# **用Unity从零点五开始做半个不能玩的小游戏（一）**

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

链接：<http://gad.qq.com/article/detail/43816>

**写在最前**

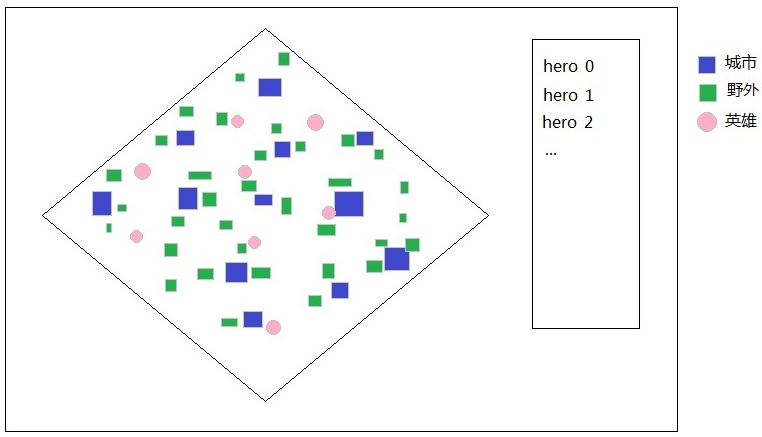
总想在工作之余写一些小玩具陶冶一下情操，因此从此篇开始，预计花费三、四次的时间，记录一个小游戏**从无到有、再到不能玩**的搭建过程。

**游戏概述**

运行这个小游戏将创建出一个温馨、和谐的世界，这个世界里着很多美丽的城市，每个城市周围又辅有若干可供英雄们探索的神秘野外；英雄们会根据自身属性选择探索目标，**完全不用你操心**！伴随着它们探索的深入，将谱写出一幕幕可歌可泣、催人泪下的传奇故事。

而你，作为游戏的绝对主角，将在这个世界中扮演一个十分重要的角色：**一个没有任何主观能动性的看客**。

由于写到这时还没开始构建项目，因此仅用我**精湛的画技**描绘一下游戏运行起来的样子：



哎？怎么感觉看完草图后，**连我自己都不想写下去了呢……**

但是没办法，**自己画的饼，跪着也要烙完。**

**预计内容安排**

第一回、描述游戏玩法、基本架构，生成随机地图，周围探索点。

第二回、构建英雄并可开始自动探索世界。

第三回、英雄与城市、探索点、以及其他英雄的互动 。

第四回、添加UI来增强游戏的表现力，并对全文做出总结。

**一些解释**

之所以叫从**零点五**开始，是因为我认为你至少应该具备以下能力后才开始看这篇文章：

1、会新建场景。

2、会创建对象。

3、会创建Prefab。

4、会写简单的C#脚本。

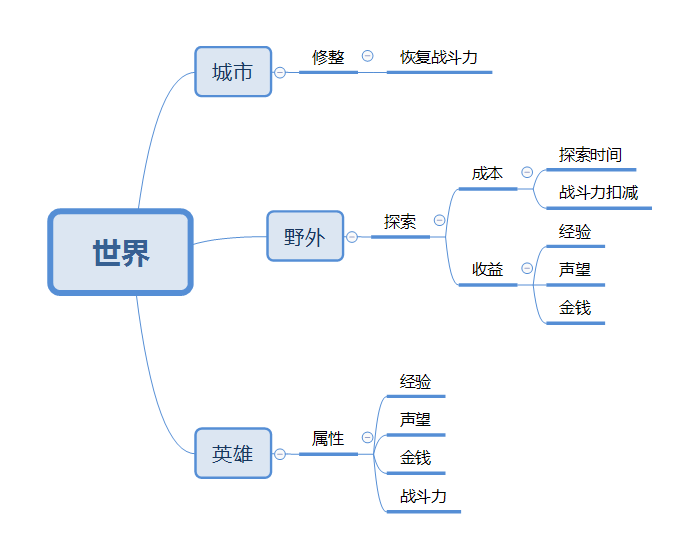
因为本文并不会对很基础的操作做详尽说明，比如：现在我们用鼠标轻轻点中这个对象，然后轻轻将它拖向文件夹，注意我拖了拖了拖了……唉！唉唉？！成功了？**成功了**！这个蓝色的就是预制体！**我们成功了……**



