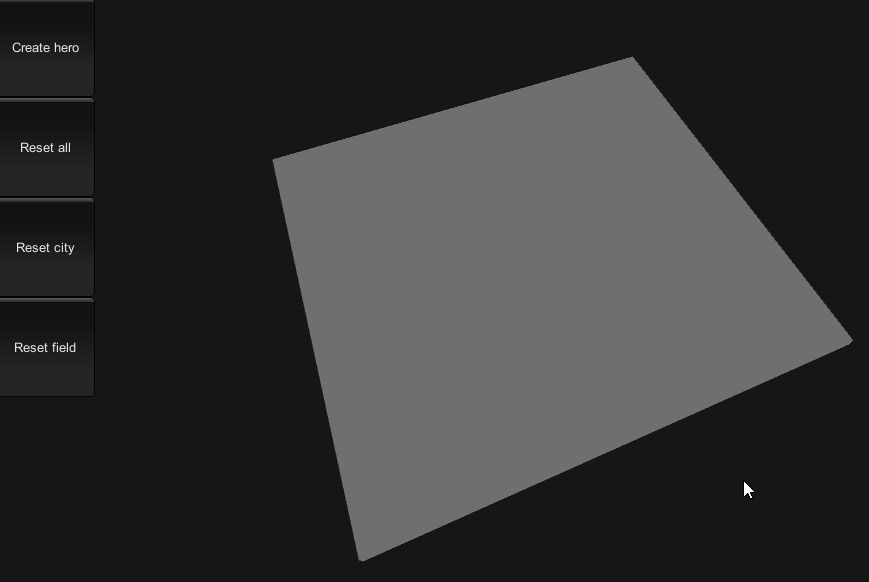
英雄向目的地移动



探索野外

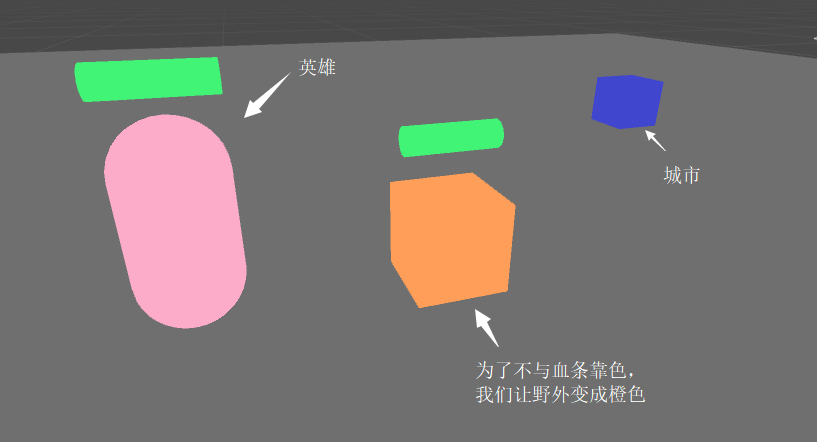


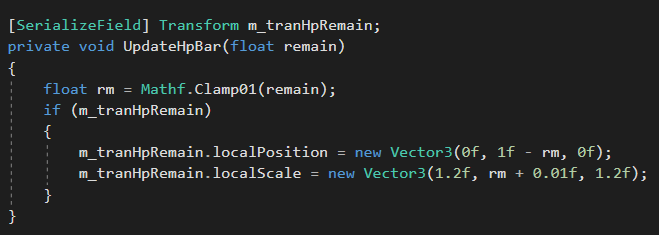
恢复生命值



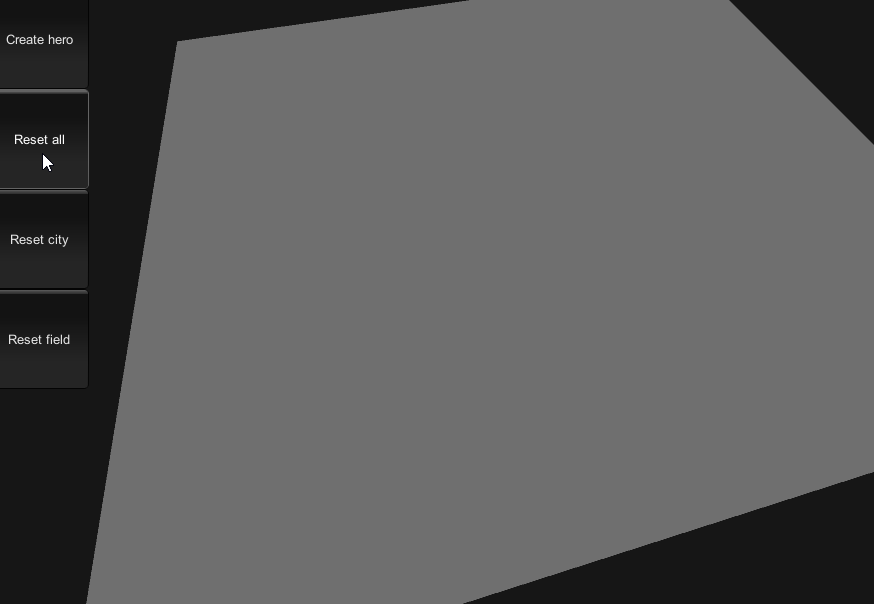
疲于奔命的英雄

为了能更好的看到野外剩余资源量和英雄的血量，我们分别为它们添加两个血条意思一下，并用脚本控制。





用个圆柱体冒充一下血条吧



疲于奔命的带血条的英雄

好了，我们已经有一个可以毫无原则，像没头苍蝇一样探索世界的英雄了。

虽然很多时候在现实中我们做事就是如此，但是游戏与现实毕竟是不同的，所以我们要为它们（英雄、野外等）添加一套简单的数值体系，来让英雄的探索更有逻辑性一些。

**设置一个简单的数值体系**

我们先把上面提到的规则稍微细化一下：

1、野外的数据包含**资源量**、**经验值**、**金钱**、**声望**。

2、英雄的数据包括**等级**、**经验**、**力量**、**力量成长系数、生命值、金钱、声望、金钱向**和**声望向**。

3、每个被创建的野外随机**[10. 30]**的资源量，每个被探索完的野外将在若干秒后**重置**成新的随机野外。

4、每个野外都有难度系数，难度系数的范围在**[1, 5]**，**系数升序难度递增**。

