# 网络游戏平衡性设计与测试

网络游戏的可玩性=网络游戏环境+规则。

网络游戏环境是给玩家第一感应，新颖、漂亮的界面，美丽和谐的3D场景，顺畅简捷而又能体现不同层次玩家的水准的网络游戏操作，安全可靠的服务器都是玩家的第一要求，或者说是他们首先能够感受到某个网络游戏的特色与好坏。

网络游戏的规则是网络游戏的核心，规则的平衡性是网络游戏生存周期的重要衡量标志，设想一个网络游戏的规则导致网络游戏中有最强或最弱的单位，属性。那么这个单位将在玩家的网络游戏中出现的频率越来越多，或越来越很少出现甚至不再使用它。这样的网络游戏可玩性就很差，因此它的生存周期也不会长了。

1． 什么是网络游戏平衡？

Sid Meier 曾经说过：“一个网络游戏是很多有趣的选择的集合。”因此得出的是如果网络游戏失去平衡，就会减少这些选择而影响网络游戏性。一个理想的网络游戏应该经过一系列的选择，最后以胜利或其它完成的条件结束。有时一些选择明显成为唯一的选择，或明显是无效的。如果在某一阶段，网络游戏出现仅有唯一的选择，而网络游戏却没有结束，就说明网络游戏的平衡性有了问题。

 几乎所有通常所谓的不平衡都来自选择权的减少。例如，在一个策略网络游戏里，如果某一种部队的作用和费用相比过于划算，就会造成其它的部队几乎或完全没有作用。这种情况不仅只留给玩家一个选择（无从选择），而且使玩家受到很多不相关的干扰。这些干扰实际上让网络游戏变得比较迷乱，减损了网络游戏性，而且让玩家感到灰心。

2． 如何达到可平衡性

网络游戏平衡性通常被认为是alpha或beta测试的事情，但事实上就像任何工程，好的准备工作是实现良好网络游戏平衡的关键。优秀的网络游戏设计具有极大的可平衡性，也就是指网络游戏系统可以较容易地调整到平衡的状态。如果系统没有可平衡性，费尽周折也不可能将网络游戏调整到平衡。

一个网络游戏是一个系统，在设计初期应用良好的系统设计方式将带来较好的可平衡性。好的系统设计方式可以分成三个重要步骤：网络游戏要素的模块性，连贯的设计宗旨及对复杂性的控制与调节。在设计的早期就采用这些方法将为设计师在网络游戏测试的alpha和beta阶段节省大量的时间。

3． 网络游戏平衡性的概念

网络游戏平衡性概念的分类：

1、 按平衡性涉及的范围来看，有局部平衡性（微观调控）和整体平衡性（宏观调控）

2、 从时间或网络游戏结构观察，则有静态平衡性与动态平衡性

3、 广义的网络游戏平衡性可分为：

玩家/玩家间的平衡性

玩家/网络游戏规则间的平衡性（严重影响网络游戏的可玩性）

网络游戏内部的平衡性（网络游戏可玩性/网络游戏可玩性间的平衡）

网络游戏平衡性

玩家/玩家间的平衡性

具等级（时间）

玩家/网络游戏规则间的平衡性

网络游戏内部的平衡性

4．基本网络游戏平衡过程

除了基本的规则和技巧之外，过程是非常重要的。网络游戏的平衡过程有几个步骤，每个步骤都有各种各样的技巧。

首先要考虑的是让网络游戏进入一个有趣及可玩的境界，这就需要宏观调控，或者说让网络游戏中的大部分要素至少达到基本上平衡，而且不存在任何要素过分地不平衡。只要达到这个状态，就可以继续细调网络游戏要素的具体部分，如RTS网络游戏里的种族或派系。

当然在网络游戏alpha测试阶段之前通常应已进行了宏观调整，所以可能随着新功能的增加要重新进行调整。家园（Homeworld）的主策划Erin Daly提出，应将相关的功能在同一时间加入，然后做一个宏观调控，基本上这是在整个开发过程中保持网络游戏可玩性的最有效的方法。