当然，除了你应该有点基础外，你最好**不要具备**很高的Unity水平，因为这样我就不能忽悠你了。

**游戏结构**

游戏的玩法已经在上文进行了描述，现在来看一下游戏结构及各游戏元素间的关系。

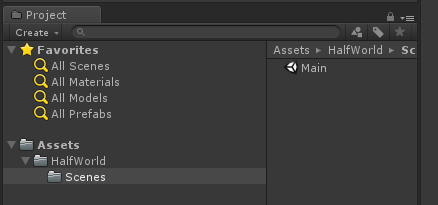


结构实在太简单了，就不做赘述了。

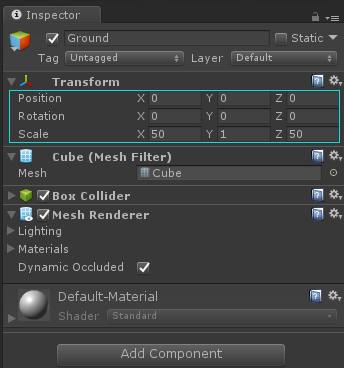
**正文**

**一、构建基本场景**

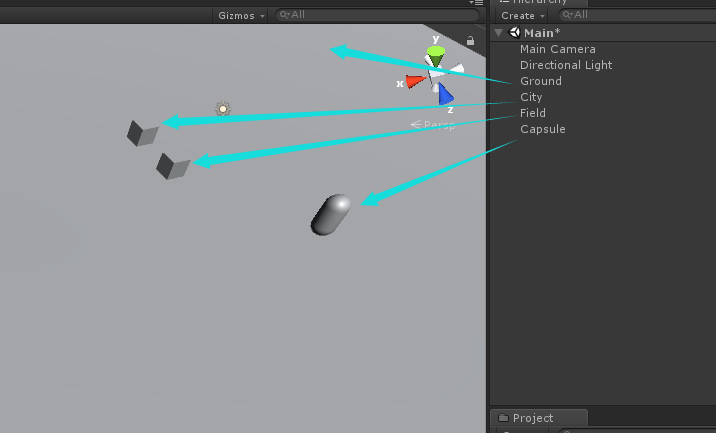
1、新建场景，并保存在场景文件夹中。



2、新建一个Cube作为整个世界的地面，尽量大点。

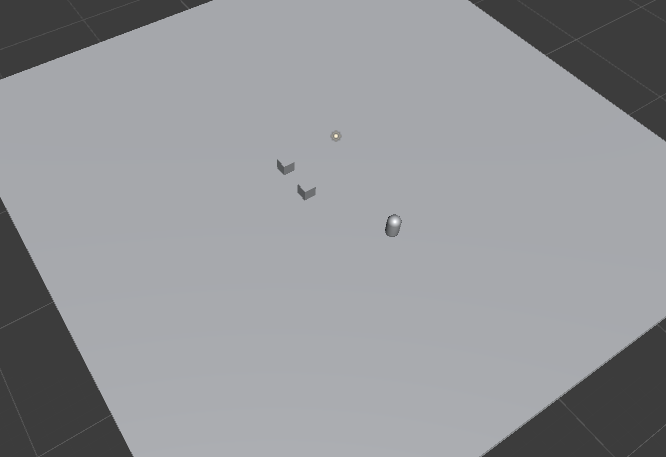


3、新建一个Cube作为City，一个Cube作为Field，还有一个Capsule做为英雄。



万事俱备。