5、每个难度对应一个探索最低力量值，英雄等于此力量值时，探索花费的时间**等于**资源量，低于此力量会**延长**探索时间，高于则会**缩短**探索时间。

6、英雄探索野外会获得经验值、金钱、声望，英雄得到足够的经验值会升级，升级会根据力量成长系数来增加力量，英雄的最高等级为**30**级。

7、英雄每花费**1秒钟**探索野外，将消耗**1点**生命值，每个英雄都有**100点**生命值，**不会**随等级提升而提升，英雄生命值消耗到0时将前往最近的城市进行恢复 。

8、每个难度对应一个英雄等级，高于此等级的探索会得到**较少**的经验和声望，反之亦然。

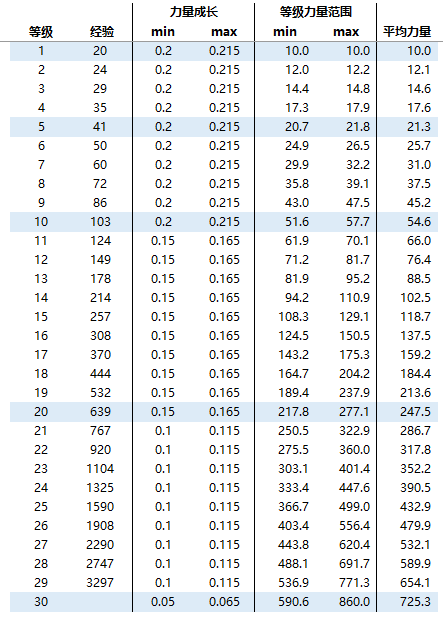
9、英雄可创建多个，且它们的初始等级、力量成长系数会有一些差别。

10、英雄会结合自身情况及野外情况，根据**收益最大化**原则确定下一个要探索的野外，而非简单的随机。

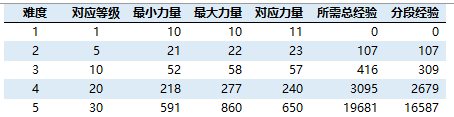
11、收益由野外的资源量、经验值、金钱、声望、探索时间、前往野外的路途时间共同决定。

12、英雄的金钱向、声望向影响英雄对野外的行动策略，金钱向较高的英雄会对产金币更高的野外抱有更大兴趣；声望同理。

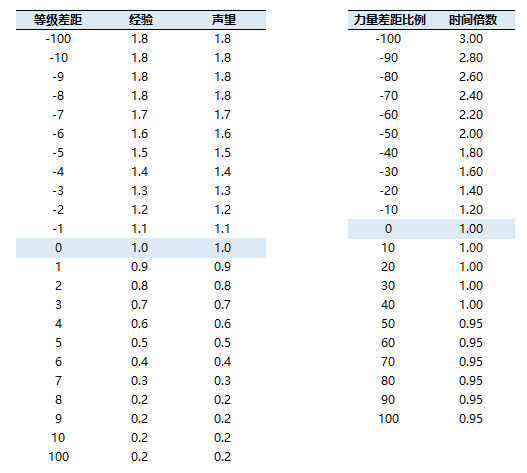
做这样一个简单的数值系统并不麻烦，虽然我没有数值策划的经验，可是我有Excel啊。