一旦实现最后的宏观调整，最好在alpha测试阶段的后期，就可以对网络游戏进行微观调控1使网络游戏平衡达到完美的程度。

宏观调控

提供一个可平衡的网络游戏系统显然只是达到网络游戏平衡的第一个步骤。即便是最完美的设计也需要变成现实，而在实施的过程中错误就会出现，在初期设计中就经常会出现小错误。许多网络游戏价值在整个网络游戏实现之后才能被清楚认识到。在这些情况下，设计者必须在alpha测试阶段之前及测试期间运用宏观调控技巧校正平衡值。

宏观调控应在微观调开始之前结束；如果网络游戏的基础还在不断改变时，较小的平衡性的改变将变得没有效果而无用。在进行宏观调时，目的是“找到”在设计案中描述的网络游戏性目标。当然，在你还不清楚如何表明核心网络游戏性时是不可能进行网络游戏细节的调整。

为了瞄准核心网络游戏性，明确地说明核心网络游戏性及其如何体现是很重要的。只要做到这一步，就可以建立一定的基线，也就是Ensemble Studios1所谓的“定锚”。举例说明，你也许设立网络游戏速度的基线为“大约10分钟长的网络游戏”，或者设立角色韧性的基线为“被一个危险怪兽攻击3次是致命的”。一旦你为每个网络游戏因素（一个地图、一个角色类型，一段对话等等）都找到满意的基线，就可以利用这些网络游戏要素基线为根据扩展网络游戏。

微观调控

一旦网络游戏已受宏观调控，网络游戏的平衡必须要进入细节调校。如果网络游戏至少达到有点乐趣可言，且不存在明显的问题，则已基本上完成宏观调控并可开始转向微小细节。微观调控是网络游戏策划为了进一步完美平衡性而实施的小手术。一个小手术一般被定义为：变化值相对于一个“全球”数值（影响许多其它的网络游戏要素）要少于10%，相对于一个“地方性”数值（一个单一网络游戏要素）则应少于30-40%。

微观调控最大的挑战是找到问题。一旦找到问题，就可以开始稍微调整数值，但要注意不要因此再产生出新的问题。良好的要素模块和预先计划在这一阶段很有效果——没有它们，就可能做不到在一个合理时间范围内完成网络游戏的平衡。

辨识较小的不平衡性

策划有几个技巧可辨识较小的不平衡性。其中最明显的做法是大量地测试网络游戏，寻找一贯受惠或占优势的方法，或寻找从不被使用的方法。另一个常用方法是与一个试验人或另一个策划讨论假设的情景或与其对战，找到一个一致认为会产生的结果，然后在网络游戏中测试是否会发生同样的结果。

如果策划用第一种方法，只是寻找占优势（或从不被使用）的方法，确定这种情况产生的实际原因是很重要的，并确认事情是否应该这样发展。对不平衡性进行分类，并尝试将其归类到典型的不平衡性中，有助于理解问题。基本上越是了解不平衡性的类型及特征就越能够调整它。

发现不平衡性的第二个方法有时被称为“追逐不平衡性”，也就是一个假定的情景被定义后，而由此产生的各种可能的行动及结果都应是符合设计的。例如，一个坦克部队的冲锋应该被认为打败一个轻型车队的进攻，但同时也应该受到轻度伤害，而面对防坦克步兵团的反攻则应受到重创。如果在实际的网络游戏中，一个坦克部队的冲锋可完全歼灭一个轻型车队并可以与防坦克步兵团不分胜负，此时坦克部队的过于强大就造成了不平衡性。追逐不平衡性是十分重要的，如果严格地执行，很容易就可以发现75%以上的较小不平衡性问题。网络游戏往往不按照策划所愿望的某种特别方式发展，特别是在一个对抗型的多人网络游戏中，一个“地区性”平衡价值的微小变化就可以造成网络游戏的平衡与不平衡。