微调视角到合适的角度，就会惊喜的发现一个美轮美奂的游戏世界已经呈现在了我们眼前。



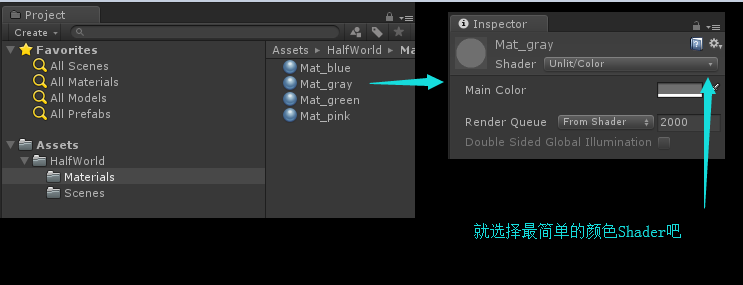
**真是完美。**



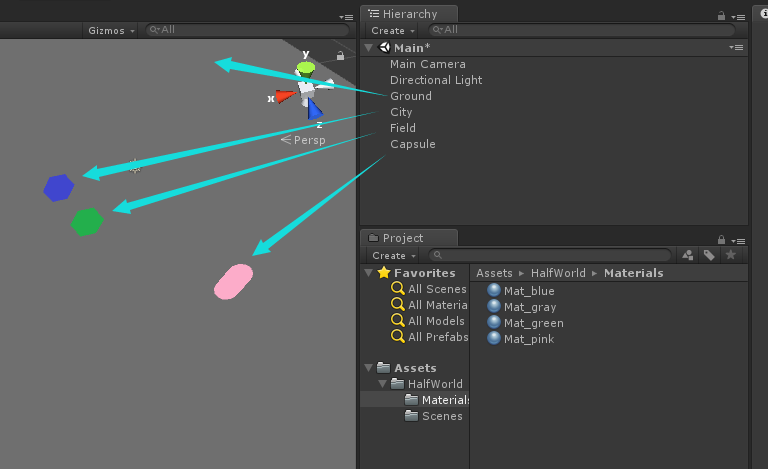
“小当家，这当真是3A级别的画质啊！”

但是为了能稍加**区别各种对象**，让游戏感觉尽可能的再好一些，我们决定加入不同的**颜色**用来区分对象。

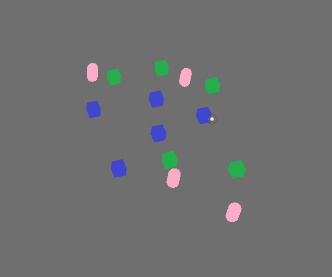
4、新建一些简单的颜色材质球



5、将材质球直接拖到各个对象上面，专业点讲就是：give them some colors to see see.



稍稍复制一些对象，预览一下整体效果：



**漂亮**，画质上升了不止一个档次，让人看了就想赶快充钱。

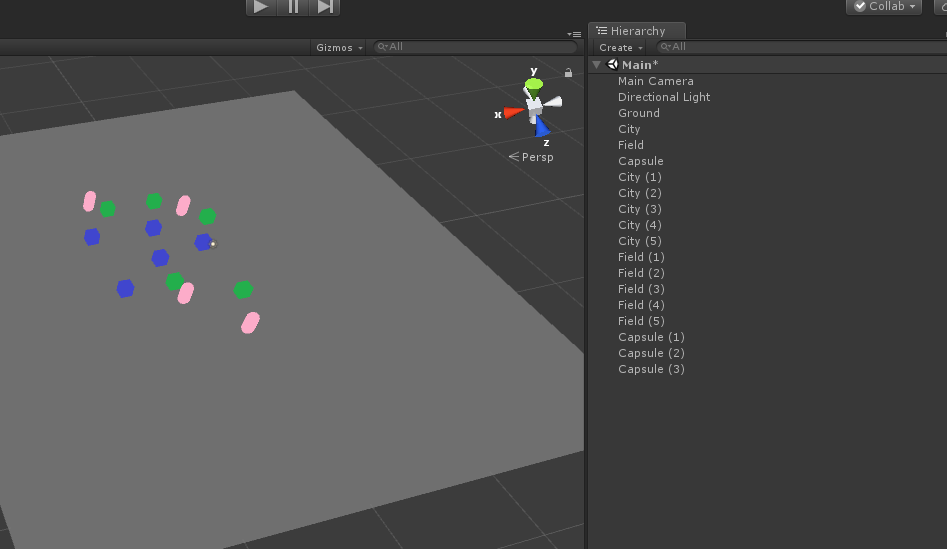


“我先充个钻石VIP为敬”

