# MMO游戏数值框架概述（偏模拟方向）

作者：雪焰  2011-8-14

**以下为文字内容：**

**概述：**

**本文讨论了以下几个模块：**  
　　1.国内游戏数值的现状  
　　2.游戏数值策划所需要具备的素质  
　　3.数值策划的职业规划  
　　4.数值策划对游戏能控制的程度  
　　5.框架设计和一些数值技巧  
　　6.计算方法  
　　7.设计思路问题

　　注：本文观点是建立在成为一个好的数值策划上，而并非是仅仅是成为数值策划，个人觉得，任何人，任何水平都可以做数值策划，但并非任何人，任何水平都能做好数值策划。本文主讨论框架设计，不讨论细节地方。也许前面的叙述会较长，请耐心看下去，也许会有您想要看到的东西。想喷的朋友随意！  
  
  
**国内数值现状：**

　　无论是看到，还是听到，甚至被面试，总有人议论魔兽世界为什么会成功，或者说国内游戏和魔兽的差距在哪，而且如此经久不衰，许多人给出了各种各样的答案，而又有很多人按照自己的答案去做自己的游戏，但事实证明，并没有游戏成功（注：本文并非想讨论魔兽如何如何，仅仅为引出关于数值的讨论）。

　　作为一个游戏玩家，也许会说，它好玩，但这是一个毫无意义的答案，对于一个游戏开发者来说，基本就会从以下几个方面来考虑，运营，市场，研发。

　　市场和运营，没什么好说的，大家都知道，魔兽世界的成功和它们没有非常大的关系，而且作为游戏开发者，更关注的是研发层面。

　　对于研发层面，关注的就是最基础的三个点，技术（程序），画面（美术）和游戏性（策划）。对于技术来说，我咨询过不少的资深程序，确实，国外比国内要强不少，但顶多是功能更强或者更全面一些，肯定没有强到有巨大的差距的程度（此处指的是重大功能缺失，并非指一些锦上添花的东西）。

　　美术，国内的美术水平不会比国外差，甚至某种意义上来说，比国外更好，而且对于魔兽来说，其画面肯定比不过09年整得风风火火但又很快销声匿迹的永恒之塔。

　　策划，对于策划方面来说，分的可能也就三方面，主要为系统，数值，剧情。许多人说魔兽很大程度上因为剧情而成功，但并非是这样，魔兽的剧情是建立在魔兽3的基础上才广为流传的，因为有了魔兽3的成功，才有玩家对于其剧情而津津乐道。而并非因为剧情才导致魔兽3的成功，当然，也不能否认因为魔兽3的剧情的原因，给魔兽世界吸引了一大批的用户，但这不会是大量用户一直留在魔兽世界中的原因。除了早期的单机游戏外，就几乎没有过会因为剧情而留在网络游戏中的玩家。（注：此处并非想讨论剧情是否重要的问题）

　　系统，魔兽的系统做的确实相当出色，但相对而言，我见过不少的制作人，也听说过不少的设计理念，然后也看过和研究过不少游戏的系统，发现一个问题，国内不少游戏的系统，无论是关联性还是设计的巧妙性都不比魔兽差，甚至有的比魔兽好，但为什么却达不到魔兽的高度？

　　最终的结论是，国内的数值水平远远落后于国外，而且并非设计思路的落后，而是执行能力的落后。为什么这么说，从以下几个方面来说明：

　　首先从人员配置上来说：很少有公司的数值组会超过5个人，一般都1-2个，这还是大公司，而对于小公司来说，甚至一个都没有，而仅仅是一些从来没干过数值而对数值有兴趣的人在兼职。也许有人会说，要那么多数值干嘛，活有那么多么？其实对于策划而言，工作量最大的就是数值，游戏内的所有东西，除了资源和命名以外，所有东西都和数值有关。

　　其次从重视程度上来说，在早期，因为市场原因，导致游戏只要能上，就能赚钱，而且数值本身就是一个在大概念上兼容性比较强的东西，即即使有不平衡，有数值问题，游戏一样能够赚钱，而更有很多的人觉得，即使没有数值，或者有简单数值，游戏一样能火，而在近几年开始拼游戏品质的时候，更多公司更关注的是引擎的先进，画面的优化，功能的强大，却很少有人关注到数值这一块。甚至非常多人觉得数值策划仅仅只是一个执行，给系统干活的。而且非常多人，甚至非常多数值本身都觉得，数值是靠测出来的，而并非算出来的。但我想强调一个概念：如果数值能力够，是可以不用测试的。（也许这句话很多人会嗤之以鼻，后文会具体讲述原因）

　　再次从业者的素质来说，很多数值策划几乎都没有数值背景，不懂得好的计算方法，如何能够快速高效并准确的计算出值。曾经看到过一篇帖子说，小学四年级就能做数值策划，说实话，我很想笑，我并不否认小学四年级就能做数值策划，但要做好的数值策划，这个说法纯粹做梦，也许有人会觉得，数值策划有那么复杂么？我可以很负责任的说，如果想做精品，那么数值策划就会那么复杂，复杂并非取决于数值本身，而取决于游戏是一个大世界，也许有人会说我这么光说理论是没有说服力的，那么举个例子，大家都知道暗黑2，这个游戏10多年了都经久不衰，到现在还N多战网不停开荒，想来大家都知道这个游戏里面只有这么几个系统：装备（包含神符之语）、掉落、技能，一个纯数值游戏。而它抛开操作行的话，就主要玩一个装备搭配，从几千几万件装备中选出12件（包含副手的两件）。这中间就是一个最优化模型的问题，试问一下，小学四年级的数值策划能够算得清楚？或者说即使他用加减乘除算清楚了，那么得花多少时间？

　　大家提起魔兽必然会提起魔兽的技能和天赋系统，但实际国内游戏一对比，就会发现，比魔兽的技能和天赋系统做得复杂的游戏很多，但为什么没有魔兽的影响力和游戏性，当然，这中间确实有操作性的成分在内，但抛开操作性，也是能看出一些问题的。

　　也许会有人觉得，其实暗黑的数值策划也许没没有算清楚，那么咱们还不说这么复杂的问题，再举个简单的例子，一个套装有四件装备，现在有三套装备，一套是平衡装，一套是攻装，一套是防装，在套装激活属性不起决定影响的基础上，这12件装备穿哪四件会最占优势，怎么穿会对其他穿法有什么影响，我相信国内的数值很少能够清楚的说明其计算方法并对其设计出有效的数值来。再也许会有人觉得，咱们干嘛弄这么复杂，管他怎么玩最优化，反正只要没问题就行了。那么我就想问，既然设计者自己就不知道会有啥结果，怎么可能指望玩家按照自己的设计来玩？怎么能确定没问题？

　　这还仅仅是12选4，如果是24选8，抑或是更多，那么在设计者都不知道玩家会怎么玩，只能靠市场去检测的情况下，碰到运气不错，那么这游戏就好了，碰到运气不好，那么这游戏就萎了，打个比方，很多游戏都有加点玩法，这个玩法的初衷实际上是想让加点变得有选择性（暂时抛开加点属性占人物属性比例较低的因素），但是由于数值无法计算清楚，导致很多时候，这种选择会变得唯一，即会存在最优配点方法，那么原先想达到的选择目的就无法达到，后果就是轻的会废掉加点玩法，重的会影响到其他玩法（如果加点玩法跟其他较多系统关联的话）。

