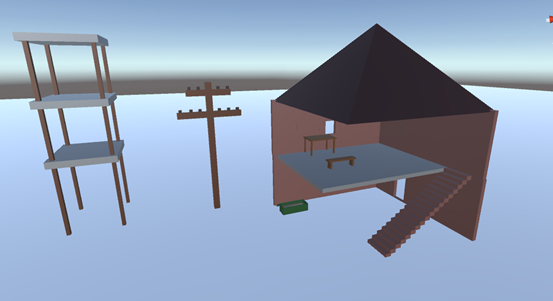
# **Unity完全自制游戏纸箱战争项目记录（20180706）**

作者：[Pluto](http://gad.qq.com/user/index?id=2267380)

链接：<http://gad.qq.com/article/detail/214988>



今天完成了一些场景模型的布置搭建工作，还在发愁项目优化的问题。

AI的总体控制完成了，但个体AI离真正的实现还有很长的一段路要走，今天特意抽出了一些时间来思考个体AI的设定。

首先可以肯定的是，个体AI肯定是要比控制AI更加复杂多变。

其次，今天突然之间想到，如果是用AI来模拟真实的玩家，那么真正的玩家在游戏中的行为肯定会是无法预测的，并不会按照某一固定的模式执行规则。

因此之前设想过的通过需先编辑好的AI动作方法就只能放弃了，重新整理思路。

一定要确定的是一点，游戏中是分有不同的职业的，现阶段是分为突击兵、工程兵、反载具兵、医疗兵、狙击手。

在敌人是医疗兵或者是狙击手的情况下并不能对敌人的载具造成伤害，那么载具肯定就不会是他们的目标。

而工程兵或者是反载具兵的优先攻击目标肯定是威胁更大的地方载具单位。

医疗兵则会优先治疗友军，其次才是击杀敌人，狙击手则是要远程击杀目标。

考虑到这点，我就思索是不是应该把个体AI的寻获目标方式给封装成独立方法，因为并不需要时刻切换目标，和控制AI脚本一样，每隔一段时间调用一次就可以了。

在确定敌方目标的时候，我使用到之前写过的Camp脚本上的阵营，这样一来就需要多次的获取组件，担心多次的获取组件会影响到游戏的性能，对本就高负荷的游戏再增加更多的负担。

查阅资料后，发现获取泛型组件是相对节省资源的方法，但尽是这样还不够，考虑到了在遍历循环中每次遍历判断的时候都要获取几次组件，干脆把获取组件集体构造一次，这样应该能节省一些资源。

在获取了目标之后，就应该是对目标发起攻击了，这点考虑到了还有进攻目标点的原因，我认为应该添加一个追击时间，在追击了一段时间后，AI则会放弃追击，进攻目标点。

到目前位置，没有框架的致命缺陷就暴露了出来。

自我反思，在项目确立的时候就没有构思好全部的步骤，有些功能的实现只是因为一时的兴起而制作了，这样反而是打乱了整体的设计路线，如果有一个成熟的框架，代码编写则会简单很多。

早在项目建立阶段就没有考虑到这个问题，对自己的能力认识的还不够清晰，同学制作的手游几近完成，而我还卡在敌人的AI设定方面。

不过这也并没有打击到我的自信心，毕竟是PC游戏，复杂性远非那些手游所能比拟的，这对我是一次锻炼，也是对前段时间学习的一个总结。