**可惜我们没有商店这个功能。**

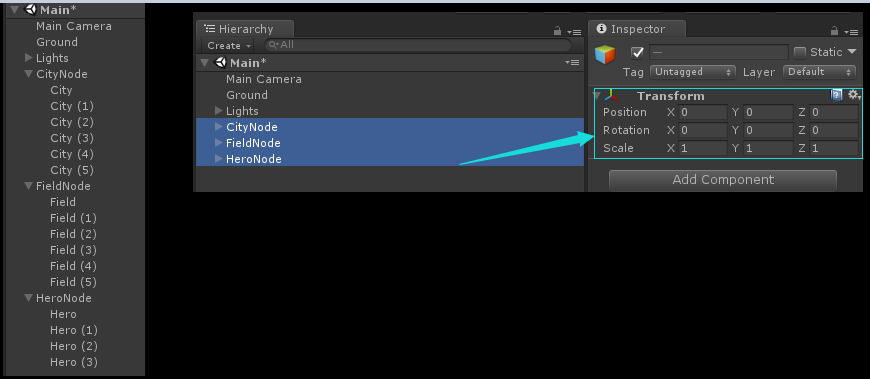
**二、整理场景**

浏览一下Hierarchy窗口下的场景结构，可以用四个字来形容：“**整齐而有规律，简单而不简约**”。



但是我们不能满足于此，于是简单调整一下**结构**，新增一些节点管理它们吧。

1、新增管理节点，并重置Transform的空间属性



新建好管理节点后，将之前复制的对象拖到对应的节点下，我们发现果然**清爽多了**，当然这么做最重要的原因是：**会让你看起来很专业**。

至此，我们的游戏开发已经接近了尾声。

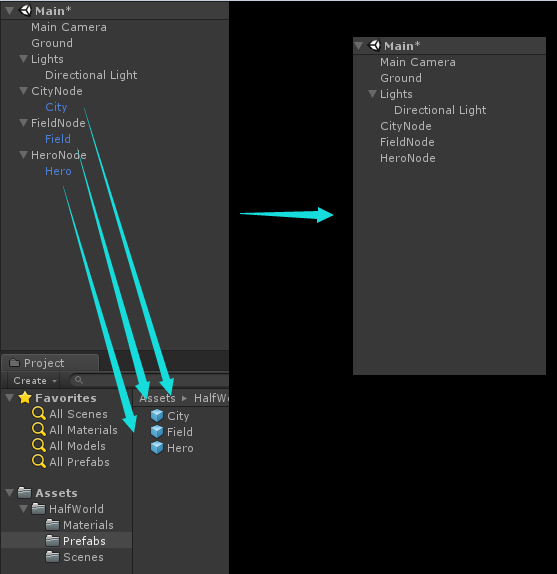
逗你呢，当然不可能，之前的对象都是直接复制出来的，这似乎**有一点点不专业**。我们更希望在游戏启动时或运行中，城市、野外和英雄可以动态随机的出现在这个世界里，因此我们需要做两件事：