英雄升级所需经验及力量提升表

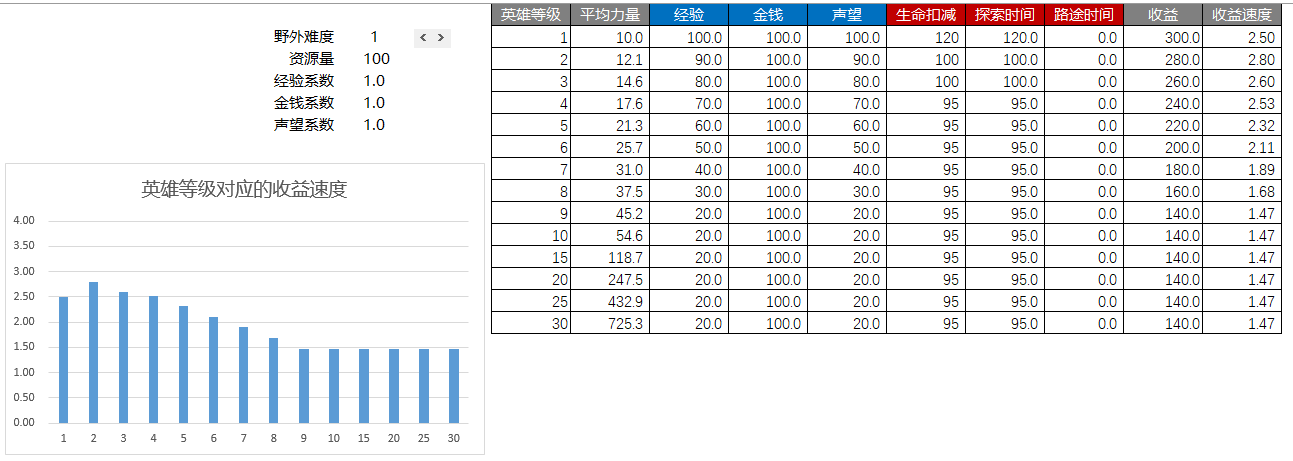


野外难度对应英雄等级及力量表

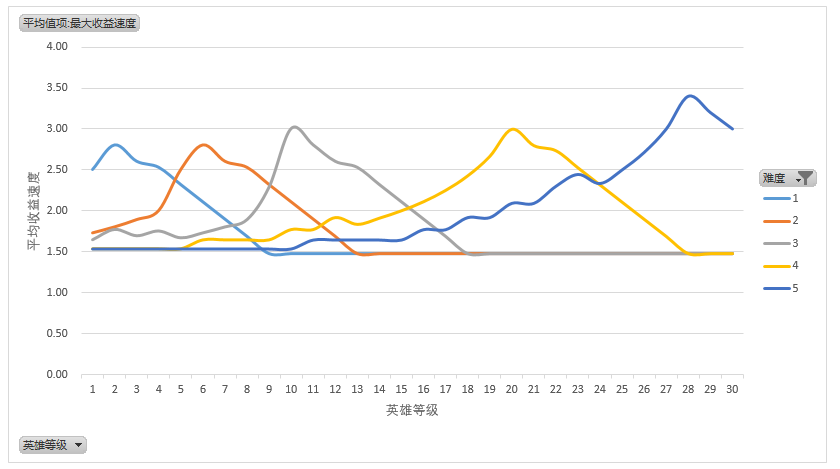


探索野外时等级、力量差距对应的经验、声望及探索时间倍数关系表

因为数据量并不大，数据的维度也不复杂，我们用Excel搭建一个小工具来模拟一下这套数值系统。



可以看到，**随着野外难度的提升，对应的收益最大等级也在增加。**



为了保证收益最大化，**英雄会选择适合自己等级的野外来探索。**

我们的目的是：

