

1. 我们团队参加这一次实训的目的不只是为了完成学习目标，更多地是为了做出一点属于自己的作品。现代社会都是高度专业化分工的，只有依靠着团队的力量才能完成更加伟大的事情。据我了解，很多身边的软件工程师、设计师同学对游戏开发有兴趣，但是苦于没有一个完整的团队。游戏作为一种技术和艺术相结合的产物，对各个领域要求都很高，所以组建团队是在所难免的。所以这次实训就给了我和我的同学们一次机会来合作，群策群力，融入自己的idea，最后做出属于我们自己的作品。(舒服.jpg)
2. 当然了做项目肯定会遇到各种各样的坑- -。例如说在开发的过程中，unity 2018 1.4f突然用着用着报了个“申请60多G内存”错然后崩溃了。我们在一天晚上排查的时候开始了“甩锅大会”，网络同步、FPS模块、GUI、模型面数、骨骼动画等都排查了一下，然后又在审查他们之间集成而产生的bug。然而这个bug必须修啊，因为游戏在玩的时候总不能无端端地崩掉吧。结果我们还在以为自己代码写了什么不该写的时候，unity在编辑模式崩掉了。于是在项目进行了一半的时候，我们顶着压力果断把unity换成了2017，项目里很多细节因为兼容性的问题要重新制作。结果之后再也没出现过问题了。不说了，先把unity钉上耻辱柱上。

还有一些bug在单元测试的时候没有显露出来，只有当各个成员各自开发的模块进行集成测试的时候，才有些bug冒出水面。就例如网络模块和AI模块，两个模块看上去只有简单的耦合，就是把AI的一些状态同步到其他客户端去。结果集成测试发现，某一个晚上，AI状态根本无法同步。原因居然是自己造轮子实现的网络同步的远程过程调用(RPC)，只支持了函数名的区分，然后AI模块在下午的一个commit之后，重构了一下，加了几个重载函数，导致在RPC的时候在接收端的反射(Reflection)时候产生了二义性，导致没法同步。

当然还有很多其他问题和困难啦，例如很多游戏策划的需求都被不断砍了，因为要先保证游戏完成度，所以只能牺牲部分游戏性和细节；团队协作方面也有很多的坑，例如一开始用github进行版本管理的时候经常发生冲突(conflict)，后来大家都比较熟悉团队协作开发和版本控制之后就越来越少conflict了；

1. 这一次实训相比起以前的学校学习、训练有很多不同。第一个，实训的目标是做一个完整的项目出来，这就需要踩很多书本里面不会跟你提到的坑。比起以前的相对trivial的“项目”，实训项目对整个团队的要求都高了很多。第二个，这次实训是一个团队协作，以前的大作业一般都是单人完成，就算是组队完成，成员间的合作和耦合程度从未达到此次实训的程度。这就非常考验团队的合作、沟通、协调能力了，这些实战经验是书本不会教的。第三个，短时间做一个完成度较高的作品出来需要很高强度的工作，这堆我们团队所有成员来说都是一次历炼。

4.这次校企合作的感觉挺好的，我们团队其实也一直想接触一下业界，并参考下业界的做法，包括工作流(workflow)和部分技术细节。网易游戏的导师也给了我们很多的指导和帮助，再次感谢。