① 分别为城市、野外、英雄准备好一份**Prefab**，用来**动态实例化**出更多的游戏对象。

② **添加**一个简单的**管理器**来实施实例化及管理实例化后的游戏对象。

因此：

2、移除无用的对象，创建城市、野外、英雄的Prefab



将多余对象删除，仅保存一个**最可爱的**作为prefab，恢复场景的干净整洁。

3、建立世界管理器

写代码最忌讳的是没有想明白就开始动手，所以在写这个管理器之前我们先梳理一下对它的需求：

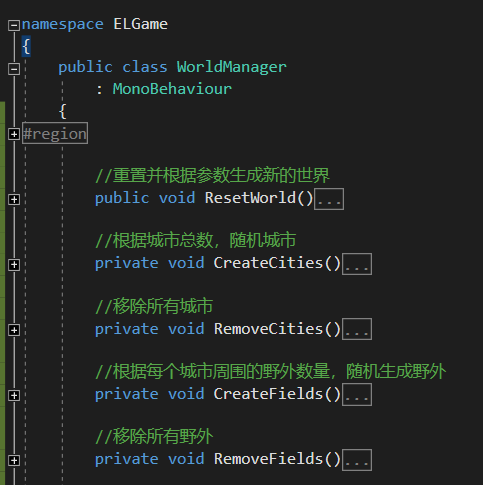
① **它要看起来要很专业**。

② 可以动态、随机生成一定数量的城市，野外。

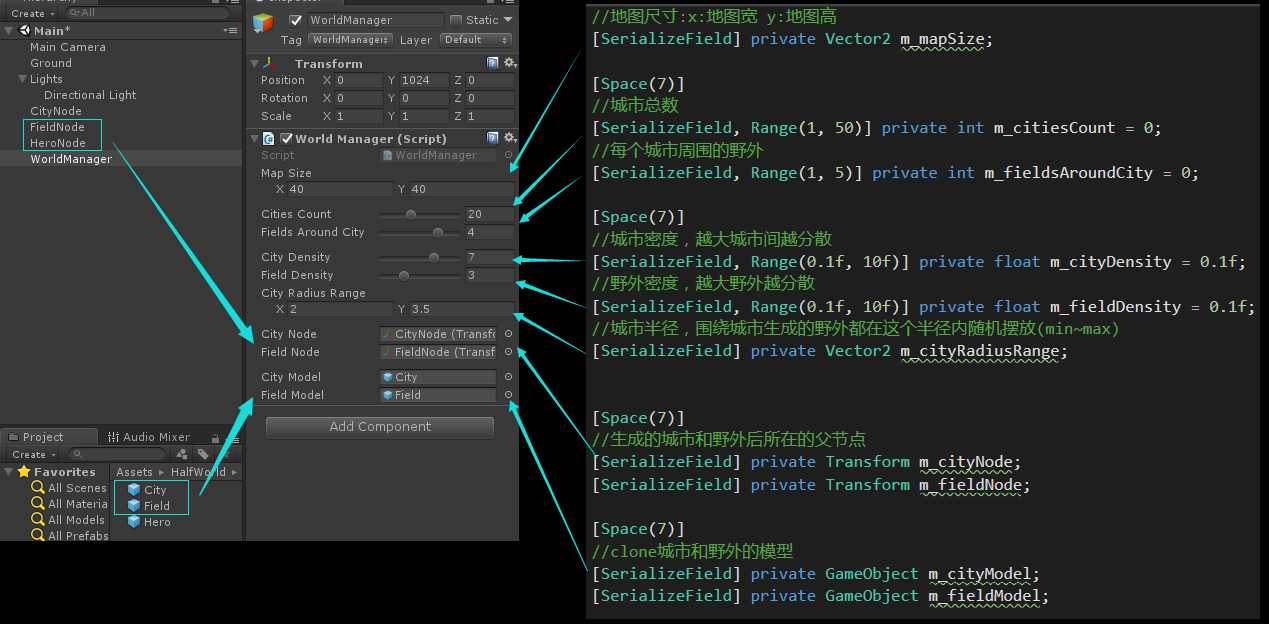