1、英雄随着等级的提升，逐步抛弃低收益的低难度野外，选择更高级的野外进行探索。

2、世界管理器则根据英雄平均等级的提升，调整新野外的平均等级。

我们希望看到的是：

1、多个英雄时，英雄们自主选择探索目标、自由探索并提升等级。

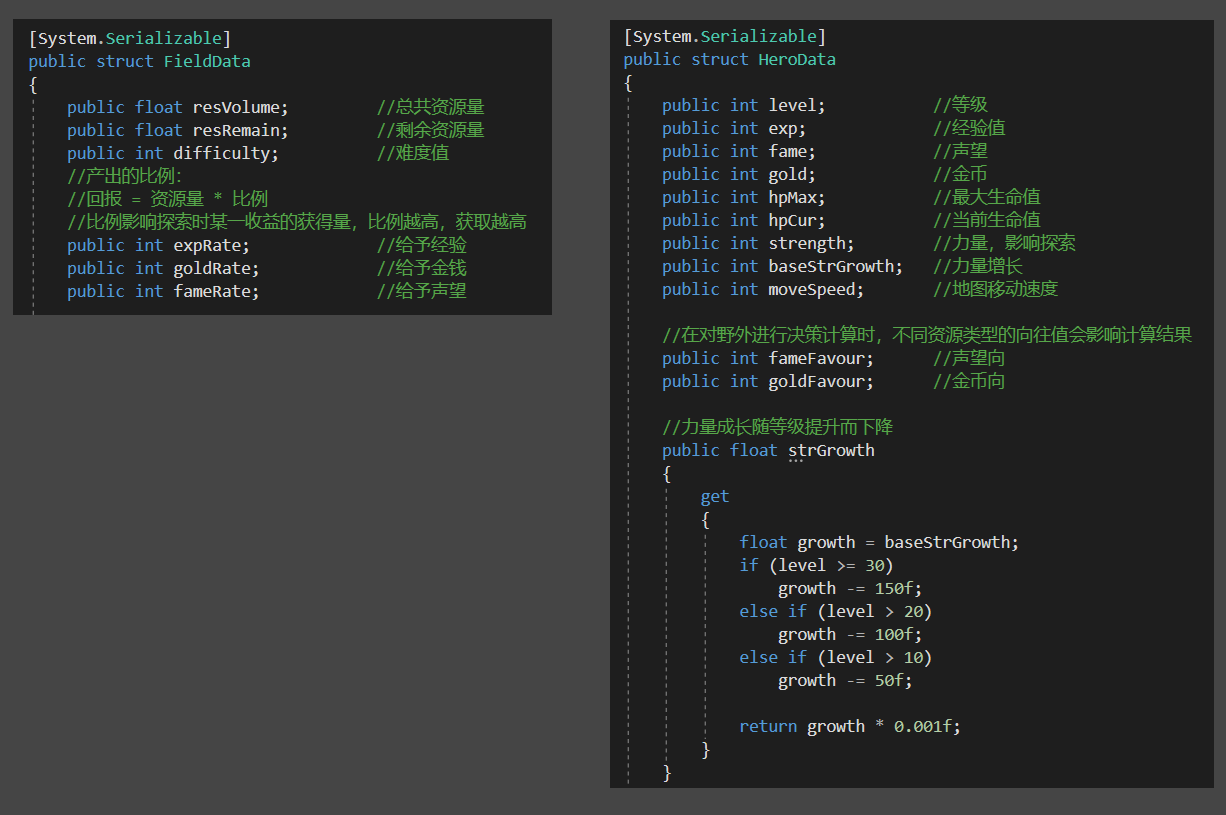
2、由于初始等级、力量成长系数不同，英雄间会拉开差距。

3、随着英雄平均等级的增长，世界管理器将维护出更多更高级的野外。

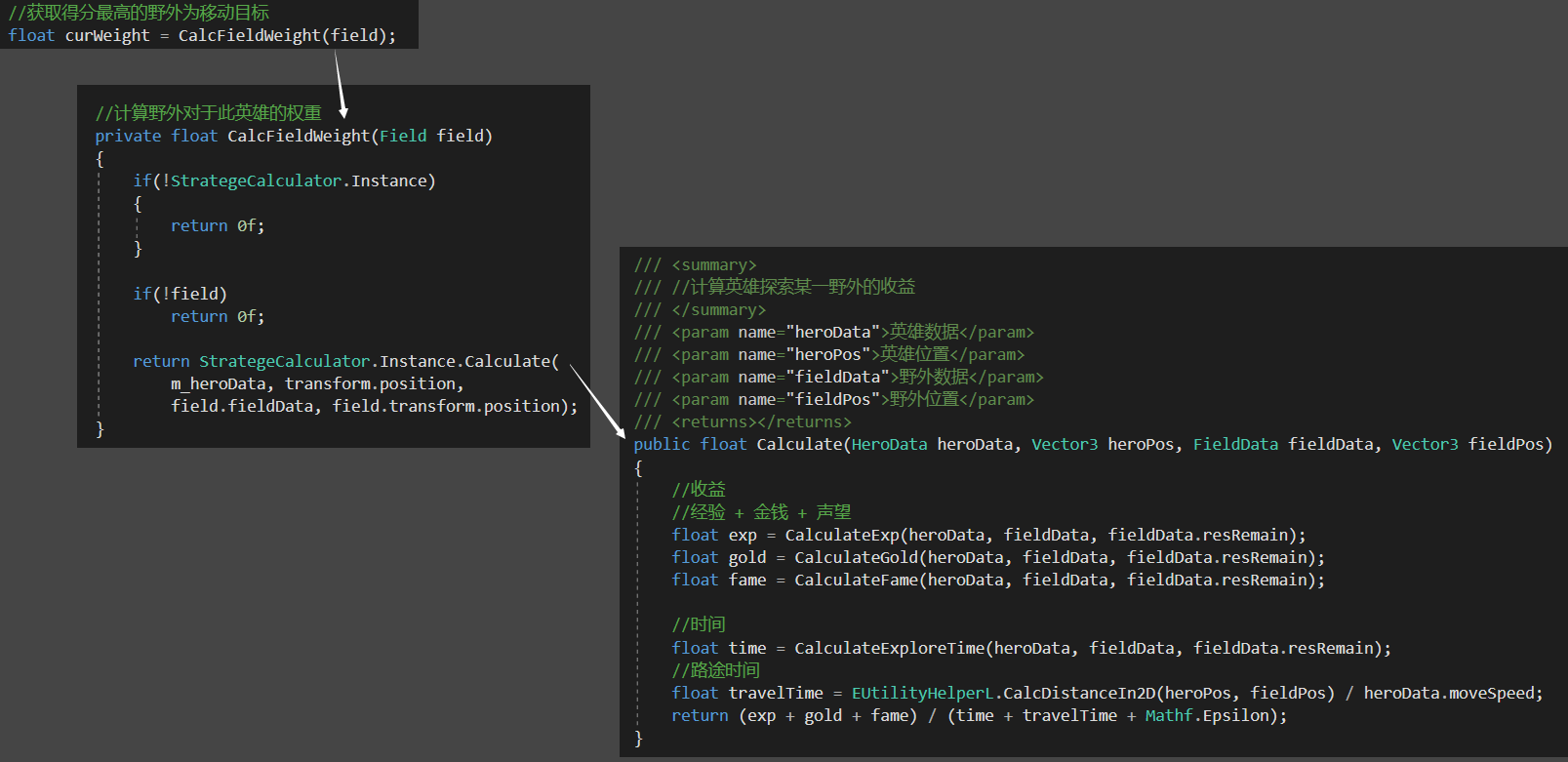
让我们奔着这个目标开始吧。