　　在一个庞大的系统架构中，一个系统连着一个系统，如果关联性没那么强，发生达不到目的的情况还算好，但是如果关联性较强，则会发现，一个链条的中间断了，然后后面一堆全跟着掉了。

　　或者再举个例子，大家众所周知的装备升星玩法，只要是个数值都会涉及到这里的消耗计算上，那么，先举个简单的例子，规则如下，升星失败不会降星，1到9星有不同的成功率，现在要计算升到9星的平均消耗，相信绝大多数人都能计算出来。但事实上会发现，几乎没有游戏公司用这种规则。进一步，失败会降一星，再计算升到9星的平均消耗，相信就有90%的人算不出来或者算错了，这种方法下，不懂概率论的人想要计算出来就有点搞笑了。或者再复杂点，失败有X%几率降一星，Y%几率降到0，有Z%的几率不降，再计算平均消耗，我相信业内大部分数值是不会计算的（具体算法我会在后文中讲述）。

　　再次，对于工具的应用上，目前来说，对于数值策划最多的用的是excel，但大多数数值策划的应用水平仅体现在简单的填表，简单的函数上，对于熟练使用查找函数，数据有效性，条件格式，甚至是更高层的数组公式，VBA，甚至是数据透视表，更甚至ribbonX的少之又少。不会excel，无法建立起能够关联，整体计算的数据模型，那么这个数值策划至少不能算是一个成功的数值策划（当然，我并不否认存在凭感觉成功的数值策划，但这并非常态，需要包含运气和大量经验在内），而业内绝大多数的数值策划都达不到这个水平。如果对于excel这个基本工具都不能保证掌握的话，那更不用提掌握其他sas，matlab等数学专用工具了（当然，这个我也不太精通，学习中）。

　　最后，在于数值的教学上，一般来说，国内的数值策划大多都是自学成才，而这些数值更多都感慨一个问题，就是无法学习到一些比较先进的计算方法，每次听说好的计算方法都会眼前一亮。面对一个游戏，可以很容易的反推出其系统关联性，但几乎不可能反推出其数值框架、数值公式和给值思路。所以导致数值策划的成长大多都是自悟和经历项目的经验，能够学习知识的面非常窄。

　　然后打个比方，系统策划好比“前锋”，而数值好比“后卫”，系统策划容易出彩，而数值策划即使能力非常强，做得非常好，也仅仅只是让游戏没有问题，很难有比较出彩的地方，即较难看到效果，所以大部分人都会奔系统而去。

　　所以，以上种种，都导致了一个问题，国内数值水平普遍偏低（并非贬低别人，我自己能力也偏低），并且因为市场的原因，导致很少有人认识到这一点。所以国内很少出或者很少复制出精品游戏。  
  
  
**好数值策划所需要具备的基础素质：**

1.必须至少深入玩过几款网游：

　　这一点主要是，如果数值没有玩过网游或者说玩过太少网游的话，首先是无法理解系统需求，无法明白游戏元素，其次对于玩家心理完全不理解，设计思路会成问题。

2.熟练应用excel，并有好的工作习惯

　　熟练应用包括（按照深入程度划分）：  
　　基本功能方面：拉表、简单公式（sum，if等）、插入、删除、隐藏、冻结窗格  
　　进阶功能：简单排序、筛选、查找与逻辑函数、选择性粘贴、定位  
　　更高一点：名称、条件格式、数据有效性、高级筛选和排序、分列、删除重复项  
　　再高：表、数组公式、数据透视表、VBA  
　　其他：ribbonX

3.好的工作习惯：

　　不能在公式中加入常量（当然，1,0这些特殊数值除外）  
　　多使用名称，让公式简洁明了  
　　工作表sheet分类明确，某些表就是变量定义，某些表就是模块计算

4.有数学背景或者对数学算法了解较多

　　该点主要是用来计算一些较复杂的算法，设计一些要满足需求的公式等（例如后文中伤害公式的改造公式）

5.有逻辑思想，最好懂一些程序（首推VBA，其次C++），自己个人是lua用的最熟，其次VBA，再次C++

　　因为数值经常涉及到超大量的计算，有时候仅仅靠算法是没有用的，更多的时候还需要进行大数据量的遍历计算，这 时候靠人工去计算肯定是不现实的，这时候就需要有程序逻辑进行循环。

　　也许有人会说，有程序需求找程序就行了呗，但是，很多时候这种想法是不现实的，首先，程序不一定有空给你做东西，其次，程序不一定能理解你想做什么，再次，对于数值来说，大量数据是建立在excel上的，而程序开发的东西大量是建立在C++上的，不大好兼容，当然，你确实是可以找一个熟练使用VBA的，但这种绝对是可遇而不可求的。而且最关键的是如果要修改，自己来做重复工作量小。

6.懂系统关联

　　做一个系统策划可以不懂数值，但一个好的系统策划肯定对数值有不少的了解，同样，一个数值策划可以不懂系统，但一个好的数值策划必须对系统有较深入的了解。

　　并因为玩家如何玩一个游戏最终还是取决于数值策划的值给的如何，所以，事实上，数值策划必须清楚一套数值给出去之后，其所服务的系统跟其他的系统的关联性会怎么样。也就是说，系统策划定义的系统关联是否能够达到预期效果是由数值直接决定的，所以，数值策划必须要理解系统的关联以及系统所要达到的目的。

　　举个例子：某个BOSS掉升星必成的道具和金币，这个BOSS究竟与道具关联更重，还是与金币关联性更强仅仅取决于道具的掉率和数量，以及金币的掉率和数量。而系统策划仅仅只能告诉你他想要的是：道具多产点，金币少产点，但究竟是多少，他其实自己都没有太确凿的概念。所以数值必须清楚中间的关联度有多高，而给出相应的数值。  
  
7.要明确玩家如何玩

　　这个是属于比较难的，就是一个系统案子做出来之后，首先要清楚设计目的，其实要想好玩家如何去玩。自己先设身处地想到玩家有多少种可能的玩法。玩家是交易、卖店、不玩、还是其他情况。

　　就比如说，同一等级段有几套套装，那么首先得想到玩家穿齐一套的，但其次，还必须想到玩家穿散件的情况。甚至跨等级段穿装备的情况。

　　当然，以上仅仅是最关键的几点。其他的就暂且不说了。

　　总结语：评价一个数值策划最重要的标准在于：他对于自己所设计的数值的把握程度，或者说拍脑门的次数。

　　理由：游戏开发的流程：设定->执行->测试，然后从测试这里进行一个while do的循环，while测试不通过，则do执行->测试继续循环下去，而这个循环的次数有多少，就是评价一个数值的最好的指标。对于一般数值来说，自己并不清楚或者是靠感觉大概知道给出的数值是个什么情况，然后只能靠测试反馈，反馈完之后，再测试，再反馈，差的数值会将这个过程拉到很长，而导致大量的测试工作量，执行工作量，而产生大量的工作周期。而好的数值则会将该过程无限往下压缩，接近于1，而节省大量的工期。如果哪个数值能够1次给值就没有任何问题，那么无疑，他的数值能力绝对是强到一定程度了。因为他知道自己给的数值有什么问题，能够达到什么目的，在什么地方会盈余多少。游戏的一切尽在掌握之中，这是数值策划发展的最终目标（也许这个目标确实很难达到，但是可以无限接近，在接近的过程中，你会发现，你给项目带来了巨大的效益，无论是缩减工期，还是对游戏的掌握）。