③ 城市尽量要摆的分散，**避免过分的集中，尽量均匀分布**。

④ **野外围绕城市摆放**，这样建立在均匀分布的城市基础上，它们也会相对均匀，同时也更合理些。

针对以上需求，我们只实现一个简单版的世界管理器即可，它的接口及主要功能如下：



在场景中建立一个世界管理器对象，贴上这个脚本并修改Inspector面板上的参数：



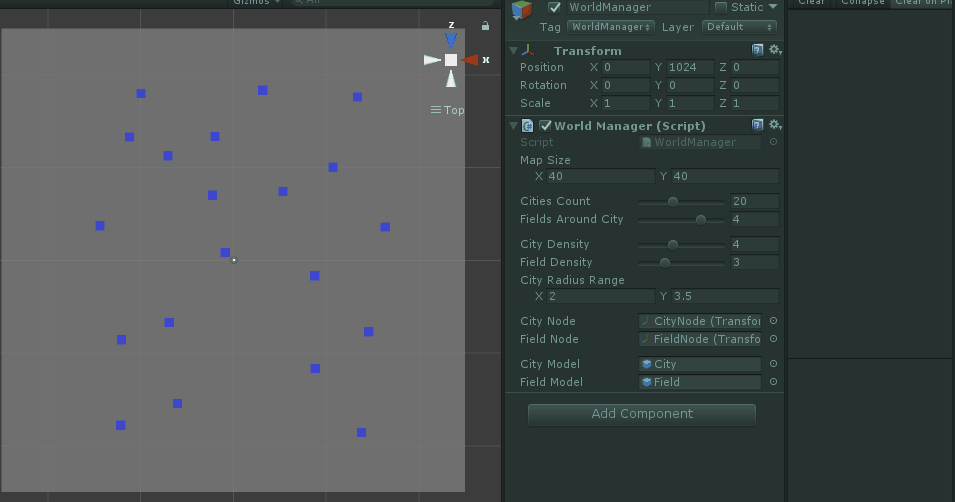
简单总结一下这个世界管理器的关键功能：

① 给定**地图范围、城市数量、城市间的密度系数**，在随机位置生成城市（蓝色方块）。

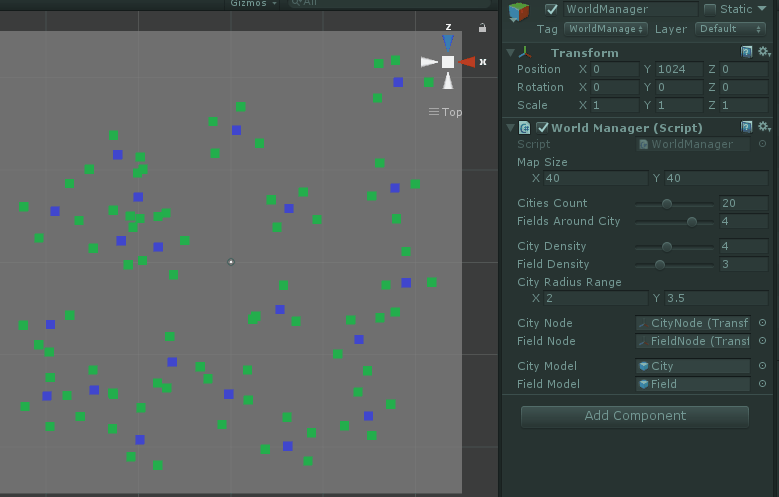
② 给定**野外生成半径、野外数量、野外密度系数**，围绕每一个城市随机生成野外（绿色方块）。