修整较小的不平衡性

一旦辨识并证明了不平衡性，是否容易进行修正呢？是的，如果网络游戏已被设计为容易调整！ 一个非常可调的网络游戏具备的素质能让设计师在不间接影响其它网络游戏因素的情况下，专门对付某一不平衡性。

校正的重要一点是保持细调的水准（往小的方面想），尤其是升级网络游戏的时候。一个过于强大的网络游戏要素容易使其它要素失去效力，而一个过于无力的网络游戏要素则会被忽略而毫无效果。

还很重要的一点是细调时不要影响到其它的网络游戏数值，譬如，在角色扮演网络游戏里考虑一个叫做“火球”的符咒，它是火系符咒的一种。如果火球威力过大，策划可做的就是全面降低火系魔法的威力，或将火球降级。很明显你应该做的是选择“地方性”的解决方法，在细调全面火系魔法之前将火球降级。这是一个很简单的例子，在大多情况下，网络游戏要素之间都存在一定程度的互相依赖。谨慎地考虑一个改变将会带来的冲击力，尝试使用专门解决问题而不影响其它网络游戏要素的方法。

最后，避免“过度解决”不平衡性。当策划在同一时间运用多重不同的细调方法来解决一个特定问题时就会产生“过度解决”的情形。这样就很难决定变化所带来的效果，因为你应用了多重独立的可变性来影响一个不独立的可变性。“过度解决”也有可能因意外地影响其他网络游戏要素而带来麻烦。

 mmorpg（多人在线角色扮演类网络游戏）

平衡性设计及测试的方法

第一部分 mmorpg网络游戏平衡性的设计

数据库的分析：

是一个多人在线角色扮演类网络游戏的数据库的存储过程，其中包含了多个多个视图，此存储过程可以修改和删除任一数据表中的数据。

模块的划分

分析此数据库从平衡性影响的角度来观察，可以分为3个模块：

1.角色，职业，技能，武器，防具，这几个类必须划分到一个模块中，它们之间互相影响，存在这5个组件结构上的动态平衡性。

2.物品类比较特殊，它不影响以上几个类的数据平衡性，每个玩家使用的物品是自由，谁都可以使用任意物品。它的平衡性只会影响玩家/网络游戏规则间的平衡性

3.敌人（NPC）类则是另外的一个模块，模块内的平衡性也需要设计与测试。

将角色，职业，技能，武器，防具类组成的模块是网络游戏平衡性的关键，因此将数据库中这几个类拿出构成一个类图，如图1-2。

每个类中包含3个方面：类名，类的特有属性（特性），类与其他类的关联关系。

角色类与职业类比较特殊，它们是多对多的关系，并且职业类依赖于角色类。

由此模块的每个组件的关系不难看出，它的平衡性由两部分组成：角色类组件的平衡性和职业类组件的平衡性。

两个组件都应该达到随时间（在MMORPG类网络游戏中时间意味着玩家的等级）变化的动态平衡性和组件结构上的动态平衡性。

尤其是职业类的结构动态平衡性，它与技能，武器，防具3个类由关联，这3个类的时间动态平衡性组合起来构成职业类的结构动态平衡性。只有具有结构上的动态平衡性才会使得网络游戏的可玩性大大加强。