**实现数值体系对世界的影响**

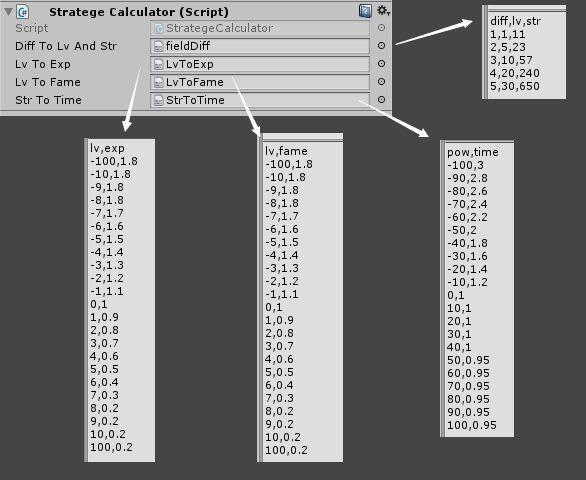
先确定一下英雄属性、野外属性的数据结构：



替换英雄原来的随机计算权重方式，变换为由一个**计算器**统一计算：



当然，具体计算的公式是参考上面Excel结果写的

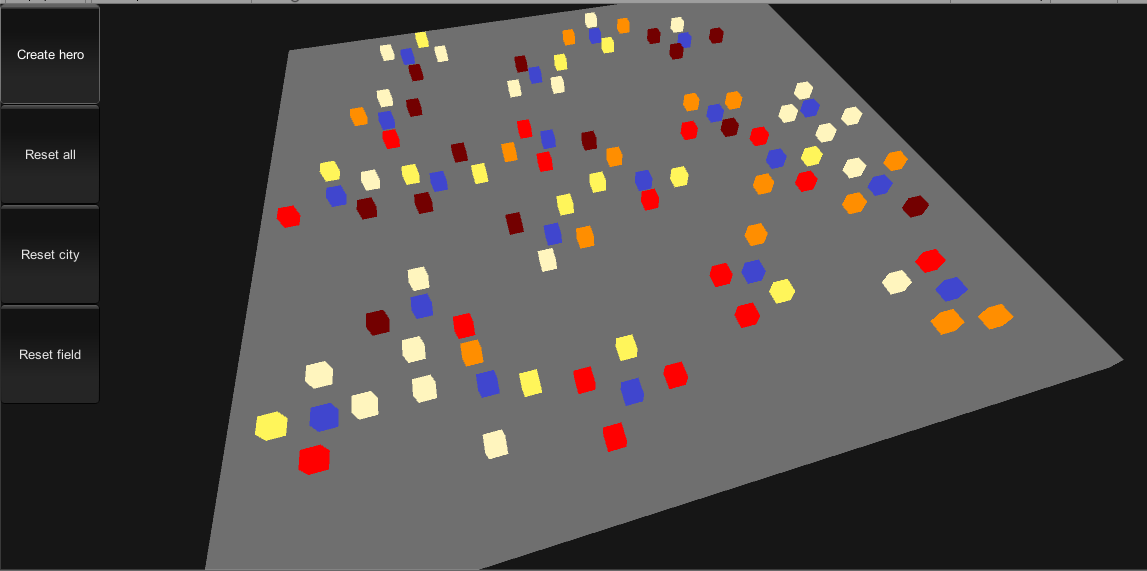


为了能快点看到效果，就偷懒的使用**.csv**这种方式读取吧

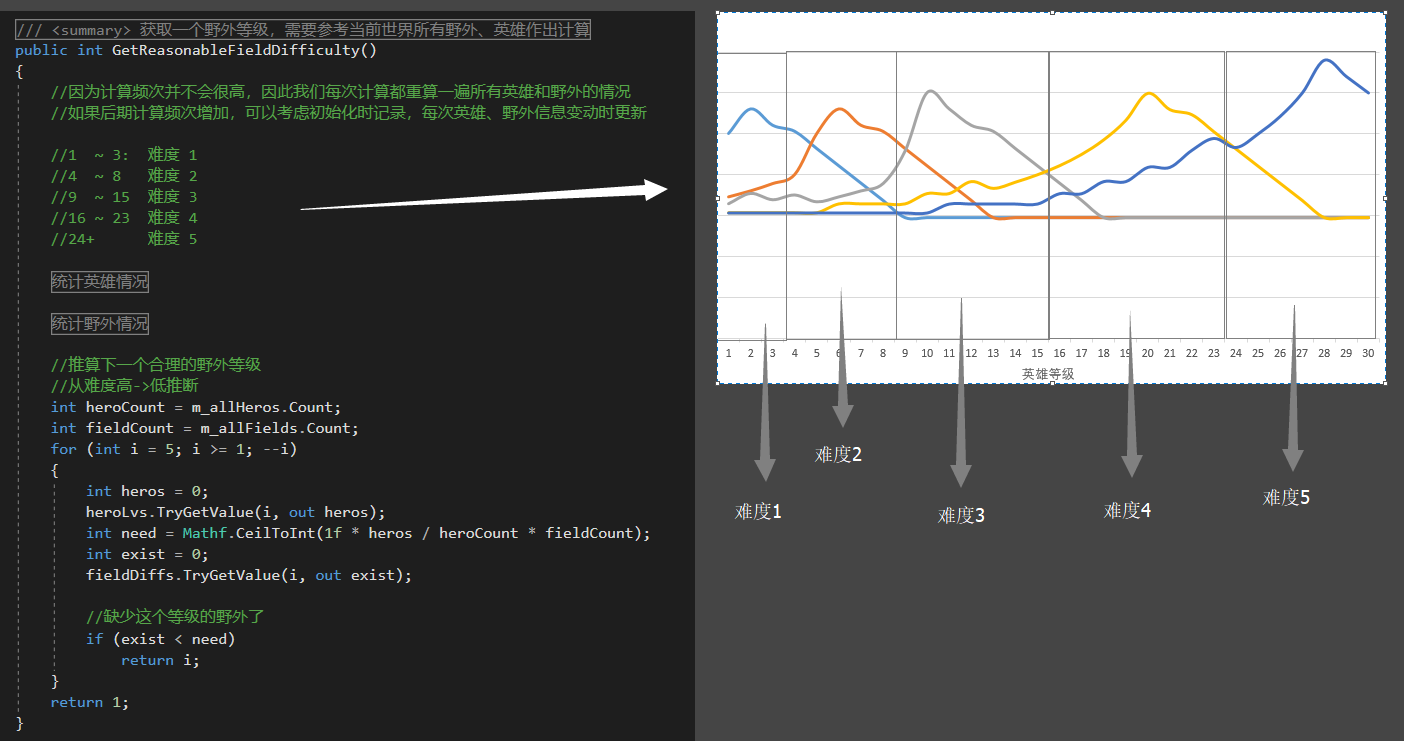
为了更直观的在地图上区别出不同难度的野外，我们为每个难度设置了**不同颜色**。