**数值策划的职业规划：**

　　仅指如何入门，如何发展，如何学习，不包括爬到管理之类的

　　首先，数值分为设计和执行两种，执行是仅仅懂执行，计算，不太理解系统关联，而设计是指既会计算，也懂系统的人。那种只懂设计不懂计算的数值，我将其定义为系统，而并非数值，而这些人做数值更多的靠的是经验和想象。数值计算是一个严谨的过程，不可能存在任何的猜测和直觉。很多游戏中存在很多“拍脑门”的值，严格来说，一个游戏中，几乎只允许存在几个“拍脑门”的值，其他都必须是经过严格计算出来的。或者，评价一个游戏数值好坏只需要看里面有多少拍脑门的值就行了。

　　对于应聘来说，纯系统的数值找工作跟系统一样，主要看运气，能碰到那个喜欢你思路的人，你就牛B，不喜欢你思路的人，你就SB。而现在业内更多的数值是这一类型。而数值的从业者更多更容易变成这样的数值。除非曾经镀金过（即在一些成功项目中当过数值），才会相对受重视，否则，能不能打动面试者，就看你是否对他胃口或者会不会侃了。

　　执行数值，对于执行数值来说，首先也得excel熟悉，其次懂计算方法。门槛相对较高，但一旦进入门槛后，就会发现，即使你说不出来设计思路，即使你没呆过成功项目，你也会很抢手。因为即使你对于设计没有帮助，但执行效率绝对远高于其他策划。

　　而对于既会执行，又懂设计思路的数值来说，绝对是各个公司热门要挖的人物，而这种人，也一般都具备了主策的能力。如果曾经呆过成功项目，那么绝对可以往总监、经理上爬了。

　　所以，综上，对于想要做数值策划的人来说，给一句忠告：先学会excel，再来谈设计。个人经历可以完全证明这一点。（如果您口才特别好，那么可以忽略这句话）

**数值策划的模拟表能控制到的程度：**

1.人物属性：

　　不包含随机玩法，可以保证游戏中人物的数值和计算表中完全一致，小数点都不差  
　　包含选择玩法，根据其选择可以将游戏中人物的数值和计算表完全一致  
　　包含随机玩法，可以保证游戏中人物的数值绝对不会超过数值设定的上限

　　总结：也就是说，如果在计算表中按照游戏内的装备配置，可以保证游戏内的数值和计算表中的数值完全一致，这是数值策划把握游戏的基础

　　主要用来控制游戏中的总战斗数值体系

2.技能：

　　根据玩家所配置的技能循环和应对策略，可以保证游戏内的单人战斗和计算表中的模拟战斗结果基本一致（因为涉及到操作准确性以及伤害浮动，暴击等，只能保证平衡值基本一致）（文尾会配表）

　　主要用来控制和模拟玩家的单人PK平衡

3.怪物数值：

　　根据当前人物能力，以及所需要怪物达到的目的来决定怪物数值

　　目的是指：怪物打人多少血，人打怪的效率，以及怪物属性引导，例如某怪闪避特高，需要玩家堆到多少的命中才能杀死等等

4.成长：

　　根据有哪些一次性任务（这些任务中哪些即时能完成，哪些需要一段时间后能完成），以及每日任务，再模拟大部分玩家的行为，可以一次性准确的设计出玩家升级所需要的时间，经验，以及玩法经验公式等

　　主要用来控制和模拟玩家的成长速度和经验获得

5.经济：

　　根据玩家每日的行为，所进行的玩法，可以完全的计算出玩家身上应该有多少钱和什么物品（不包含交易和掉落），控制和模拟每日的产出

　　商城定价之后，能够平均的控制住自己每个玩法的大概收入

6.总结：

　　主要为以上几个大模块，为什么招聘数值策划的人都会提到一个数学建模的问题，而数学建模为什么要建模，建立的就是一个数学模型，而数学模型的目的用来干什么，相信大家都知道，就是用来模拟的。也就是说，如果一个数值策划不会数值模拟，那么绝非一个好的数值策划。

　　数值策划计算模拟所能达到的最高境界：输入玩家的行为模式，输出玩家的等级、经济、物品、装备情况等数据库所记录的全部属性。

　　以上所说的仅仅为对于游戏数据的控制，我一直有个观点：只有建立在你能控制游戏数值的基础上，才能让自己的设计变得有根据。很多人在进行讨论的时候都会说玩家喜欢喜欢怎么样，到多少级可能会怎么样。很多东西都是凭个人观点或者说个人经验去猜测出来的，这些说法不可能都靠谱。

　　只有建立在能够了解游戏中的各项数据的情况下，再去设计东西，才能够准确的达到目的。

　　数值框架与一些设计技巧（注：数值框架涉及到的框架内容和细节内容太多，不做详细说明，仅仅只描述一下大概框架的设计流程）

　　相对于设计方法，我更擅长于模拟，所以后文更多的讲到一些模拟的技巧，敬请关注。

　　虽然以下介绍了一种比较简单易懂也容易实施的设计思路，但仅仅只是为了抛砖引玉，并不推荐去深入研究或者照搬，较好的数值策划基本都会有自己一套独立的数值架构和设计方法，如果总是照搬别人的，永远都成为不了好的数值策划。

　　数值框架设计，许多人不明白数值框架指什么，下面一步步的说明，先简要说明，再详细举例说明：

　　首先，先明确自己的游戏里面有什么，而这些有什么仅仅是一个框架的组成部分，例如：有职业、属性、等级、怪物、装备、技能、坐骑、货币、掉落、道具等一系列的东西。相当于先设计出游戏里有哪些“积木”。

　　其次，需要将这一系列元素组合和串联起来，职业有哪些属性，装备有哪些属性，等级对装备和属性有什么影响，货币和道具对装备有什么影响，掉落对道具和货币的影响。相当于“积木”搭建的方式，谁搭在谁上面，谁会承载谁。

　　因为现在MMORPG的战斗数值设计起来框架和流程几乎完全一样，并没有什么变化特别大的地方，所以按照常见流程进行说明，但并非指数值设计就一定得按照以下规则设计。而且以下仅仅为一些常见方法，一些细节东西不提（例如职业特色如何搭建）

*举例：*

　　战斗模块。设定游戏属性与伤害公式，这是整个游戏战斗的基础

　　游戏属性就不用多说了，按照各自游戏的风格和具体设计来设定自己所需要的属性即可（本例子中是使用的无一级属性的概念，即没有力量、智力等属性，在现今的游戏当中，他们其实并没有太大的实际存在必要。当然，也有相应的设计方法，但不在此详细叙述。）

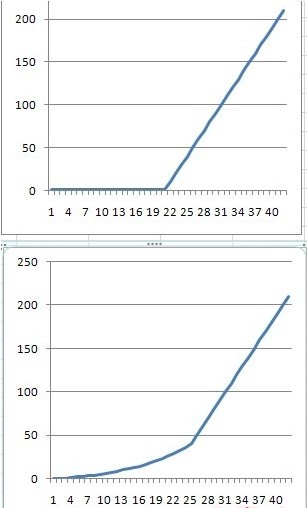
　　伤害公式：一般来说，从公式的主体上来说，是减法和除法两种公式，当然，从本质意义上来说，现在的公式远比这复杂，但是无论是数学还是物理，都很常见一句话：理想状态下，即某些东西先不管。