用平衡度来定量的代表组件的平衡性，平衡度为0时，表示该组件绝对平衡，那么：

职业的平衡度=技能的平衡度+武器的平衡度+防具的平衡度

即：

P职业=P技能+P武器+P防具

要达到职业的平衡度为0，最容易的办法就是：

让技能的平衡度=0，武器的平衡度=0，防具的平衡度=0。

这样的设计就属于一个同化的过程，这样大大降低了网络游戏的可玩性。只有异化的过程，也就是让他们在结构上达到动态平衡，才会使网络游戏的可玩性得到肯定。

网络游戏的平衡性越容易实现，则这个网络游戏的可玩性就越低，完全的同化效果会使网络游戏丧失可玩性。

网络游戏平衡性容易度与网络游戏可玩性之间的关系如图1-3所示：

网络游戏的可玩性

网络游戏平衡容易度

设计角色，职业，技能，武器，防具每个组件的平衡度：

要使网络游戏的平衡度P=0，则要：

P角色=0，P职业= P技能+P武器+P防具=0

每个组件的P值都使随时间（网络游戏中玩家的等级的衡量标志）变化的动态平衡度

具体的数值设计

一． 角色类的平衡度

角色类中包含4个特性，没有包含方法属性。因此角色类的平衡度就是它特性的综合平均平衡度。

问题求解步鄹与方法：

 对角色类中的每个特性求随时间（网络游戏中的意义是等级）变化的动态平衡度。

求所有特性动态平衡度的平均值（用Si表示，表示第i对象的特性综合平衡度）。

 使用误差法求得角色类所有对象（角色的个数）综合起来的平均平衡度（P角色）。

具体实现：

 用角色等级（时间）做为函数自变量，得到特性随时间变化得动态平衡度，

其他5个特性MAXMP，力量，敏捷，智力，速度与MAXHP相似，只不过直线的斜率不同。

每个特性的平衡度使用求一阶（二阶）导数的方法，设为Kmaxhp, Kmaxmp, K力量，K敏捷，K速度，K智力。

则所有特性的动态平衡度平均值Si = Kmaxhp+Kmaxmp+K力量+K敏捷+K速度+K智力。

用误差法综合角色类所有对象，求得角色类得平衡度P角色：

(S1+S2+S3+…….+Sm)/m —S=P角色

其中m表示由m个对象，即有m个角色。

S是任意一个角色的所有特性的动态平衡度平均值，当然S的取值最好是取中间值，最佳值是所有S中最中间的值。

设法使得P角色接近于零。

二． 职业类的平衡度

P职业=P技能+P武器+P防具

1． 技能的平衡度

技能类包含了9个特有的属性（特性），1个方法限制，方法中的属性是角色类中的特性。

问题求解步鄹与方法：

a) 对技能中的每个特性求随时间（网络游戏中的意义是等级）变化的动态平衡度。

b) 求技能类所有特性动态平衡度的平均值。

c) 使用误差法求技能类所有对象（技能）综合起来的特性平均平衡度。

d) 对限制技能类的方法中的属性求随时间变化的动态平衡度。

e) 求所有方法中的属性得动态平衡度的平均值。

f) 使用误差法求技能类所有对象（技能）方法中属性的综合平均平衡度。

g) 由其特性的平均平衡度和方法属性的平均平衡度得到整个组件的动态平衡度，即为：

P技能=P特性+P（）。

具体实现为

2 用技能等级（时间）做为函数自变量，得到技能类中特性随时间变化得动态平衡度，如图1-5, 图1-6, 图1-7, 图1-8：

技能等级分为：冰箭，雷箭，箭雨，终极箭。

由以上特性的时间动态变化函数得到每个特性的动态平衡度：

每个特性的平衡度使用求二阶导数的方法，设为K耗费mana, K威力, K回避率，K命中率，K分散度，K影响范围。

则所有特性的动态平衡度平均值Si = K耗费mana+K威力+K回避率+K命中率+K分散度+K影响范围。

用误差法综合技能类所有对象，求得技能类得特性平衡度P特

(S1+S2+S3+…….+Sm)/m —S=P特性

其中m表示有m个对象，即有m个技能。

S是任意一个技能的所有特性的动态平衡度平均值，当然S的取值最好是取中间值，最佳值是所有S中最中间的值

下面的技能类方法的属性平衡度P（）也可求得， P（）= (S1+S2+S3+…….+Sm)/m —S

Si=K力量限制 || K敏捷限制 || K智力限制 ，3个K是或的关系。

S是任意一个技能受限制的属性的动态平衡度平均值，当然S的取值最好是取中间值，最佳值是所有S中最中间的值。

由求得的P特性和P（）得到：P技能= P特性+P（）。