实际运行起来感觉像这样

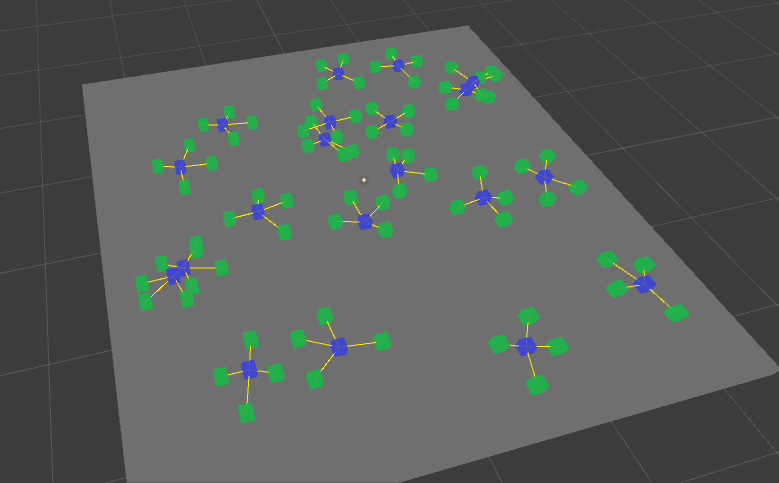
随机码放城市：



围绕城市随机码放野外：

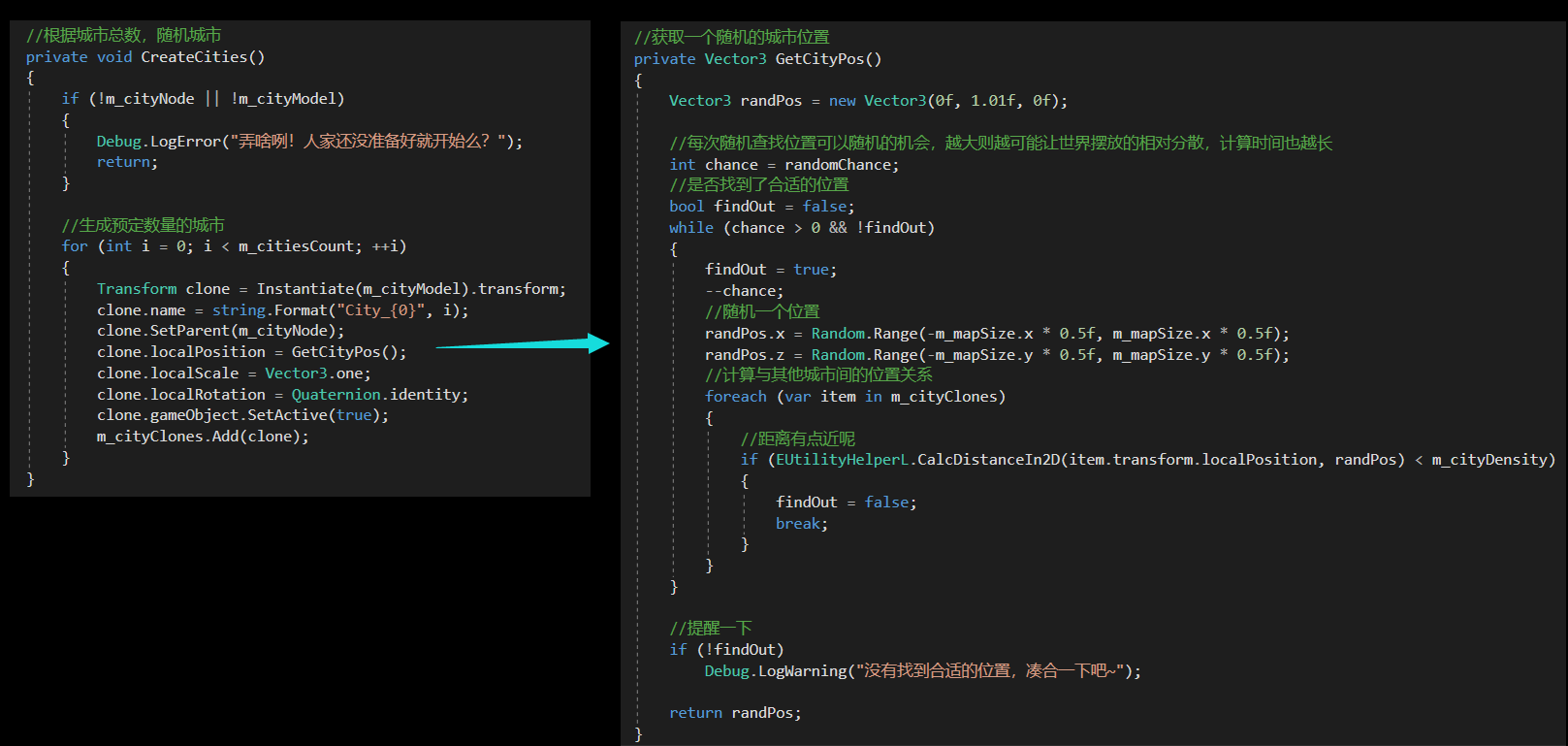


为了显示野外和城市的关系，用**黄色线段将城市和野外进行连接**，**Scene视图**下Gizmos显示为：

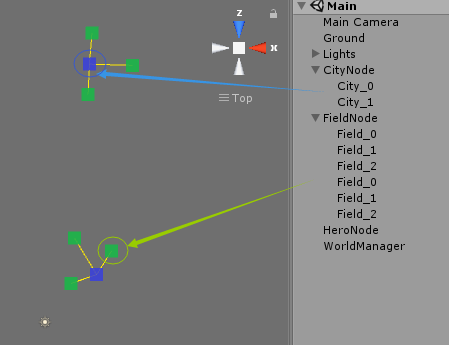


**部分关键代码**

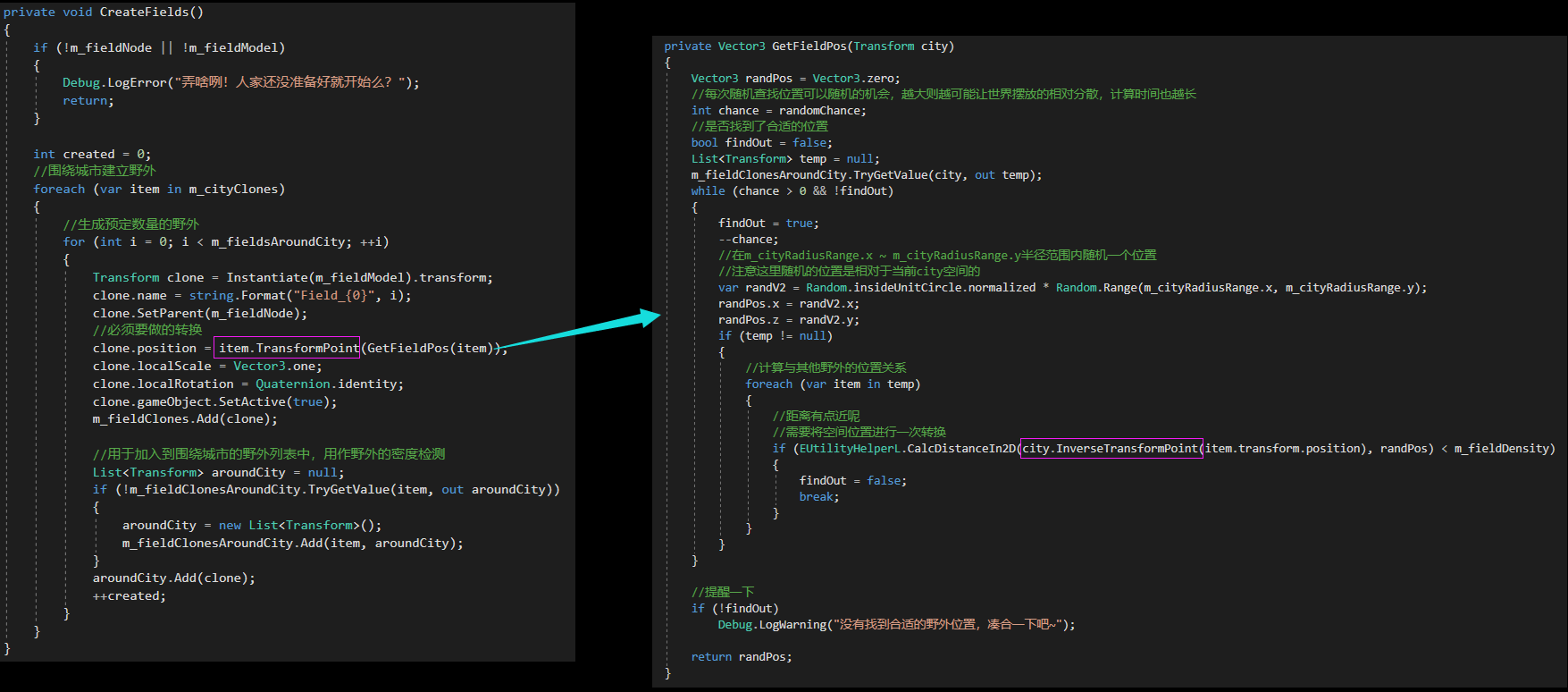
1、在随机放置城市和野外时，采用的是非常简单的**咣咣咣重复试验**的方式：



2、尽管城市和野外的逻辑关系是：**野外围绕城市**，但实际层级关系它们是**分数在不同的节点之下的**：



因此在在计算野外位置时，需要**进行空间转换**：



**小结**

此回对小游戏的样子进行了简单描述、绘制了草图，在打消了放弃的念头后创建了游戏项目，并添加了随机码放城市、野外的功能。

下回预计将把英雄引入世界，可自主探索周围的世界，具体的我也不知道会弄成啥样，因为还没开始。

附上项目的GitHub位置，有兴趣可以去取:<https://github.com/elsong823/HalfWorld>

最后，希望自己能坚持做完这件简单的事情，**下回见。**