　　我不想在这里讨论减法公式好还是除法公式好，仅仅只想说各有各的优缺点，将伤害公式稍作设计和修改即可弥补。

　　关于减法公式，最大的优点就是简单明了，而且伤害增长明显，即攻涨1点，防不变，伤害就涨1点，但最大的缺点就是不破防的问题，在不破防的情况下，会出现很多问题，一是打怪无消耗，二是人物追求攻的过程中，如果攻一直没对方防高，那么一直不破防，追求攻的过程会比较受打击，当然，还有其他一些问题，暂且不提（例如很难控制上限值等）

　　有个简单的设计方案：  


　　该方案的公式看起来比较复杂，主要目的就是为了解决不破防也会有伤害的问题，以及在追求攻的过程中能看到实际效果，也许有人会说公式太复杂，程序的计算压力会太大，但配合数值的设计（攻跟防的差距拉得稍微大点），让该功能的出现几率变小点就没有太大问题了。

　　对比一下实际效果：  
  
  
　　上图为攻-防的实际效果，下图为使用简单设计方案后的效果

　　至于除法公式，有一个最大的好处就是能够控制上限，并拥有一个递减规则。但同样拥有一个较大的问题就是，一旦到达某个阶段后，收益递减，玩家评价如何该收益已经不值得的时候，该数值就几乎无法再往上叠加了。该公式不做再进一步的说明了，如果有人想讨论的话，可以单独聊。

　　伤害公式确定后，那么就是职业和属性，以及战斗平衡了。首先明确一个概念，那就是平衡的概念：平衡并非指谁的攻击高，或者谁的血多，或者[来源：GameRes.com]谁的防高，抑或者是谁的闪避高，而仅仅是这么一个公式：生存时间A≈生存时间B。以下以减法公式为例

　　为什么这里是约等于，就是因为玩家有一个感受范围，在同样的装备档次下，超过他的感受范围了，他会认为不平衡，在公式上就表现为生存时间A和生存时间B的数值差别的精确范围，我暂定为0.2，即生存时间A/生存时间B的数值范围在0.8~1.25之间（注，此处因为0.8算为平衡范围内，则A/B=0.8，而B/A=1.25，所以定义为0.8~1.25而并非0.8~1.2）

　　即我认为A/B在0.8~1.25的范围内，则为平衡，当然，在这中间可能会有谁强谁弱，但总体差别不会太大。

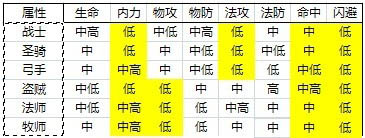
　　在减法公式下，生存时间B=生命B/（攻 A–防B），这个公式有一个好处，即如果在基础的人物属性模型下基本平衡的话，各个职业的数值按照一个固定的比例系数成长的话，生存时间是不会变化的。例如：A和B都升到了40级，A的攻为原来的10倍，而相应的B的防也为原来的10倍，血也类似。则除出来的生存时间是和1级的时候完全一样的。

　　也就是说，在这个理论下，可以建立一个在大方向上基本平衡的模型。即在各个玩法上固定攻、防、血的成长比例一致，则会出现平衡模型。也许会有人误解，这样是否会导致各个职业属性都一样，并非如此，如果你仔细想，你会发现各个职业的属性倾向会非常明显（注：这个方法是一个取巧的方法，做起来相对简单而且能够在一定程度上保证平衡性，而且还有职业属性特色，但也有不少的局限性，并非这么做是最好的，所以仅仅作为介绍来叙述一套设值思路，觉得有价值的项目可以使用）

**设定职业模板**

　　首先，先设定一个所有职业共用的模板，例如：这里的各项数值可以根据自己的喜好凭空产生，不过即使是这样，中间的数值也会有一定的来源，例如生命、攻、防的比例是一个需要注意的数据，即你希望你的职业在光看属性的情况下的平均生存时间。内力可以随意给，它只跟使用技能时消耗多少有关系，所以，初始设定多少都无所谓。至于命中闪避，则看其数值空间有多大，最终能放大多少倍，可以根据后面设定的放大比来进行调节。

http://upload.gameres.com/20118/sf_16175847_679.jpg  
  
　　其次，设定职业特点，并根据职业特点设定职业的属性倾向，然后使用该数值模板进行偏向，如下（其中，中代表和模板数值一样，低代表低两档，中低代表低一档，然后在系数表中配置每低一档的差距，这里每档的差距可以用等差数列，等比数列，或者是枚举数列都行，仅仅是为了体现职业属性倾向，以下以等差数列为例，每档差距为25%）：

  
  
　　偏向完之后，就会产生1级时各个职业的属性数值，如下：



　　然后根据该数值，先简单计算一版的数值平衡（注：该表仅为示意，里面包含的是各种取值倾向下的各种职业的战斗平衡数据，选择该项倾向，则该项属性为原先的1.1倍，该设置没有什么意义，可以理解为不同加点模式）（其中粉色背景的表示为不满足我预先设定的可以接受的范围内的，即不太平衡的可能性，那么观察出现这种情况是因为什么，然后返回之前的属性倾向设定步骤调整一下数值继续看，直到相对满意为止）：



　　小插曲：注意单元格中的公式：

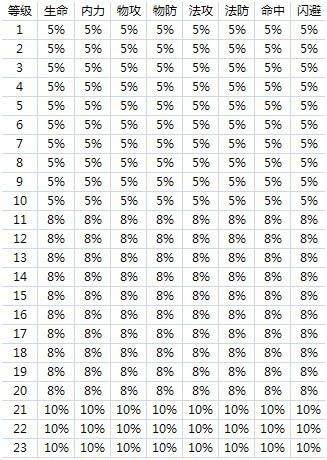
http://upload.gameres.com/20118/sf_1618013_4049.jpg

　　使用名称会让你的公式变得简单易懂

　　该步结束后，则已经完成了职业属性模板，后面一系列的东西往上放大即可

**升级获得属性**

　　因为在上一步已经设定好了职业属性的倾向性，按照百分比放大，则能保证其倾向性，并且能够保证平衡性，所以，升级获得的属性职业按照1级时的属性进行百分比放大即可

　　如下图所示：  


　　设定能力分布比例（这里主要谈攻防血这些影响战斗的基础属性，以装备系统为例，有其他系统类推即可，至于像命闪爆韧等其他特殊属性可以单独做限制）

　　假设装备系统上有基础属性，颜色属性，品质属性，其他属性，其中装备强化程度对基础属性有影响，装备颜色对装备的基础属性有影响（与强化叠加），对颜色属性也有影响，根据设计的需求，先分配一下各个部位属性所占的比例大小，如下图：其中装备总和代表的是装备对人的能力的提升是人物本身能力的多少倍，下图例子中即为9倍（仅为举例，该数值并非一定合理，不必细究）  


**设定好能力分布后，怎么来设计实际的数值，首先先从基础属性说起：**

    基础属性作为所有装备上都必备的属性，一般来说是装备的保底属性，即可以理解为玩家至少会拥有这个属性，那么对于平衡来说，把至少会拥有的东西做平衡，其他部分即使不平衡，影响也会小很多。

    而作为人物基础运算的几个属性，攻、血必须是要有的，防最好也要有，为什么这里说必须要有，一是为了保证攻防比仍旧和初期设置的一样，其次为了保证生存时间不随着玩家的成长直线变低。