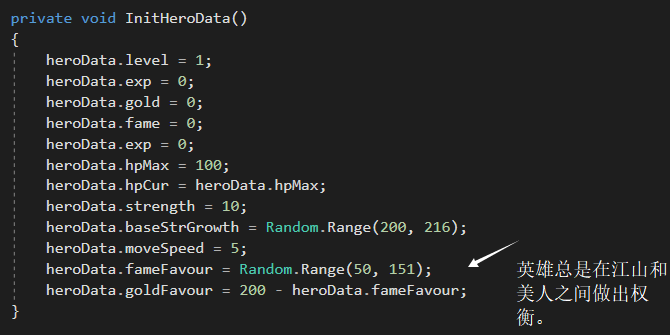
这时我们再设置随机难度的野外时，地图会变成这样：



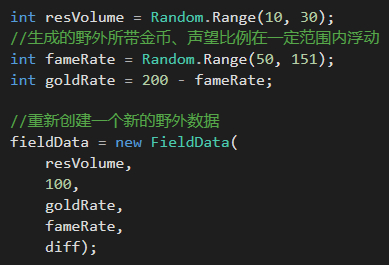
再调整一下世界管理器，因为一个野外被探索完毕时，在一定时间后会重置；而**重置出的野外，需要根据世界所有英雄、野外的情况来动态调整其难度**。而世界管理器作为唯一对所有对象情况都有了解的家伙，需要承担这个“主持大局”的责任。