2． 武器的平衡度

武器类包含了2个特性，2个方法，两个方法分别影响角色类中的力量和敏捷这两个特性。

问题求解步鄹与方法：

a) 对武器类中的每个特性求随时间（网络游戏中的意义是武器的级别）变化的动态平衡度。

b) 求武器类所有特性动态平衡度的平均值。

c) 使用误差法求武器类所有对象（武器）综合起来的特性平均平衡度。

d) 对增加角色基本属性-力量或敏捷值的方法中的属性求随时间变化的动态平衡度。

e) 求所有方法的属性的动态平衡度的平均值。

f) 使用误差法求武器类所有对象（武器）方法中属性的综合平均平衡度。

g) 由其特性的平均平衡度和方法属性的平均平衡度得到整个组件的动态平衡度，即为：

P武器=P特性+P（）。

具体实现

2 用武器级别（时间）做为函数自变量，得到武器类中特性随时间变化得动态平衡度，如图1-10：

武器等级分为：木弓，硬弓，宝石弓，暗杀弓。

由上面特性的时间动态变化函数得到每个特性的动态平衡度：

每个特性的平衡度使用求二阶导数的方法，设为K价格， K攻击力

则所有特性的动态平衡度平均值Si = K价格+ K攻击力

用误差法综合武器类所有对象，求得武器类得特性平衡度P特性：

(S1+S2+S3+…….+Sm)/m —S=P特性

其中m表示有m个对象，即有m个技能。

S是任意一个武器的所有特性的动态平衡度平均值，当然S的取值最好是取中间值，最佳值是所有S中最中间的值。

2 用武器级别（时间）做为函数自变量，得到武器类影响其他类的方法中的属性随时间变化得动态平衡度，如图1-11：

同理：

武器类方法的属性平衡度P（）也可求得， P（）= (S1+S2+S3+…….+Sm)/m —S

Si=K增加力量 || K增加敏捷，2个K是或的关系。

由求得的P特性和P（）得到P武器= P特性+P（）。

3． 防具的平衡度

如图所示，防具类包含了4个特性，1个方法，方法影响角色类中的智力这个特性。

问题求解步鄹与方法：

a) 对防具中的每个特性求随时间（网络游戏中的意义是防具的级别）变化的动态平衡度。

b) 求防具类所有特性动态平衡度的平均值。

c) 使用误差法求防具类所有对象（防具）综合起来的特性平均平衡度。

d) 对增加角色基本属性-智力值的方法中的属性求随时间变化的动态平衡度。

e) 求所有方法的属性的动态平衡度的平均值。

f) 使用误差法求防具类所有对象（防具）方法中属性的综合平均平衡度。

g) 由其特性的平均平衡度和方法属性的平均平衡度得到整个组件的动态平衡度，即为：

P防具=P特性+P（）。

具体实现

 用防具级别（时间）做为函数自变量，得到防具类中特性随时间变化得动态平衡度，如图1-12，图1-13，图1-14：

防具等级分为：布衣套装，轻皮套装，光明套装，精英套装。

由上面特性的时间动态变化函数得到每个特性的动态平衡度：

每个特性的平衡度使用求二阶导数的方法， 设为K价格， K物理防御，K魔法防御，K回避率。

则所有特性的动态平衡度平均值Si = K价格+ K攻击力

用误差法综合武器类所有对象，求得武器类得特性平衡度P特性：

(S1+S2+S3+…….+Sm)/m —S=P特性

其中m表示有m个对象，即有m个技能。

S是任意一个武器的所有特性的动态平衡度平均值，当然S的取值最好是取中间值，最佳值是所有S中最中间的值。

用武器级别（时间）做为函数自变量，得到武器类影响其他类的方法中的属性随时间变化得动态平衡度，

具等级（时间）

玩家/网络游戏规则间的平衡性

+智力值

同理：

防具类方法的属性平衡度P（）也可求得， P（）= (S1+S2+S3+…….+Sm)/m —S

Si=K增加智力。

由求得的P特性和P（）得到P防具= P特性+P（）。

由此得到P职业：P职业=P技能+P武器+P防具

要想职业得到平衡，就必须设法使得P职业接近于零，从而可以反馈到影响职业类的每个组件中去，在每个组件中调节组件的属性达到最终需要的效果。

三． 总结网络游戏综合平衡度

根据以上设计的方法，我们只要将P角色和P职业都接近于零，或者使二者的矢量和接近于零就能达到MMORPG网络游戏的基本平衡性，当然对于NPC我并没有考虑进来，因为NPC和物品类都不会影响玩家与玩家之间的平衡性和网络游戏规则的平衡性。