    如果血不发生变化，光攻击在涨，那么一定会出现生存回合数降低的情况，那么，到一定等级的时候，就会一直困扰于玩家之间的战斗全是秒人的尴尬境地（也许有的游戏并不介意这种设计，但大多数还是比较介意）。而这种情况下，就会采取各种方法拉长生存时间，无论是降伤，还是加防，还是加血，但作为一个补漏的系统，无论怎么做，都会处于比较尴尬的境地。

    而在这里，能够保证装备的基础属性和人物基础的攻防血比例一致，就较大程度上不会出现这个问题，而装备强化一般来说都是对于基础属性的一个数值放大。也就是说，从人物基础，升级获得，装备基础属性，装备强化这么几个大块上，已经能做到属性平衡了

    而颜色属性（或者叫随机属性），不是那么好控制，那么我们只需要按照这个攻防比，作为随机的上限和下限（简单一点，下限按照上限的一个百分比即可），对于随机属性，我们只需要控制其上限值和平均值即可。

    品质属性类似，不细说

    因为关注平衡的目的仅仅是要让玩家感觉不出不平衡，而玩家感觉是建立在一个装备是同一个水平上的，如果装备有差异，那么即使不平衡，感觉很明显，玩家也会觉得这是理所当然，所以对于随机属性，如果随机范围比较大，那么压根就不用担心平衡问题。

    镶嵌属性，在各种数值模型中，宝石镶嵌都是一个比较尴尬的玩法，对于各个职业来说，如果有属性倾向的话，对于每个职业来说，每个属性的价值都不一样，而一旦镶嵌的话，同样的宝石镶嵌上的数值一般来说必须一样（如果你愿意冒风险挑战玩家的价值观念的话，可以做同一个宝石不同的职业镶嵌后出不同的数值），而镶嵌上一样的数值就会产生该属性对于该职业的价值不一样，在一定程度上会影响平衡。

    影响有多大，就在于属性玩法的比例大小了，但是，首先战斗属性已经建立在一个占比重比较大的基础上（人物，升级，强化等），所以这一块的影响就会相对减弱很多，同时，再在镶嵌部位的限制上做点小文章，让某个部位对于某些职业强些，在另外部位弱些即可。

    例如，法师血少攻高，战士血高攻少，同样的生命点数，对于法师的性价比远高于战士，同样的攻击点数，对于战士的性价比就远高于法师。所以，做如下设置，武器上只能镶攻击石头，衣服上只能镶生命石头，那么就可以将两者的优势拉平一些。

    其他装备系统根据其类型可以大概按照上面几类的方法进行设计配值。

**属性量化：**

    在设计数值的过程中，经常会涉及到属性平衡的问题，即在某一个部位上，如果只能出现一个属性，但有多个属性可供选择的话，就会涉及到属性量化的问题。

    在数值系统中，属性量化一直是一个很大的课题，很多人都在寻找能够一次性解决问题的量化方法，如果能够找到，就会发现做战斗平衡会非常容易。

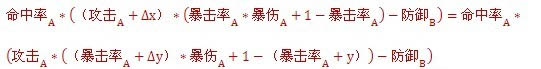
    除法公式我并未太过研究，但看过青木公写过的一份除法公式属性量化的文章，这篇文章非常有水平，分析相当深入，如果有人想看，可以在我的群空间中找到共享文章。

    我主要研究的是减法公式，可以得出一个结论：在减法公式的基础上，想做一劳永逸的属性量化几乎不可能，只能在一定数值范围内做一定程度的量化（慎用）。

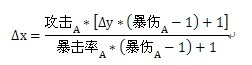
    假如定义一个简单的伤害公式为：先判断是否命中，再判断是否暴击，再判断伤害，暴击是乘以攻击，那么对于A攻击B的平均伤害则为（假设B没有降低暴击和暴伤的属性，也没有降低命中的属性，暴击属性就是暴击率）：

http://upload.gameres.com/20118/sf_16181042_3136.jpg

    则在这种公式下，计算两个数值的价值关系，例如A的攻击增加x点和暴击增加y如果要相等，则可建立以下等式：



    计算结果为：



　　即攻击和暴击的关系在暴伤、攻击、暴击率不变的情况下，为以上等式，完全取决于基础的攻击、暴击率和暴伤

    但是，如果中间暴伤发生变化的话，就会发现等式又是一个样子了。而且这个公式在基础攻击、暴伤、暴击率取不同的数值的时候，其数值曲线的增长幅度也会差别很大，所以在这种情况下就会导致两个属性之间的关系波动非常大，如果再加上命中、闪避、韧性、降暴伤等属性，则会发现这个影响关系会更复杂。

    所以，在这种情况下作属性量化是属于一件比较不靠谱的事情，容易发生较大误差（当然，不排除有合适的方法但我不知道，如果有，求交流）

    所以，在这种情况下，我一般都会规避这种情况，而做这个模块的整体属性，将这个模块所能提供的最大属性提取，然后利用伤害公式去模拟计算一下属性平衡，然后再回头调，就可以不用去面对属性量化的问题（即无法从公式理论的角度上去找到方法，就从模拟上去找方法），虽然实际工作效率要比属性量化要低，但准确度要高于属性量化

　　**细拆各个系统：**

    首先是基础属性：

    装备的基础属性与人物有一个比例关系，按照前面所设置的各个属性部位的比例关系，可以计算出该装备的基础属性的数值，例如人物50级的攻防血为200,100,1000，而人物基础与装备基础属性为1:1，则50级所有装备所能提供的攻防血的总和为200,100,1000，而每件装备上的属性数值则根据装备的权重（例如武器的数值会高一些），以及每件装备上的属性数量来计算。

    其次是强化属性：

    根据强化属性与基础属性的比例，计算出强化属性所能提供的最大提升比，即如果基础属性比重为1，强化属性比重为2，则代表强化最大能够将基础属性放大为原先的3倍

    然后根据强化的次数，在总和一定的基础上，计算每一次强化的提升比例

    颜色属性：

    先根据比例计算攻防血的最大值，然后折算一个比例，作为随机范围。折算比例根据项目具体需求来定，例如如果需求低等级段的颜色属性有可能比高等级段的还好，那么就可以将这个比例稍低一点，让其可以与上一个等级段的范围交叉即可

    涉及到有其他不同种类的属性，则利用属性量化或者模拟计算即可。只模拟计算最大值。

    镶嵌属性：

    根据基础属性与镶嵌属性的比例关系，先可以决定攻防血所能给予的最大值。

    根据当前等级装备上的孔的总数，以及镶嵌规则，将这些属性分摊到每件装备的每个孔上去，这个数值就是该宝石在该等级段的数值。

    如果宝石的种类与攻防血不一样，则根据上一步属性量化的计算方法，来判断使用属性量化还是利用伤害公式进行一下模拟的属性计算，在其他孔恒定的基础上，选择两种不同的宝石镶嵌，让模拟计算的生存时间几乎相等即可。

**其他类推**

    模拟计算器：

    将这些能力系统的框架搭建完之后，就可以进行模拟计算器的搭建了。模拟计算器可以在输入了人物等级（装备默认跟人物等级类似），强化等级，随机属性以及镶嵌的宝石种类后，输出人物的所有战斗属性数值。

    其中唯一不太好计算的就是随机属性，这个主要在于不好模拟人物当前的属性值，但可以手动输入该部位的属性

    输出战斗属性数值之后，就可以看到当前等级的装备配置下，人物的精确属性数值

**技能**

    技能计算：

    技能的计算是一个比较麻烦的模块，很难用一个统一的公式去衡量一个技能体系是否合理（个人一直在摸索能够有一个比较系统的计算方法来计算总的技能平衡，但并未找到很满意的方法，只能从部分入手，就不提了）