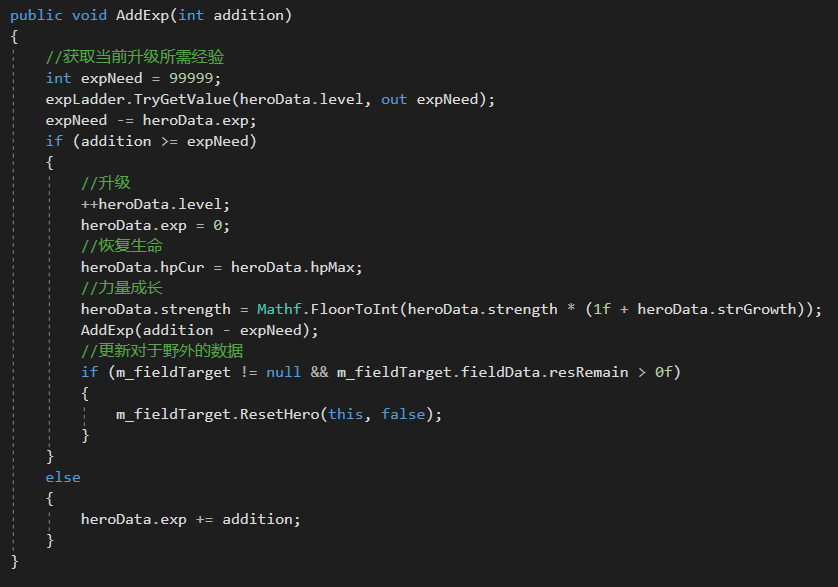
最后，我们为其他对象补充上应该具有的一些功能，就差不多了：



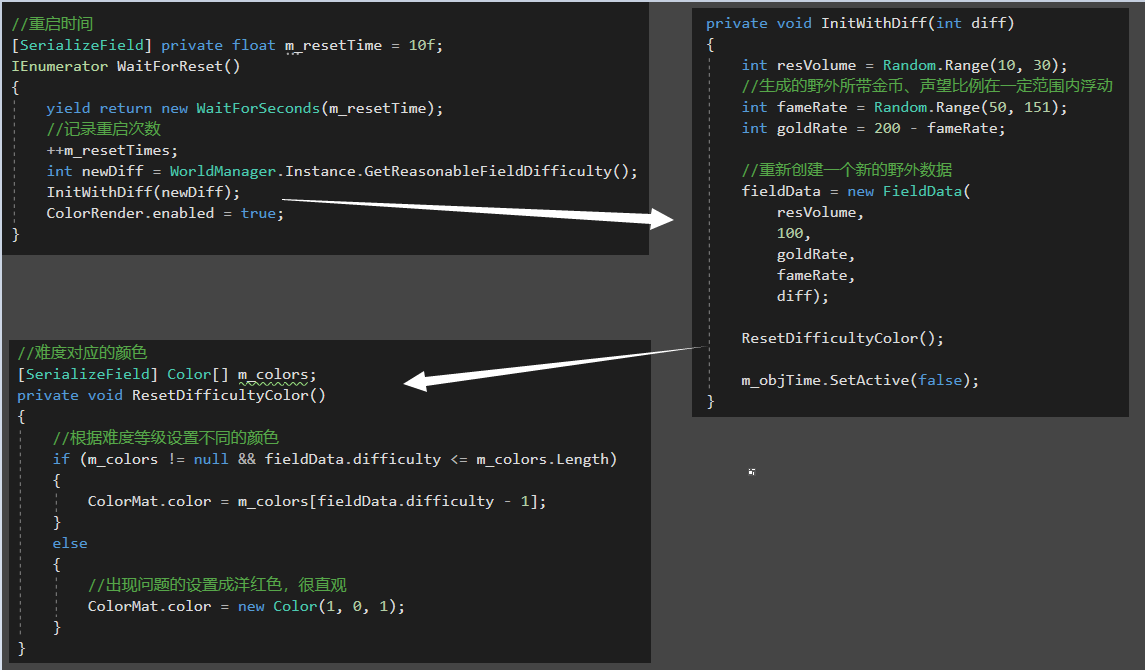
英雄的金钱、声望向需要随机设置



野外产生金币、声望的比例也要随机浮动

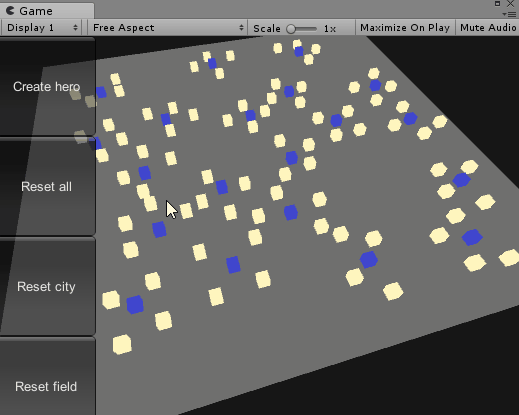


英雄获取经验并提升等级



野外自动重置并获取新的数据

看看效果吧：



加入很多英雄后，我们的世界变成了这个样子

**验证**

如果你能看到这里，应该也会发现一个问题：我们如何验证这个数值系统是否存在问题呢？

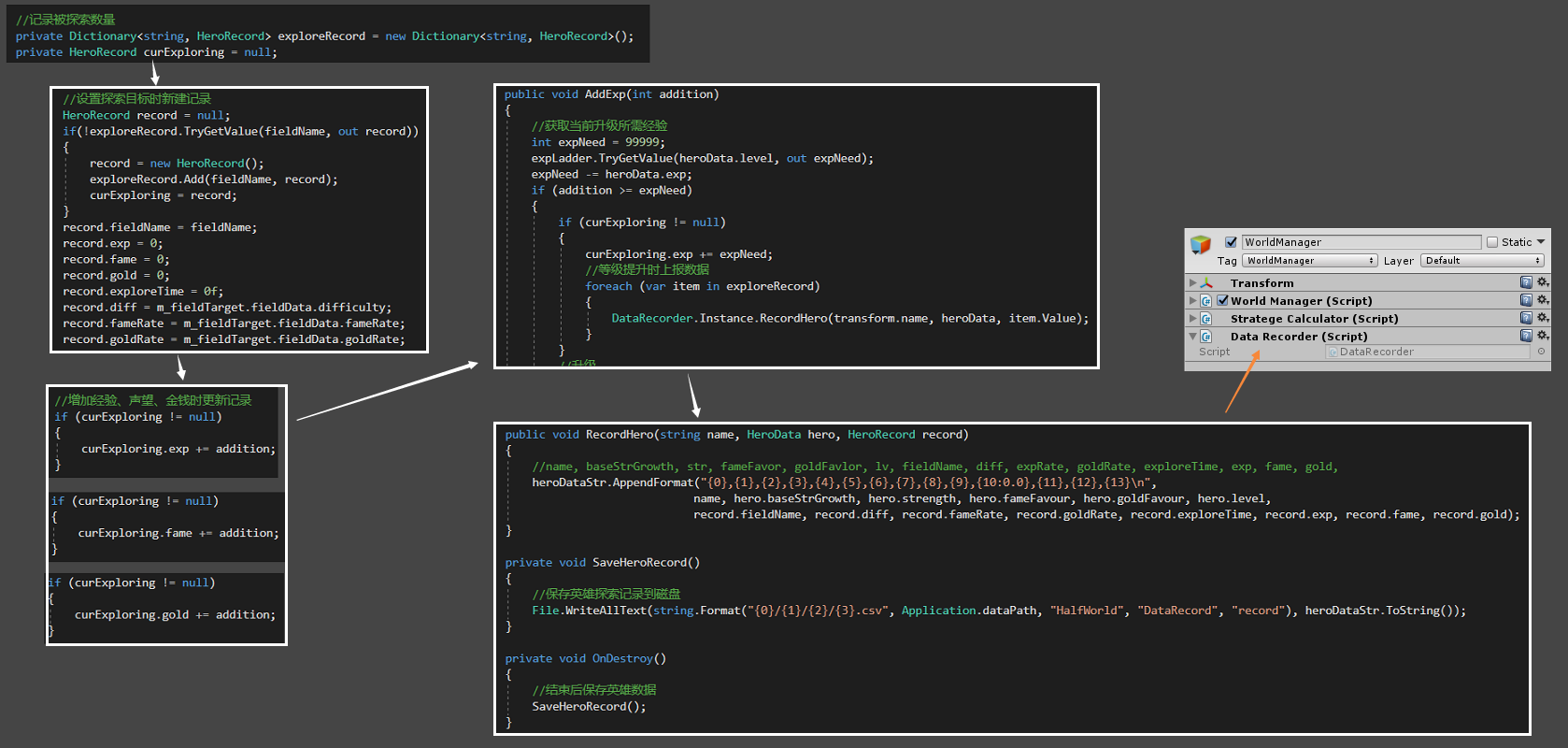
如:

1、英雄是否根据自己的喜好（金币，声望）去选择对应产量不同野外探索？

2、不同的力量成长是否会拉开英雄之间的距离？

巧了，我也不知道。

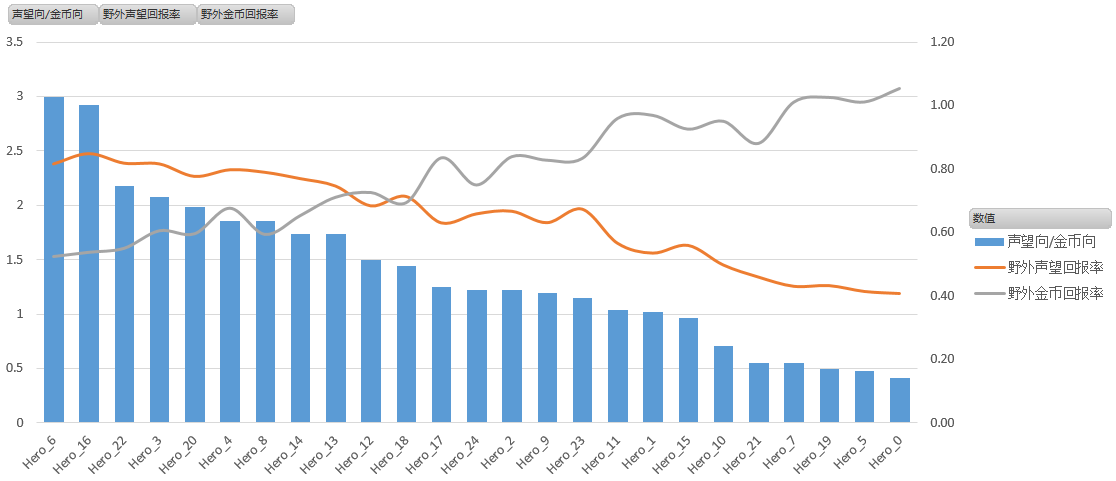
能很快想到的办法就是：让世界运转一段时间，然后收集英雄们的数据来看一下与预期是否相同。



简单的数据收集系统

用多倍速运行一段时间后，我们得到了一份英雄成长记录，用excel简单处理一下后，让我们使用公式来验证一下：

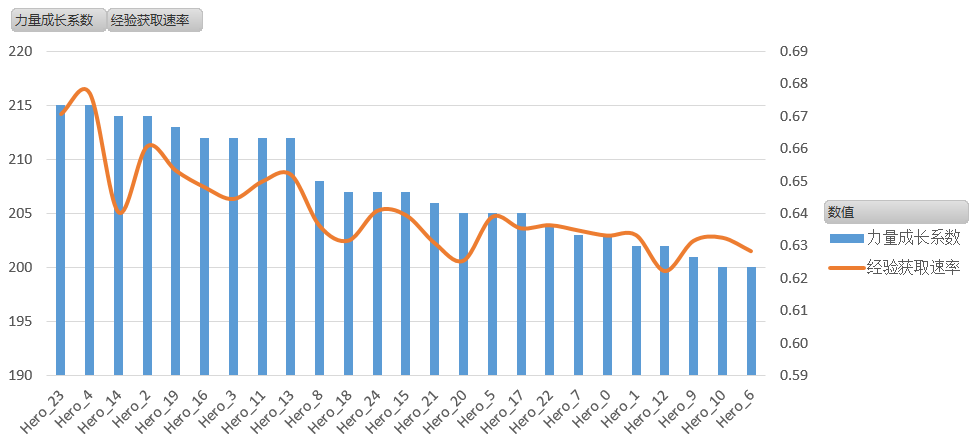




1、声望向/金币向高意味着英雄**更看重声望**的获取，因此它选择探索的野外通常**具有较高的声望回报率**。

2、相反的，金币向高的英雄**更看重金币**获取，因此它选择探索的野外通常具有较高的**金币回报率**。

3、当英雄的声望向与金币向基本相同时，所探索野外的金币回报率也会高于声望回报率，是因为**声望获取是随英雄等级提升而减少的，而金币则不会**。



总体而言，较高的力量成长系数提高了英雄的经验获取速率，因为较高的力量成长系数影响英雄的绝对力量值，而**力量值影响野外探索时间**。

嗯，貌似还差不太多呢，有bug也就先这样吧。

下回，将添加一个简单的UI界面，用以实时观察英雄和野外的情况。

附上此回的GitHub位置，有兴趣可以去取:<https://github.com/elsong823/HalfWorld>

**请关注微信公众号：偶尔学学Unity**



再见。