而且NPC和物品之间也没有联系，它们都可以作为一个独立的组件来设计平衡性，至于设计的方法，是与角色/职业类的方法完全相似的。

至于玩家/网络游戏规则间的平衡，显而易见是取决与设计中的函数表达式的合理性。

上面的设计中，函数的取值是离散的，也就是说你可以把离散的值放在一个连续的函数中，也可以直接使用离散的，哪种决策更好那就要实际情况实际分析了。但这都不是难点，尽管它们很重要。函数式的设计我认为一个高级的网络游戏玩家加上一定的数学功底就可以设计的很出色。

第二部分 mmorpg（多人在线角色扮演类网络游戏）

——平衡性测试的方法

一．网络游戏测试的阶段

网络游戏测试包括网络游戏软件测试和网络游戏可玩性测试两打部分。

在软件的后期测试中包括两个测试阶段：alpha测试阶段，beta测试阶段。

二．网络游戏平衡性测试阶段

对于网络游戏的平衡性来说alpha测试阶段，beta测试阶段是完成网络游戏平衡性重要阶段，它们占据了网络游戏生命周期的30%，占据了网络游戏平衡性设计和修改的70%。

提供一个可平衡的网络游戏系统显然只是达到网络游戏平衡的第一个步骤。即便是最完美的设计也需要变成现实，而在实施的过程中错误就会出现，在初期设计中就经常会出现小错误。许多网络游戏价值在整个网络游戏实现之后才能被清楚认识到。在这些情况下，设计者必须在alpha测试阶段之前及测试期间运用宏观调控技巧校正平衡值。

alpha测试最大的挑战是找到问题。一旦找到问题，就可以开始稍微调整数值，但要注意不要因此再产生出新的问题。良好的要素模块和预先计划在这一阶段很有效果——没有它们，就可能做不到在一个合理时间范围内完成网络游戏的平衡。

近年来，愈加流行的方法是秘密记录（不告诉玩家）网络游戏成果及统计数据。网络游戏世纪帝国（Age of Empires），雪乐山（Sierra）发表的几款网络游戏及斗争阴影（Strifeshadow）都受益于这个技巧。有时这些统计数据具开导性，有时也极具误导性。

对所有的数据都应有所保留。有时一个不成熟的测试人群会带来很不正确的结果，只因为他们不熟悉网络游戏，而且没有机会全面尝试（或只是尝试最容易的部分）。同样的，一个过于成熟的测试人群也有可能忽视其它策略的潜力，或被困在一个很高级却较模糊的不平衡点，而这些不平衡与其它更明显的不平衡点相比显得不那么紧迫。Ethermoon娱乐公司在网络游戏斗争阴影里应用的一个极为有效的技巧就是夸大在beta阶段的补丁中的网络游戏平衡性变化，来怂恿玩家尝试新的战略，而不再继续“抵抗”新的变化。

要紧记的一条是无论何时进行不平衡性的搜索，在网络游戏早期所设置的网络游戏要素，往往比后期的网络游戏要素要敏感得多。仅仅因为一个早期网络游戏要素的不平衡性会影响在它之后设置的所有东西，而后期网络游戏要素能够制造麻烦的时间有限。正如有必要在做微观调控之前先做好网络游戏的宏观调控，也有必要先平衡早期的网络游戏因素。

beta阶段的补丁中的网络游戏平衡性变化是通过网络游戏玩家的反馈完成的。

本文本源 http://www.cdyouxi.com/wangluoyouxi/477/