    一般来说，技能主要包含这么三个大模块：放大自己对别人的伤害（伤害、dot等），减小别人对自己的伤害（控制、减伤、降攻、无敌等），回复自己受到的伤害（加血等），而这几个模块衍生出来的主要需要计算的数据为：dps，消耗，CD，持续时间等。至于距离属于一个非常麻烦的数据，基本大多游戏都是用特殊技能来规避其计算（例如冲锋，瞬移等）

    关于dps，主要为技能伤害的放大比，根据CD时间，单个技能的伤害放大比，然后求出总的放大比。例如：有三个攻击技能，第一个CD8秒，放大比为1.2，第二个CD为30秒，放大比为3，第三个CD为10秒，放大比为1.5，那么，以三个技能的最小公倍数为一个循环，即120秒内，三个技能分别可以使用15,4,12次，而普通攻击为1秒1次的话，则在中间可以插入120-15-4-12=89，所以dps为（1.2\*15+3\*4+12\*1.5+89\*1）/120=1.141667。

    控制技能表面上看起来是用来控制别人，但实际效果确实让自己少受到或者不受到伤害，所以使用对方的DPS乘以自己所能控制的时间，就是这个技能对于对方的价值，保持对战双方的防御价值相等，即是控制技能的平衡。

    消耗主要是用来判断技能能够使用的次数，以及能够持续战斗的时间，这个根据技能的使用序列计算即可，主要用来限制玩家的输出（但这一点现在各个游戏大多都简化了，很少有游戏缺蓝放不出技能，有足够的蓝无限使用技能，少了技能的使用选择和搭配，在有限的技能中将伤害最大化的判断，操作简单了，自然游戏性要弱很多，试想，如果dota完全不用考虑蓝的问题，那么这个游戏至少失色一大半）

    CD时间以及CD组主要是用来控制DPS，消耗以及玩家使用技能的频率以及技能序列的问题，该点的设计方法不做详细叙述。只在后文中简单叙述一个计算小技巧。

**技能模拟**

    技能的实际模拟首先是只关注单人PK，群P的效果主要还是由玩法以及其他大型PK机制而决定，而那种效果并非在数值关心的范围内（因为在大型PK的情况下，几乎都是XVY，玩家的PK行为不会太关注平衡的问题，除非是某项特殊属性导致某些玩家能够1VN，例如闪避过高导致N多人打一个都打不着，防太高导致N多人打一个都不破防等等）。

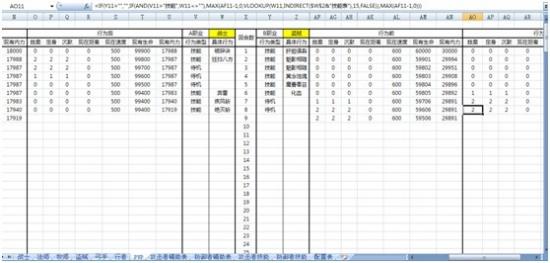
    单人PK的模拟上，暂不考虑地形，操作的影响因素，就技能本身，有两种模拟方式：

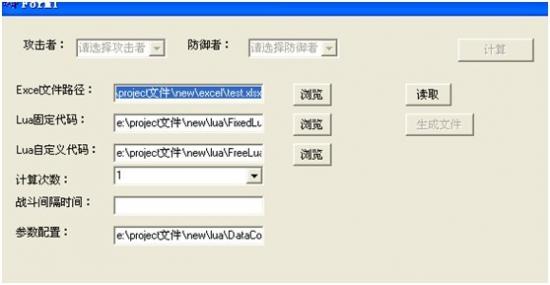
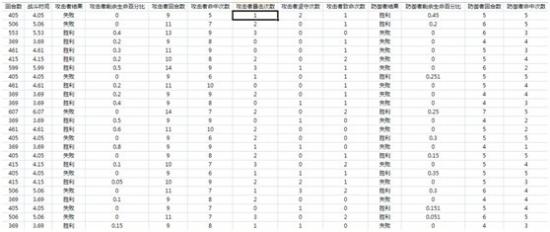
    第一种模拟方式是手工模拟，即制作一个选择列表，根据一方选择的技能，进行另一方的操作（或伤害，或被控制，或控制，或移动），两边按照所能释放的技能进行实战模拟。该方法相当于将游戏中的战斗逻辑脱离了操作（即建立在操作一样的基础上），进行一个数据模拟，该数据模拟的真实度较高，因为是人工操作，所以能够模拟出任何的实战策略，但有一个缺点，就是测试效率较低（虽然较之游戏中已经高不少了，而且游戏中的影响因素太多，影响实际结果），操作一次估计得一两分钟。

    第二种是程序模拟，输入人物的属性，写好程序逻辑，并制定好技能释放AI，默认技能顺序、应对制度（即对方使用XX技能，自己使用YY技能，自己的血低于X%时，使用什么技能等），进行高次数的循环计算，可以短时间内得到大量数据，并可以统计出胜率以及各种随机情况。该方法的最大优点就是效率高，能够短时间内获得大量数据，并进行大样本的分析。但缺点首先要实现这个工具和逻辑，门槛较高，在于技能释放AI是否准确比较难做，但可以通过多制作几个技能释放顺序，多做几套技能释放以及应对模型来研究技能的最优化配置，从而获得技能的实际数值效果。

    技能的模拟不论是哪种方法都不仅仅是数值的模拟，还有行为和系统的模拟，所以，该步一直以来都是最复杂和麻烦的。

    这两种模型我都已经实现了其模拟工具，以下附图：

  
图表 1手工模拟的工具

  
  
  
图表 2结果统计页面，由上面程序直接输出到Excel表中

    怪物数值：

    对于怪物，一般来说玩家只使用伤害技能，那么对于怪物数值来说，有了人物属性（包含各种能力来源），以及技能的平均放大比，那么就可以计算出人物的dps了

    通过模拟器，以及该怪物所需要的能力配置，即多少人多少回合，是否使用技能能杀死该怪物，就可以计算出该怪物的血量，同样，反过来，根据需要怪物对人的伤害量，以及人的防御，可以计算出怪物的攻击数值，血和攻是怪物的两个基础属性，至于其他属性，可以根据实际的需要进行配置，同样。模拟计算即可。

    作为数值，可以在制作初期先设定多个能力段的怪物模板，在实际制作中，无论是剧情，还是玩法，都可以根据其需求直接调用数值模板即可。

    有一些数值计算怪物的属性直接使用以下方法：先定某个等级的数值，然后以后的怪物就按照一定的公式直接计算，BOSS按照小怪的乘以一个数值。

    这种做法是一种很山寨而且很偷懒的做法，该方法实际上是脱离了人物在计算，那么实际游戏中，会发现跟人物的关系，要么是不破防，要么就是打人太疼，很难碰出正好合适、符合需求的数值，于是就带来各种测试和各种调整。而很多时候很多数值所给的值都是以这种方法做出来的。

**回复药品**

    其实药品无论怎么设置，花样多少，都只涉及到一个关键数值，即dps，当然，这个dps指的不是伤害量，而是单位时间内的回复量。

    可能各种游戏中有各种药，但回复类型不外乎这么两种，持续回复（每隔多少秒回复1次），瞬回。

    回复量上分为按照固定数值回，或者按照百分比回复

    无论游戏中有多少类型的药，无论是否共用CD组，只需要通过其CD，回复量，去计算其每秒最大回复量是否在你的设计范围内即可

    到此，战斗属性的几个较大模块就已经基本介绍完了，该系统是可以先脱离其他系统计算，最后再进行和其他系统关联的一个大系统。

    战斗平衡相对而言，除了技能模块较麻烦，其他较为好做，而且游戏中真正的平衡并不取决于战斗，而在于下面的第四点，经济与战斗、成长的关联。

**经验、成长**

    严格上来说，经验也应该算为一个货币，但是该货币有一定的特殊性，同时，游戏中几乎其他所有东西都建立在经验（也就是等级）的基础上，所以，将其提出来单独说明。

    首先，设定经验框架，也就是经验来源，一般有以下几种：

    一次性任务：包括剧情、引导或者其他一个账号只能做一次或者恒定次数的任务。

    循环任务：根据时间周期，以各种形式进行循环的任务或者玩法，包括每日循环，每周几次等形式。

    道具：使用某些特定道具获得经验，有次数限制，或者使用道具后可以让某些任务的经验发生变化等

    打怪：杀死怪物获得经验，该经验计算一般分单独模块

    功能：某些功能会直接获得经验或者让经验发生变化，例如师徒组队杀怪，夫妻组队杀怪，或者徒弟升级，师父获得一定经验等。

    其次，设定框架中的经验比例：

    分为任务、杀怪、道具、功能四块单独配比

    最为主要的是任务和杀怪，道具中提升杀怪经验的道具放入杀怪系统中计算

    设定每日经验获得的比重，假如40%来源于任务，40%来源于杀怪（计算双倍经验后），道具19%，功能1%的话，那么该游戏的经验任务和杀怪并重，道具空间占任务的大概50%。

    设定任务中的效率差，即多长时间内的任务是高奖任务，奖励系数为多少，使用时间多少。中奖任务多少，奖励系数，占用时间，类推。

    该框架建立后，可以模拟出一个情况：输入玩家每日上线时间，是否打怪，使用多少RMB道具，可以清晰的知道该玩家N天后的等级。同时，也可以模拟出在多少天后不同行为玩家之间的等级差，如下图（最左边为天数，表头的2,4,6,8,16,18为在线时间，中间\*\*背景的格子为道具空间的使用率，即RMB道具该玩家使用了多少）：

    框架建立后，前几天（即升级速度较快的，一天能升一级以上的天数内），最好进行人工手动模拟计算，能够尽量精确到人物的等级，在后面的升级过程中，经验是允许有误差的，但是在前期（前期以任务为主流升级途径的游戏），玩家，任务不够的情况下，如果玩家经验有误差，某些级别段卡住了，没法做任务升级，只能靠打怪，就会比较尴尬了。

    理论框架建立好之后，就根据该理论框架上的设置，给游戏中所存在的任务进行配比，同时调整RMB道具所能提供的经验不能超过该空间。

    一般来说在游戏上线后一段时期之后是可以突破该空间的，包括任务，但个人认为最好不要突破，因为升级速度是一个数值架构经济系统的重中之重，一旦它发生变化，经济系统全变，就会出现各种问题，为什么这么说，在第四点中会详细说明。

    升级速度：设计好玩家升到各个级别的升级时间，然后根据每日所能产出的经验，乘以该升级时间，就是该级别升级所需要的经验，不建议先按照公式设定好升级经验，再往下分，那么做会很尴尬的。注：尽量保证游戏中的实际升级速度跟你的数值设定一致，越精确越好。

    经验消耗：一般来说，经验是最好不要做成一个消耗的东西，因为会对升级时间、以及整个游戏的经济系统产生影响，但非得要这种设计的话，那么就涉及到经验的一次性消耗和循环消耗。将其计算入上面的经验框架中即可。

    但一旦涉及到经验消耗，而且该系统并非强制消耗的话（即不消耗后面游戏内容无法进行），就会出现玩家人工不愿消耗的情况，在这种情况下如果计算精确经验会非常尴尬，计算消耗的话，会发现如果玩家不去消耗经验，升级速度会比预想的快，如果不计算消耗的话，会发现，如果玩家一旦点了消耗，有时候会卡等级，所以，个人不推荐使用经验消耗的方式。

    同时，如果经验消耗非强制的话，会涉及到另外一个平衡，就是消耗经验所获得的东西和提升升级速度的平衡。举个例子：消耗一级的升级经验可以升级一个技能，而不学习该技能直接升级则可以穿上一件新装备，则涉及到这个技能的提升和这件装备的能力提升幅度的一个平衡上。如果技能加的还没装备明显，那么多半的人会选择升级穿装备，升级之后也会涉及到更多的玩法，获得更多的经验，而后再来学习技能即可。

    经验系统、升级速度其实计算起来并不复杂，但关键得精确，再次强调一次：尽量保证玩家实际升级速度跟设计的升级速度保持完全一致（平均升级速度和极限升级速度）。

**经济（和大概念上的经济系统并不一样，或者应该叫产出系统更贴切）**

    其实单纯的经济非常简单，并没想象中那么复杂，但是几乎没有几个游戏的经济系统做好，关键在于：其实经济复杂并不复杂在经济系统本身，而在于第四点，与战斗、经济的关联方面，第四点中细说。

    经济本身，其实跟成长几乎完全一样：拉一张大表，各种货币的产出途径，产出频率，产出等级，然后根据玩家升级速度，计算出各个玩法每日的产出，就能得出玩家在当前等级的各种货币的数量。

    注：b点中仅仅说的是产出，并未说消耗，关键在于消耗的来源是建立在战斗和成长上，单纯的经济系统中（我个人所定义的经济系统）是没有消耗的，只有产出，所以该系统并不复杂。

**战斗、成长与经济的关联**

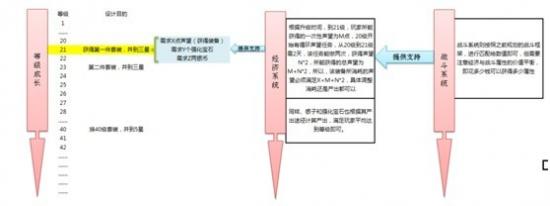
    终于说到了最关键的问题上，为什么说该点最关键，是因为该点才决定了游戏怎么玩。简单的说：战斗做得再平衡，还需要在该等级能获得才行，经济能够支撑玩家获得才行，否则，无论是玩家到该等级段经济无法支持获得相同的战斗属性，或者获得相同的战斗属性后，所花的经济代价不一样，都是属于不平衡的。

    在该关联中，是建立在以成长为基础，以经济为核心的一个数学模型上，而战斗仅仅只是提供一套平衡的数值而已，这套数值怎么放出去才是最大的问题。

    有一个观点：如果某个货币不消耗，那么是不需要它的产出的。即产出是为了消耗。这句话也许有人会觉得是句废话。打个比方，例如一个游戏，在20级之前，是没有银子消耗的，20级以后才开始有，那么在20级之前，给银子其实是一件没有意义的事情，而20级后要给它银子的话，无论是前二十级分散给，还是二十级的时候用一个一次性任务给，除了玩家感受有细微差别外，对于经济系统其实没有区别。

    所以，也就是说，产出什么东西是要支持某个既定的目的去消耗它，而产出过程中所要达到的目的是系统策划所要仔细研究的问题（即让玩家交友，提高粘度，或者提高在线时间，或者拉收入等）。

    下图是理论模型：



    *举个详细的例子：*

    某游戏设计到21级开始能够穿第一件20级的套装，28级能够穿齐5件20级的套装，并在20级的时候能够强化到全3星。而套装兑换是需要用声望，强化到3星需要一种道具和银币。

    声望的获取来源为前期的剧情和引导任务，20级开始一个循环任务，银币的来源和声望一样，道具的来源为新手礼包赠送，以及掉落。

    根据产出规则，和升级时间，计算到21级时，需要几天时间，即循环任务可以做几次，玩家所能获得的声望总数为多少，银币为多少，再来制定换装备所需要消耗的声望。

    强化中，所需要消耗的物品和银币的数量和声望处的计算规则类似。

    如果并不限定等级，那么一定就会出现玩家花多长时间，或者多少钱达到该设计目的，那么分时间计算，花时间按照时间计算产出即可（注意时间关系中玩家的等级会发生变化），花钱按照预期计算花钱的幅度即可。

    注：注意一个平衡，就是经济和战斗的平衡，即花多少钱获得多少属性的问题，除了一些特殊地方，例如新手教学等地方，其他最好保持一致。

    该数学模型建立后，事实上建立了一个基于升级成长的扩散性的大森林，基于升级成长中的设计目的，以及该处所牵涉到的规则，来计算对于经济系统产出的需求，然后再联系到战斗系统中的给值。

    此点上，有些观点认为只能靠经验，但如果仅仅只是当执行数值的话，是可以不用考虑经验的，按照以上框架，建模计算即可。

    综上，一个游戏的整体数值架构方法和关联关系就已经叙述完毕。

    按照正规做法来说，先设定各种大的基础设定，即职业设定，升级设定等，然后制定规则。

    这些大积木制作好之后，就得开始搭，即开始设计玩家在游戏中的实际行为，即上文中的多少级，或者多长时间让玩家干什么。

    根据所做的设计目的，数值开始计算各种系统的产出消耗。以及战斗系统数值的配比。

    即到最后，游戏中的各种东西，都会成为一张计算串联在一起的大计算表。一旦修改某个地方的规则，其他地方都会跟着发生一系列的变化。这种变化如果没有模拟计算表，都是属于短时间无法看出问题，而模拟计算表做的好的是能够全部模拟出，修改系统规则会产生什么问题的。

**建表**

    作为数值主要是建表思路和实际建表，一般来说，大多数数值策划在建表思路这一方面都比较欠缺，更多的都只会局部的简单计算，几乎很少能够见到能将整个游戏的成长、经济、战斗串联计算的特别精密的数值策划，这也是我评价国内数值策划水平偏低的原因（强调：并非看不起其他数值策划，仅仅是感慨，我自己水平也并不高）。

    即使有了思路，能有较好建表能力的数值策划更少见。

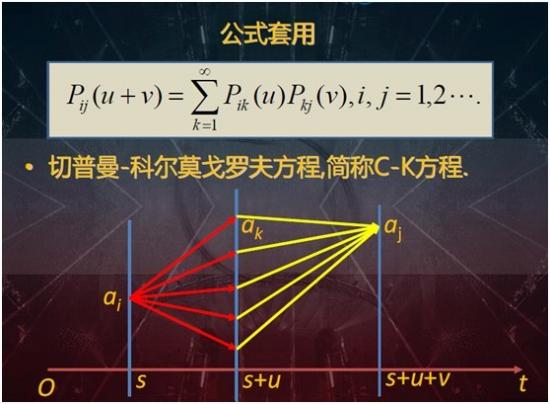
    即使有能力建立起这个数值计算模型，但还涉及到维护，在项目过程中，修改是必不可免的，维护量和维护水平就是数值策划的另外一个评价标准（当然，压根不建表的策划确实不需要维护量，而很多项目确实压根都没有这种计算表，或者是简单、不维护的计算表）。

    EXCEL有一个天生的缺陷，就是因为其数值读取都是取的表地址，所以导致如果改动涉及到改动表结构，就会非常麻烦，例如：原先设定6个职业，按照6个职业做计算表后，发现需要新增职业，这个改动基本上会让数值之前所做的工作全部白做，需要重新做一遍，所以，就需要数值策划在建表之初，进行预留许多东西。

**小技巧**

    计算强化升星的概率

    武器为1～3 级时 ，升级概率分别是 95% 90% 85%,升级成功则级别加1，升级失败，保持等级不变。  
    武器为4～6级 时 ，升级概率是 80% 75% 70%,升级成功则级别加1，升级失败，降1级  
    武器为7～9级 时 ，升级概率是 65% 60% 55%,升级成功则级别加1，升级失败，爆到0级

    则概率矩阵为：  
  
    计算方法：  
  
    或者用超大循环次数的程序模拟计算也是可以的

    技能是否能够循环起来的判断方法

    所有可使用的技能CD的倒数和如果大于等于1，则该技能序列是可循环序列，即不停的有技能使用

**Tips介绍和维护的办法**

    在游戏中牵涉到道具系统就是各种各样五花八门的道具，这些道具涉及到各种不同的属性（用途，产出途径，规则）等等。

    而物品Tips可以说是玩家的GPS系统，这个系统做得有多详细，强烈关系到玩家的感受和游戏便捷度，但事实上，各个游戏的这个系统一般来说都是采用人工手写的办法，而且一旦发生变化，就涉及到大工作量的维护上。

    所以，为了简化其操作和维护量，并能够保持其详细和精准性，可以采取以下操作（注：需要一定的excel水平，并且有详细的建模计算表，否则不要学习）

    首先，剥离道具的要素，例如：名称，简介，用途，规则，产出途径，制造需求材料（如果该物品是生活技能制造的话），需求生活技能等级，材料产出途径，产品简介和规则（如果为生活技能配方的话）。

    其次，将其中的要素，跟其所联系的系统关联起来，直接调用相应计算表中的数据。如果为商店出售，则调用商店表中的数据。如果为宝石，则调用相应宝石表的数据和宝石镶嵌规则。

    再次，将有关联的物品联系起来，将道具表形成一个自动计算的循环关联表。

    该表在建表之初是需要花一点时间去制造，但一旦制作完成，就会发现维护量非常小，同时效果非常好（在自己的项目中已经实施，实际效果非常不错）

**RibbonX**

    大多使用EXCEL都会面临一个问题，就是sheet页太多导致翻页特麻烦，多的上百页都有可能，而在这种情况下，要找东西就麻烦大了。

    介绍一个学习起来可能有点麻烦，但用起来非常简单的办法，就是ribbonX，使用它可以对excel的界面进行自定义，同时可以自定义excel的按钮，配合VBA，将自己的那些sheet页进行分类，归到各个类别中去。  
  
　   
    **系统设计思路**

    陆陆续续的已经写了很多了，关于系统设计思路方面，各种游戏，各个策划想的都不一样，一般来说，即使再牛的人写出来之后都会被喷，所以就不在此叙述了，如果愿意私下交流，可以联系。

**其他**

    写本文的目的，一是感慨一下，发发牢骚，二是抛砖引玉，希望能够引出一些大神，学习一些好的东西（当然，不知道我写的东西分量够不够引出这些大神），三是想尝试一下看能否引起这些大佬们对于数值的重视。另外也想找一些能力，风格比较接近的同仁，开发一些东西。

    愿意交流的同仁们可以加入群19482041，请注明是数值策划。

雪焰  
2011-8-14

http://bbs.gameres.com/showthread.asp?threadid=171173